A DYNAMIC TARIFF MARKET DESIGN FOR ELECTRICAL ENERGY IN A DISTRIBUTION SYSTEM

by

Craig Poulin

A thesis submitted in partial fulfillment of

the requirements for the degree of

Master of Science

(Electrical Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

2015

Contents

Abstract v
Disclaimer vi
Abbreviations & Acronyms vi
List of Tables vii
List of Figures vii
1 Introduction 1
1.1 Motivation and the Dynamic Distribution System 1
1.2 Statement of Problem and Research Questions 3
1.3 Technical Approach 5
2 Literature Review 9
2.1 Overview 9
2.2 Distributed Energy Resources 9
2.3 DER Impact on Utility Business Models 15
2.3.1 Traditional Business Models and Volumetric Rates 15
2.3.2 The Incentive Regulation Approach 24
2.3.3 The Network Utility Approach 25
2.4 Electrical Energy Wholesale Markets 27
2.4.1 Marginal Costs and Market Clearing 28
2.4.2 Market Structures 29
2.4.3 Designing and Testing Market Rules 31
2.4.4 Forward and Spot Markets 31
2.4.5 Locational Pricing 33
2.5 Dynamic Pricing and Price Elasticity of Demand 34
3 Dynamic Tariff Distribution Marketplace 42
3.1 Overview 42
3.2 Design Objectives 43
3.2.1 End Goals 44
3.2.2 Guiding Principles 45
3.3 Network Structure 47
3.3.1 Physical System Representation 50
3.3.2 Non-Physical System Components 51
3.4 Marketplace Parameters and Process 53
3.4.1 Marketplace Parameters 53
3.4.2 Marketplace Process 55
3.5 Demand Curves 59
3.5.1 Node Curve Generation 62
3.5.2 Curve Aggregation 67
3.5.3 Tariff Curve Generation 69
3.5.4 Curve Adjustment 75
3.5.5 Aggregation and Adjustment Illustration 76
3.5.6 Translation 79
3.6 Market Operation 80
3.6.1 Marketplace and Supply Nodes 80
3.6.2 Market Auction 82

3.6.3 Node Marginal Pricing 85
3.7 Real-Time Actions and Settlement 85
3.8 Implementation 90
3.8.1 Practical Considerations 91
3.8.2 Limitations and Assumptions 92
4 Data Sources 95
4.1 Pecan Street 95
4.2 NYISO 100
5 Simulation: Matlab Implementation 103
5.1 Overview 103
5.2 Create DDS Object 105
5.3 Create Node Objects 108
5.3.1 Construction and General Parameters 108
5.3.2 Demand Curves 110
5.3.2.1 Qbin 110
5.3.2.2 Linear Interpolation 113
5.3.2.3 Pmin 114
5.3.2.4 Pcap 117
5.3.3 Linkage Loss Parameter 120
5.3.4 Power Factor 123
5.4 Create Tariff Objects 124
5.5 Prepare Simulation 125
5.6 Simulation: Market Operation 128
5.6.1 Demand Curve Generation 129
5.6.2 Demand Curve Adjustment 133
5.6.3 Demand Curve Translation 138
5.6.4 Demand Curve Aggregation 150
5.6.5 Market Clearing Point 157
5.6.5.1 Clearing Point: Islanding Mode 157
5.6.5.2 Supply Nodes 159
5.6.5.3 Clearing Point: Supply and Demand 161
5.6.6 Node Marginal Pricing 167
5.6.6.1 Clearing Price to Contract Price 168
5.6.6.2 Contract Price to Clearing Price 172
5.7 Simulation: Real-Time Actions 177
5.7.1 Set Bottom-Level Node Consumption 177
5.7.2 Determine System-Wide Energy Flow 179
5.7.3 Run Dispatcher Nodes 182
5.8 Simulation: Settlement 190
5.9 Analyze Results 192
6 Simulation: Behavior Models and Proposed Tariffs 197
6.1 Overview 197
6.2 Load Node Behavior Model: Isoelastic with Anchor Point 198
6.2.1 Consumption Preferences and Demand Curve Generation 199
6.2.2 Real-Time Actions 214
6.2.3 Limitations and Areas for Improvement 216

6.3 PV Node Behavior Model 217
6.3.1 Consumption Preferences and Demand Curve Generation 217
6.3.2 Real-Time Actions 221
6.3.3 Limitations and Areas for Improvement 222
6.4 Storage Node Behavior Model 223
6.4.1 Consumption Preferences and Demand Curve Generation 225
6.4.2 Real-Time Actions 230
6.4.3 Limitations and Areas for Improvement 233
6.5 Tariff Design Overview 235
6.6 Capacity Tariff 236
6.6.1 Goals and Tariff Curve Creation 237
6.6.2 Settlement 241
6.6.3 Limitations and Areas for Improvement 244
6.7 General Target Quantity Tariff 245
6.8 Ramping Tariff: Simplified Moving Average 248
6.8.1 Goals and Tariff Curve Creation 249
6.8.2 Settlement 253
6.8.3 Limitations and Areas for Improvement 254
6.9 Ramping Tariff: Exponential Moving Average 256
6.9.1 Goals and EMA Technical Basis 256
6.9.2 Tariff Curve Generation 262
6.9.3 Settlement 264
6.9.4 Limitations and Areas for Improvement 265
6.10 Flat-Rate Tariff 265
7 Simulation: Case Studies 269
7.1 Overview 269
7.2 Scenario 1: Load Elasticity 276
7.2.1 Configuration 276
7.2.2 Results 277
7.2.3 Observations 280
7.3 Scenario 2: Capacity Tariff 282
7.3.1 Configuration 282
7.3.2 Results 282
7.3.3 Observations 286
7.4 Scenario 3: Ramp SMA Tariffs 289
7.4.1 Configuration 289
7.4.2 Results 290
7.4.3 Observations 297
7.5 Scenario 4: Ramp EMA Tariff Location 299
7.5.1 Configuration 299
7.5.2 Results 300
7.5.3 Observations 307
7.6 Scenario 5: Dispatcher Node 308
7.6.1 Configuration 308
7.6.2 Results 310
7.6.3 Observations 315

8 Conclusion ... 317
8.1 Answers to Research Questions ... 317
8.2 Contributions Made ... 323
8.3 Future Work ... 324
8.3.1 Improved Modeling Flexibility ... 325
8.3.2 Physical Implementation ... 326
8.3.3 Improved Simulation Behavior Models ... 326
8.3.4 Advanced Market Rules ... 327
8.3.5 System Operator Optimization ... 328
8.3.6 Utility Business Model Analysis ... 329
8.4 Summary ... 331
9 References ... 333
Appendix A: MATLAB Case Study Initialization ... 335
Appendix B: MATLAB Object Classes and Parameters ... 341

Abstract

The work in this thesis is motivated by the need to design an energy marketplace for use in a Dynamic Distribution System. The Dynamic Distribution System is a proposed techno-economic approach to increased penetration of distributed energy resources in the electrical energy system. Distributed energy resources pose new challenges and opportunities to traditional electric energy business models. This research reveals proposed business model alternatives that provide insight into possible roles of a distribution system marketplace. Further insights are provided by existing wholesale energy markets and studies of dynamic pricing in retail energy markets.

First, the proposed marketplace rules for a Dynamic Tariff Distribution Market are defined including the network configuration, communication processes, actor participation, and market processes. The network includes nodes, tariffs, and linkages. The Dynamic Tariff Distribution Market protocols and processes seek to provide all actors their network- and time-specific energy price, comprising the unit cost of energy, energy loss adjustment, and the impact of system externalities.

Second, a method for simulating the Dynamic Tariff Distribution Market is provided. This section describes specific considerations for simulating and implementing the marketplace. The simulation includes market rules and processes, methods for implementing node behavior models, and methods for implementing tariff structures. The simulation is then used to examine suggested actor behavior models and proposed tariff structures. Case studies are provided to demonstrate marketplace interactions and examine behavior models and tariff structures.

Finally, this thesis provides suggested areas for future research and development of the Dynamic Tariff Distribution Market. This includes physical implementation, more flexible modeling, improved behavior models, and advanced market rules.

Disclaimer

The views expressed in this thesis are those of the author and do not reflect the official policy of the United States Air Force, Department of Defense, or the U.S. Government.

Abbreviations & Acronyms

AMI – Advanced Metering Infrastructure
ARPA-E – Advanced Research Project Agency-Energy
AWG – American Wire Gauge [conductor size]
CA ISO – California Independent System Operator
CHP – Combined Heat and Power [generation]
CCHP – Combined Cooling, Heat, and Power [generation]
CERTS – Consortium for Electric Reliability Technology Solutions [microgrid protocol]
DA – Day Ahead [energy market]
DAM – Day Ahead Market
DC – Direct Current
DDS – Dynamic Distribution System
DER – Distributed Energy Resources
DG – Distributed Generation
DSO – Distribution System Operator
DTDM – Dynamic Tariff Distribution Marketplace
EMA – Exponential Moving Average
FOA – Funding Opportunity Announcement
FRE – Flat-Rate Equivalent [energy price]
GW – Gigawatt
HEM – Home Energy Manager
ISO – Independent System Operator
LBMP – Locational Based Marginal Price
LH – Left-Hand [limit of a curve]
MW – Megawatt
NEM – Net Energy Metering
NMP – Node Marginal Price
NYISO – New York Independent System Operator
NZEB – Net Zero Energy Buildings
O&M – Operations and Maintenance
P – Price
P – Real Power [not used unless denoted]
PV – Photovoltaic [solar generation]
Q – Quantity
Q – Reactive Power [not used unless denoted]
RH – Right-Hand [limit of a curve]
RMI – Rocky Mountain Institute
RPS – Renewable Portfolio Standard

RT – Real Time [energy market]
RTD – Real Time Dispatch
RTP – Real Time Pricing
RTO – Regional Transmission Organization
SDG&E – San Diego Gas and Electric [utility]
SMA – Simplified Moving Average
T&D – Transmission and Distribution
TOU – Time-of-Use [energy pricing]
ZNE – Zero Net Energy

List of Tables

Table 1: Pecan Street Use Categories [15] 97
Table 2: Pecan Street Category Data - Example Household Annual Summary [15] 98
Table 3: Example NYISO RT LBMP Data [16] 101
Table 4: Example Demand Curve Values 111
Table 5: Example Demand Curve Values with Pcap 118
Table 6: Loss Translation Generalization 143
Table 7: Example Curve Aggregation Methods Data 151
Table 8: Pecan Street Household 22 Data 207
Table 9: Pecan Street Household 22 Data 211

List of Figures

Figure 1: Dynamic Distribution System [1] 2
Figure 2: Positive and Negative Impacts of DG on the Electricity Value Chain [2] 11
Figure 3: CA ISO "Duck Curve" [3] 13
Figure 4: Madison Gas & Electric TOU Rate Structure [5] 17
Figure 5: The Potential Impact of NEM and ZNE Customers on Capturing Fixed Costs [2] 19
Figure 6: SDG&E's Proposed Network Use Charge [2] 22
Figure 7: NY ISO Market Timeline [9] 33
Figure 8: Impact of Elasticity on Demand Curves 37
Figure 9: Dynamic Distribution System with Marketplace [1] 43
Figure 10: Energy and Cash Flow Conventions 48
Figure 11: Node, Linkage, and Tariff Ownership Illustration 49
Figure 12: Linkage Loss and Tariff Ownership and Hierarchy 52
Figure 13: Marketplace Process for 5-Minute Market Duration and Settlement Interval 55
Figure 14: Marketplace Process 58
Figure 15: Demand Curve Quadrants 60
Figure 16: Typical Load Node Demand Curve 62
Figure 17: Illustrative PV Node Demand Curve 64
Figure 18: Illustrative Storage Node Demand Curve 65
Figure 19: Example Demand Curve Aggregation 68
Figure 20: Example Tariff Settlement Functions and Tariff Curves 71
Figure 21: Example Demand Curve Adjustment 75
Figure 22: Combination of Demand Curves (Blue Demand, Green Tariff, Red Combination) 78
Figure 23: Example Clearing Price and Quantity 82

Figure 24: Example Communication Process 90
Figure 25: NYISO DA and RT LBMP Comparison [16] 101
Figure 26: Example Demand Curve 111
Figure 27: Example Demand Curve with Pcap at Limits 120
Figure 28: runMarket (1/3) 132
Figure 29: Example Trimming in Demand Curve Adjustment 135
Figure 30: runMarket (2/3) 137
Figure 32: Translated Curve Segment (floor Function) 145
Figure 32: Translated Curve Segment (round Function) 145
Figure 33: Translated Curve Segment (ceil function) 145
Figure 34: Translated Curve Segment (round Function) for Negative Prices 146
Figure 36: Translated Curve Using Rounded ΔQ for ΔP 148
Figure 36: Translated Curve Using Unrounded ΔQ for ΔP 148
Figure 37: Example Curve Aggregation Methods 152
Figure 38: Example PV Demand Curve 155
Figure 39: Example Storage Demand Curve 155
Figure 40: Example PV and Storage Curve Aggregation 155
Figure 41: Market Clearing Point Cases 166
Figure 42: Clearing Price vs Contract Price 168
Figure 43: runMarket (3/3) 176
Figure 44: setQuantActual 178
Figure 45: simEstimate, no Dispatch Nodes 181
Figure 46: Example Dispatch Levels 184
Figure 47: runDispatch Process 189
Figure 48: Two Load Nodes 199
Figure 49: Isoelastic Demand Curve Examples 208
Figure 50: Example Impact of Pmin on an Isoelastic Demand Curve 209
Figure 51: Example Load Curves with Aggregation 212
Figure 52: Example PV Node Demand Curve 221
Figure 53: Storage Node Price Thresholds 227
Figure 54 Capacity Tariff Curve Examples 241
Figure 55: Example Target Quantity Tariffs 247
Figure 56: SMA Ramp Cycling 255
Figure 57: DTDM Network for Case Studies (Baseline; Scenarios 1.1, 1.2, 1.3, 2.1, 3.1) 275
Figure 58: Scenario 1 Results 280
Figure 59: DTDM Network with Capacity Tariff (Scenario 2.2, 2.3) 283
Figure 60: Scenario 2 Results 286
Figure 61: DTDM Network with Ramp Tariff (Scenarios 3.2, 3.3, 4.2) 291
Figure 62: Scenario 3 Results 296
Figure 63: DTDM Network with Distributed Ramp Tariffs (Scenario 4.3) 301
Figure 64: Scenario 4 Results 306
Figure 65: DTDM Network with Dispatchable Storage (Scenarios 5.1, 5.2, 5.3) 311
Figure 66: Scenario 5 Results 314
Figure 67: Market Process Overview 321
Figure 68: Market Process Timeline (5-Minute Market Interval) 322

1 Introduction

1.1 Motivation and the Dynamic Distribution System

In “Transforming the Grid from the Distribution System Out,” a whitepaper published by the Wisconsin Energy Institute, a *dynamic distribution system* (DDS) concept is proposed [1]. The authors propose the DDS as an alternative to traditional, centralized electrical distribution systems, which were “not designed for the flexible load tracking required by renewables or the control of large numbers of distributed electrical energy resources.”

There are two key components to the proposed DDS concept. One, the DDS expands on the Consortium for Electric Reliability Technology Solutions (CERTS) microgrid protocols, enabling autonomous control of distributed energy resources without centralized information processing. Two, the DDS links both traditional and distributed electricity generation, storage, microgrids, and loads with a marketplace. In addition to these components, the DDS must be supported by new business models and regulatory policies.

The whitepaper goes on to claim that, in many use cases, the DDS could enable distributed energy resources to improve system reliability with decreased emissions, decreased power losses, and reduced energy prices. In particular, the DDS would be better equipped than centralized systems to support the intermittency and load swings inherent in distributed, renewable energy generation. The DDS concept provides benefits by incorporating the best aspects of both the centralized “Smart Grid” model and the fully distributed “Personal Power Plant” model (see Fig. 1).

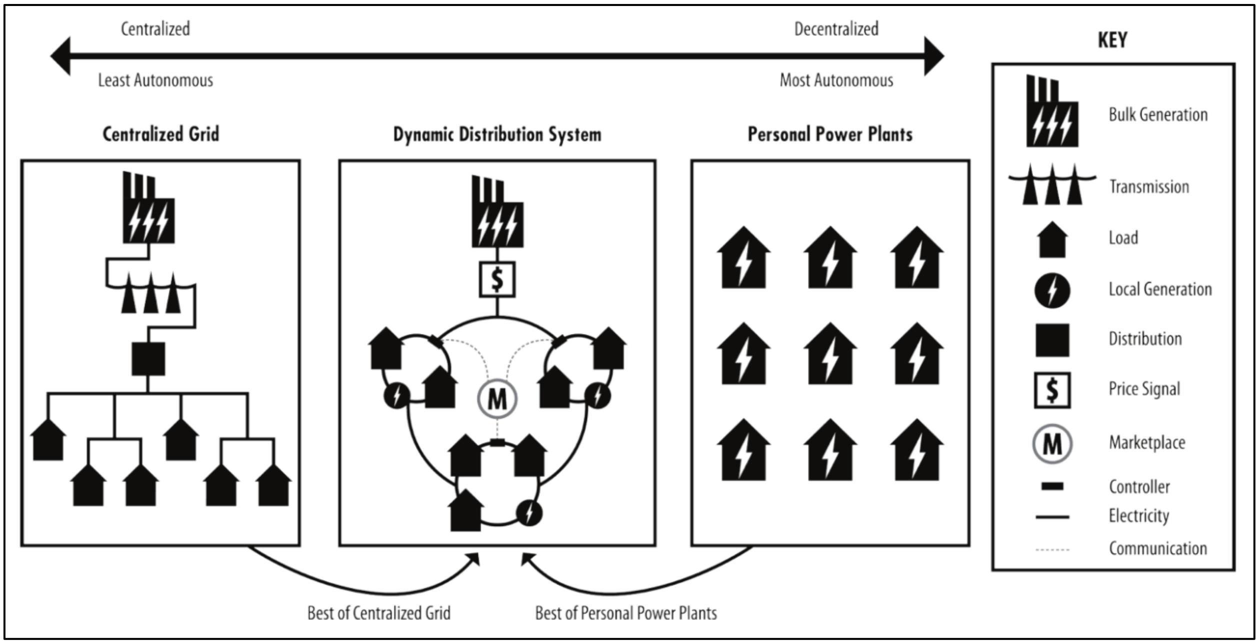


Figure 1: Dynamic Distribution System [1]

All four elements of DDS adoption are necessary for practical implementation: advanced power system controls, a dynamic marketplace, revised utility business models, and updated regulatory policies.

The first, power system controls, is a field of active research. Microgrids have applications beyond the DDS concept, and many research results can be applied to the DDS proposal. Similarly, utility business models and regulatory policies, closely related, are subject to analysis and consideration by research and industry organizations. However, these analyses are not necessarily focused on DDS adoption; instead, business models and regulatory policies often described general considerations or specific, non-DDS options.

This highlights the importance of specifying a proposed DDS marketplace. To provide for analysis of the business and regulatory requirements of a DDS, the marketplace rules and interactions must be defined. Further, the marketplace rules must be designed with consideration of power system control and monitoring capability, while ideally supporting enhanced capabilities of the power system control.

With this goal in mind, this thesis will consider a marketplace designed for implementation in a DDS. This marketplace must support the DDS concept, while enabling the other three elements of DDS adoption.

1.2 Statement of Problem and Research Questions

The Dynamic Distribution System concept does not specify a marketplace design. However the whitepaper specifies three key concepts [1]. One, the marketplace should enable energy transactions at a local, distribution level. Two, the marketplace should provide an equitable marketplace, enabling investment in distributed energy resources by any party. Three, the marketplace should facilitate intrinsic optimization. Otherwise, the marketplace design is left unspecified. This provides the primary problem statement for this thesis:

How should an energy marketplace be designed to support a Dynamic Distribution System?

Notice, this problem statement will not have a definitive answer. There are many marketplace designs that could support a DDS. This leads to two immediate follow-up research questions:

What are the marketplace goals and design principles?

In what ways can a marketplace design be evaluated?

Already, the DDS concept has provided some guidance on these questions. Specifically, the marketplace should enable energy transactions, provide an equitable marketplace for investment, and facilitate intrinsic optimization. Additionally, the marketplace should support the other three components of DDS deployment: power system controls, revised utility business models, and updated regulatory policies.

A distinction must be noted between power system control and the distribution energy marketplace. In supporting the DDS, the marketplace enables energy transactions. Power and energy are intrinsically related; however, the energy transactions will be on a longer timescale than the sub-second power dynamics. This does not mean that the marketplace will not indirectly support power system control; indeed, that is one of its goals. However, power system control is not the primary role of the marketplace. Additionally, this does not preclude the energy marketplace from including or supporting ancillary markets to directly support power control.

With marketplace evaluation criteria, goals, and design principles established, a market design can be proposed. This market design must be described, providing another series of research questions:

For a proposed market design, what are the marketplace rules?

For a proposed market design, how are externalities described and captured?

For a proposed market design, how do actors participate in the market?

In this context, the marketplace rules are the processes, restrictions, and protocols that define the marketplace. Externalities must be described and captured, to provide an equitable marketplace and incentives for market entry and distributed energy resource deployment. Additionally, the method in which market participants interact must be clearly defined, so stakeholder analysis can be completed for use in business model and regulatory assessment.

A proposed market design should then be evaluated based on its ability to meet the evaluation criteria, goals, and design principles. Again, any market design will have its own strengths and weaknesses. It cannot be expected that a proposed design is definitively successful in all cases. Instead, a proposed market design should be considered as a series of trade-offs, with analysis of advantages and limitations. This is the approach that will be taken in proposing and evaluating the market design in this thesis.

1.3 Technical Approach

From the research questions listed above, the technical approach is determined. This thesis ultimately proposes a set of market rules for use in a DDS. The market rules are collectively referred to as the Dynamic Tariff Distribution Marketplace (DTDM).

Section 2 provides a literature review for background information and insights on the design decisions used when developing the DTDM. The literature review will focus on the five following areas. First, distributed energy resources (DER) will be examined. This includes the definition, capabilities, and the positive and negative externalities of deployment. This will clarify the goals and challenges of the DDS and its marketplace. Second, the impact of DER on utility business models will be considered. This includes general approaches industry groups are considering as the penetration of DER increases. This will provide an understanding of where the DDS fits into existing proposals, and the subsequent evaluation criteria of its marketplace. Third, existing wholesale electric energy markets will be examined. This will provide lessons for use in the development of the DDS marketplace. Finally, studies on the dynamic pricing of retail electric energy will be examined. This will shape expectations on how actors will participate in the DDS marketplace.

In Section 3, the proposed Dynamic Tariff Distribution Marketplace will be described in detail. In summary, the DTDM determines minute-scale clearing energy; incorporates autonomous, independent actors; and establishes differentiated prices based on energy, losses, and externalities. The fundamental market process is a single-price double auction, although the market price will be adjusted as it propagates through the network. The DTDM, as proposed, applies to a single phase of the electrical system. The DTDM accomplishes this process by representing the physical network, sources, and loads as nodes connected by linkages. To capture externalities, tariffs are imposed on linkages between nodes. The marketplace process consists of three concurrent steps: Market Operation, Real-Time Actions, and Settlement.

The DTDM goals include supporting interconnection with a larger energy system; providing intra-system control capabilities, for use by a system operator; quantification and cost recovery of externalities; and support for islanding operation. Market design guiding principles include indirect control, scalability, flexibility, and efficient communication. This section will provide the general approach of the DTDM, but not the technical implementation.

After describing the DTDM proposal, this thesis will outline the process of simulating the DTDM. It is important to note that the DTDM proposal itself is a result of the recursive process of modeling, simulation, and market rule development. This thesis should not be interpreted as a sequential process of market rule development followed by simulation. Instead, a majority of the research effort was spent iteratively refining market rules and implementing them in a simulation model.

In Section 4, the data sets used in the simulation are described. This includes data obtained from Pecan Street Inc. Dataport, consisting of minute-by-minute electrical energy consumption and PV generation. In addition to using this data set as a basis of simulation, some observations can be made on the application of a DTDM in real-world residential setting. For the simulation, wholesale spot pricing is used from the New York Independent System Operator (NY ISO). This provides historical time-varying energy prices.

In Section 5, the simulation of the marketplace process is described. The simulation is done entirely in Matlab. The marketplace model includes communication processes, the methods for setting the DTDM network and parameter, and market rules. After initialization, the simulation performs market operation, real-time actions, and settlement for each timestep in the prescribed duration. The Matlab simulation also includes validation and analysis functions. In describing the Matlab simulation, many observations are made that apply to a real-world implementation of the DTDM. In particular, this provides insights into parameter limitations and communication requirements.

In Section 6, behavior models and proposed tariff designs are presented. Section 5 was limited to the Matlab simulation and general implementation considerations, providing the method of modeling a given DTDM network. However, the actors within the DTDM require behavior models to represent their actions when participating in this market. Three behavior models are presented, each with variations and observations: Load Nodes, PV Nodes, and Storage Nodes. Additionally, the DTDM rules provide a system operator or market participant freedom in implementing tariffs, to capture externalities. In Section 6, the tariff process is defined generally, to provide flexibility in implementation. In this section, five specific tariff designs are proposed and examined: capacity, target quantity, ramping (SMA), ramping (EMA), and flat-rate.

The technical approach described in Sections 5 and 6 does not provide detailed power flow analysis and does not purport to ensure dynamic system stability. Instead, this technical approach provides a fundamental basis for analysis of market interactions between actors within the system, including simplified estimates of the resulting energy flow.

In Section 7, case studies are presented and discussed. The purpose of the cases studies is solely to examine the marketplace process and demonstrate interactions between actors. The case studies are not intended to be a proof of results for actual deployment of the DTDM in a DDS. However, observations will be made that should be considered in any practical implementation of the DTDM. Case studies include the impact of load price elasticity, capacity tariff impact, ramping tariff impact, and impact of tariff locations.

Finally, Section 8 provides a summary and areas for future research. As stated, the DTDM proposal was developed for implementation within a Dynamic Distribution System. To that end, the DTDM provides many opportunities for further consideration and development, to better enable deployment in a DDS. Each of the following areas will be discussed: improved modeling flexibility, physical implementation, improved simulation behavior models, advanced market rules, system operator optimization, and utility business model analysis.

2 Literature Review

2.1 Overview

The demand for Distributed Energy Resources (DER) has increased in recent years. This can be attributed to many causes: end-use customers desiring to reduce their environmental impact; the falling cost of solar photovoltaic (PV) increasing economic use-cases; government subsidization of DER costs; and public policy initiatives to reduce the fuel mix of electrical generation.

First, DER will be defined. Specific DER devices and technologies will be outlined, following by general positive and negative impacts of DER deployment. These externalities will provide a key component for configuring incentives in a proposed DDS marketplace. Next, the impact of DER on utility business models will be discussed. This includes proposed approaches to adapt or replace traditional business models, to accommodate increased DER penetration. From these proposals, the Dynamic Distribution System (DDS) will be classified and better defined.

Next, existing electric energy wholesale markets will be considered. This will provide lessons for use in developing a marketplace to support the DDS. The DDS marketplace, by necessity, includes customer responses to price signals. Finally, attempts at dynamic electricity retail pricing will be examined, along with general price elasticity considerations.

2.2 Distributed Energy Resources

First, it is necessary to define the term “Distributed Energy Resources (DER),” as it can have slight differences in varying contexts. The Rocky Mountain Institute defines DER as “demand- and supply-side resources that can be deployed throughout an electric distribution system to meet the energy and reliability needs of the customers served by that system" [2]. RMI further clarifies that DER can be implemented on either the utility- or customer-side of the meter. Their definition includes generation, managed loads, energy storage, and any technology that provides energy, load management, or ancillary services. Ancillary services include, but are not limited to, energy reserves, voltage control, reactive power compensation, and black start capabilities. By this definition, DER includes demand management devices that adjust a customer's demand based on external signals.

Further, distributed generation (DG) is defined solely as generation sources that are connected on the customer-side of the meter. DG sources include photovoltaic (PV) solar, traditional internal combustion generators, and combined heat and power (CHP) generators. From this definition, all DG is considered DER, but DER includes many more technologies than DG. This distinction is useful, as it allows DER to encompass many complementary technologies. This paper will use the RMI definition of DER.

This broad definition of DER is useful for framing the DDS marketplace. A marketplace must be general and flexible enough to support all manners of DER. For example, if a proposed marketplace only supports supply-side DER then it may preclude actors in the system from investing in demand-side DER.

With such an expansive definition of DER, it is not difficult to imagine potential benefits DER could provide to the grid. Similarly, it is not difficult to image in the potential costs DER could impose upon the grid. These impacts range across time scales, from years to seconds, and across the grid value chain, from generation to distribution. In Figure 2 below, the Rocky Mountain Institute provides impacts DG can have across these domains. Note, this is limited to DG, so the impacts of DER can be considers more expansive.

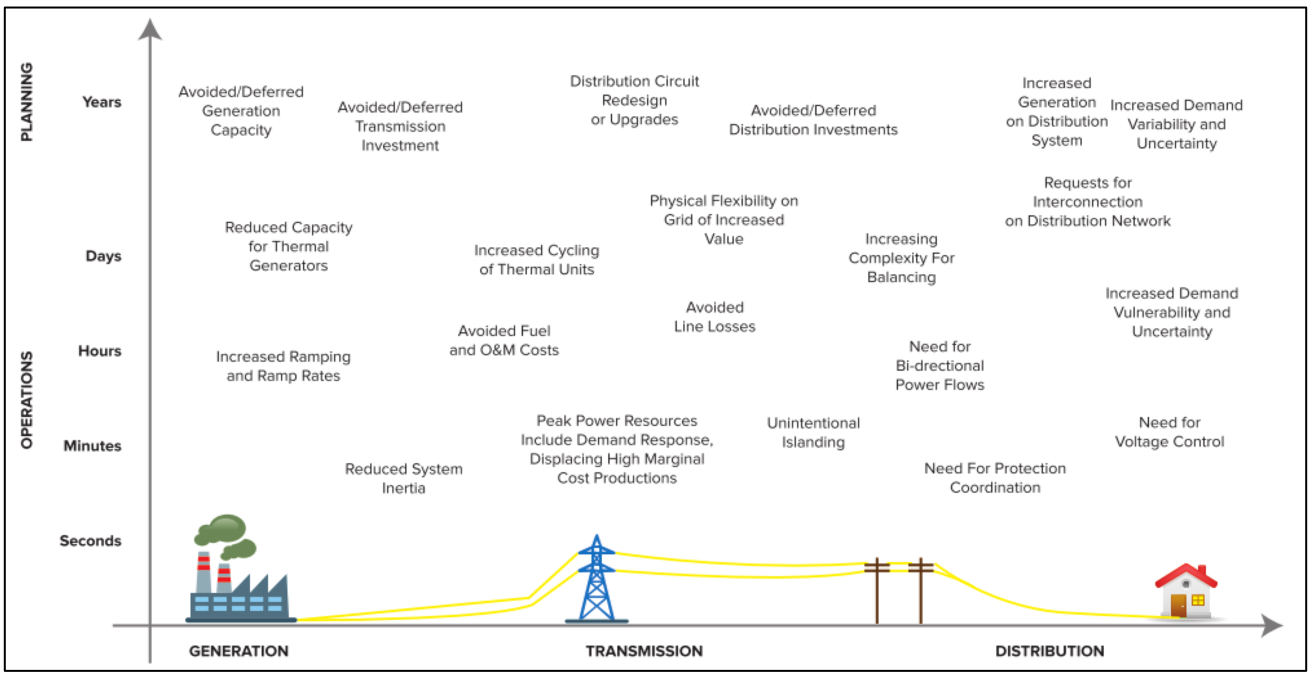


Figure 2: Positive and Negative Impacts of DG on the Electricity Value Chain [2]

The time scale for these impact ranges from seconds to years. A DDS marketplace may better equipped to take benefit from the advantages (and mitigate disadvantages) on the minute to hour timescale. At the second and sub-second timescale, the market will not have time to act; this is better suited to the power control component of the DDS. Alternatively, the actual market action is much too fast to have a direct impact at the year timescale; this is better suited to the business model and regulatory components of the DDS. That being said, a well-designed DDS marketplace should support power control actions at the faster timescale and business and regulatory actions at the slower timescale.

In the same report from RMI, their panel focused on five potential sources of value: displacing conventional generation, to include new use cases for onsite Combined Heat and Power (CHP) units; reducing line losses, in particular during periods of high demand; reducing transmission investment, such as those supporting congested, high-cost areas; reducing financial risk associated with larger scale investments, by providing smaller incremental capacities; and deferring or reducing grid investment, again particularly for congested, high-cost areas.

These sources of value provide some incentives that a DDS marketplace should enable. One, the marketplace should support cost-benefit analysis for local generation. Two, the marketplace should capture the impact of line losses on the cost of energy. Three, the marketplace should reflect congestion and, generally, externalities, faced by the larger energy grid and the local distribution network. And four, the marketplace should support local, focused investment decisions.

In addition to these sources of value, DER also poses some disadvantages when deployed in high penetration.

For example, a high penetration of distributed PV can lead to high ramp rate requirements. This impact is best illustrated by the California Independent System Operator (CA ISO) "duck curve", shown in Figure 3 below. This shows the anticipated diurnal cycle of net energy demand, defined as forecasted energy load minus non-dispatchable generation (i.e. wind and solar) [3]. As more PV generation is added to the system, the impact of sunset is more pronounced. Ultimately, by 2020, it is shown that CA ISO must bring approximately 13 GW of generation online in three hours. Accomplishing this task requires flexibility and coordination of many large-scale power plants.

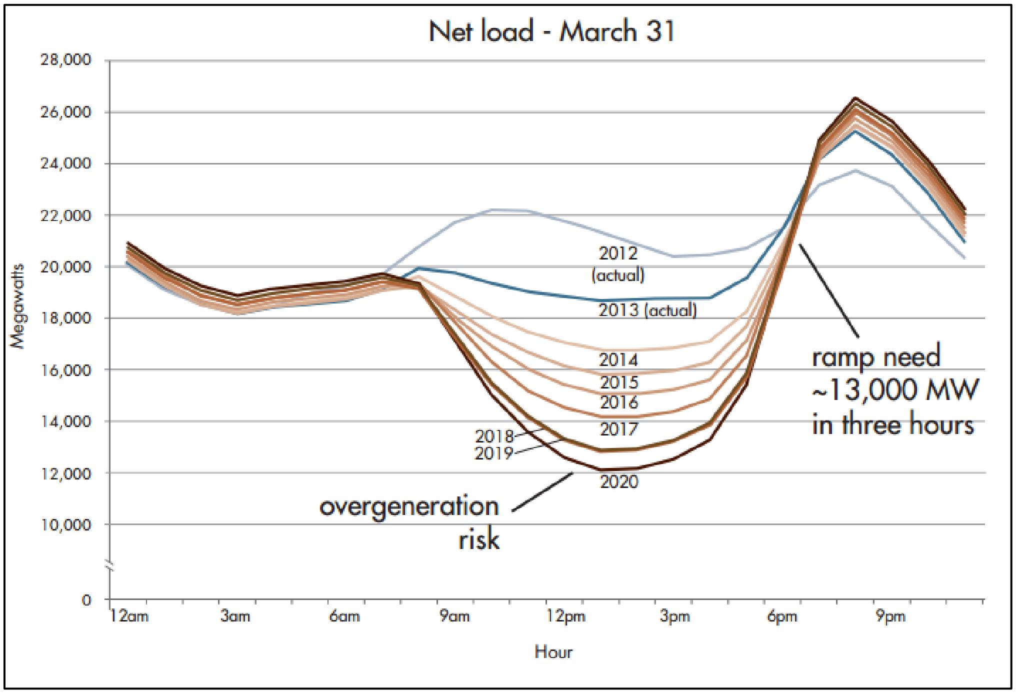


Figure 3: CA ISO "Duck Curve" [3]

Note, this illustration relates to the impact of non-dispatchable storage, which includes non-DER utility-scale renewable generation. However, the illustration is still valuable. From the perspective of CA ISO, the step ramp rate is a disadvantage of PV generation that is not necessarily captured by the direct temporal price of energy. The ramp rate is an *externality*.

Like the duck curve's ramp rate, many of the disadvantages of high DER penetration are externalities. High penetration of distributed PV adds to the grid's overall ramp rate, although perhaps not to the same extent as the duck curve; this is an externality.

A related externality of PV penetration is the additional balancing reserves required for regulation and load following. The RMI report cites the results of a 2011 Nevada Energy study that quantified the cost of this externality at $0-$0.005/kWh for ramping and $0.003-$0.008/kWh for additional regulation units [2].

Figure 2 provides some examples of additional externalities. For example, supporting bi-directional flow may require upgrades to infrastructure. In the meantime, this limits the deployment of energy exports. As a result, if one customer installs PV generation, it consequently limits other from installing PV generation. This capacity limit is an externality.

The impacts of DER on the grid, both positive and negative, rely on its location, magnitude, and timing. Additionally, the value of DER can be improved with flexibility, predictability, and controllability. As the Rocky Mountain Institute highlights [2], capacity investments are "lumpy". To enable deferment of transmission upgrades or centralized generation plants, the DER must meet the necessary demand reduction capacity. As a result, the value of DER is both highly variable and non-linear. The variation and non-linearity of DER impact makes it difficult to apply traditional cost-benefit analysis methods.

A DDS marketplace should seek to mitigate or eliminate cost of DER deployment. However, when a disadvantage cannot be eliminated, the DDS marketplace would ideally pass along the disadvantage's cost to the correct party. In other words, the system actor that causes a negative externality should bear the costs of that externality. By extension, this concept should also apply for all positive benefits of DER. This is a distributed alternative to a centralized cost-benefit analysis method.

Concisely: a DDS marketplace should seek to pass along the costs and benefits imposed by externalities to the parties responsible for causing the externalities.

2.3 DER Impact on Utility Business Models

One large set of DER externalities is the impact on utility business models. Consideration of this impact must be addressed by the business model aspect of the DDS proposal. However, it will be discussed here, as the DDS marketplace must support a feasible DDS business model.

In 2012, the Rocky Mountain Institute (RMI) organized a roundtable discussion to examine how the relationship between utilities and their customers will change with increased in DER penetration. Among their conclusions is the recognition that "existing utility rate structures and business models, which have evolved over time to meet a complex set of policy and economic goals, are poorly adapted to this new environment" [2].

2.3.1 Traditional Business Models and Volumetric Rates

RMI's expert panel went on to specifically acknowledge that current volumetric rate structures, when coupled with Net Energy Metering (NEM) rates, are not sustainable as utilities' long-term business model [2]. To understand this conclusion, *volumetric rates* and *net energy metering* must be defined. This provides an overview of many business models in place today and provides an opportunity to discuss the impact of DER on those business models.

Utilities purchase energy on the Wholesale Market, which is operated by the region's Independent System Operator (ISO) or Regional Transmission Organization (RTO). In turn, the utility sells energy in their Retail Market, which consists of end-use customers. The revenue generated in retail energy sales must cover the utility's wholesale energy purchases and their "Costs of Service," which includes operating costs and overhead.

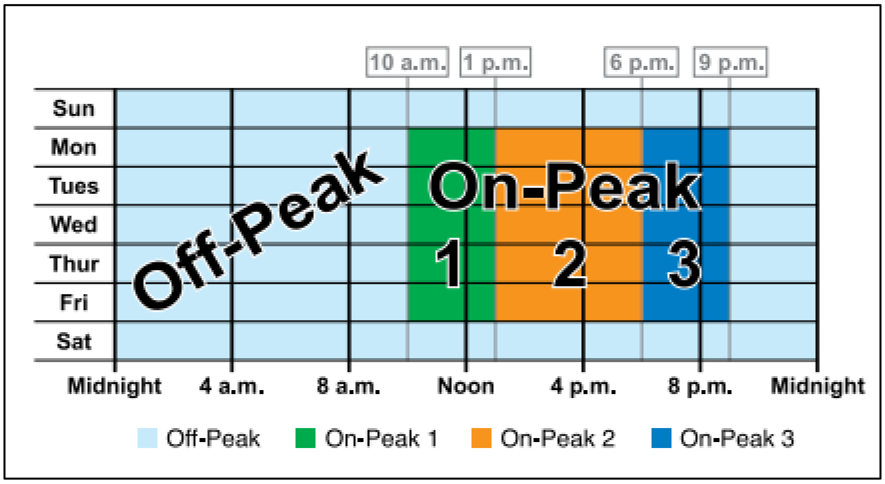
It is important to note that the unit cost of energy purchased on the wholesale market varies throughout the day, as different generation sources export energy into the larger electrical grid. This will be described further in Sections 2.4 and 4.2.

From the energy purchases and "costs of service" revenue requirements, the retail price of electrical energy can be broken into variable costs and fixed costs. Variables costs are a function of the volume of energy consumed. Variable costs include generator fuel and variable generation operations and maintenance (O&M) costs. Fixed costs comprise the capital investment required to transport the electrical energy, as well as overhead required to service end-use customers. Fixed costs are generally a function of the peak demand of energy, accounting for both the timing and magnitude [2].

Retail billing by utilities traditionally includes three categories: energy charge, demand charge, and customer charge. Energy charge is attached to the volume of energy provided. Demand charge is attached to the peak demand in the billing period. Customer charge is a fixed amount covering the cost to service the customer, regardless of the volume or peak demand. Traditionally, utilities achieve a majority of their revenue requirements through the energy charge, as a bundled volumetric rate [2].

A bundled volumetric rate allocates variable and fixed costs into a unit price for energy. As a reference point, the average price of electricity to ultimate customers in 2014 was $0.1252/kWh for residential customers and $0.1045/kWh for all use sectors [4].

The volumetric rate is established in advance of customer consumption, often a year in advance. The volumetric rate may be constant over the year, or it may change with the seasons. Additionally, a volumetric rate may shift over the course of the day; this is a Time-of-Use (TOU) rate structure. For example, the following TOU rate structure is offered by Madison Gas & Electric [5]



Charges during Winter Season (Oct. 1-May 31) per kWh

Time Period	Distribution Service	Base Energy All kWh	On-Peak	Total/kWh
Off-peak	$0.03425	$0.04127	N/A	$0.07552
On-peak 1	$0.03425	$0.04127	$0.16021	$0.23573
On-peak 2	$0.03425	$0.04127	$0.16021	$0.23573
On-peak 3	$0.03425	$0.04127	$0.16021	$0.23573

Charges during Summer Season (June 1-Sept. 30) per kWh

Time Period	Distribution Service	Base Energy All kWh	On-Peak	Total/kWh
Off-peak	$0.03425	$0.04127	N/A	$0.07552
On-peak 1	$0.03425	$0.04127	$0.18763	$0.26315
On-peak 2	$0.03425	$0.04127	$0.20988	$0.28540
On-peak 3	$0.03425	$0.04127	$0.18763	$0.26315

Figure 4: Madison Gas & Electric TOU Rate Structure [5]

The TOU rate structure attempts incentivize customers to shift energy consumption away from period of peak demand. This avoids the higher wholesale energy prices observed during peak demand, as well as serves to mitigate the risk of overloading T&D capacity limits.

The Madison Gas & Electric rate structure is voluntary for residential customer. This is a common for most TOU residential rate structures. Because TOU rates are generally voluntary, they cannot be considered the baseline utility business model. Regardless, TOU are still volumetric, because they are pre-established and asses a fixed cost per unit of energy.

Customer demand for DER has disrupted this traditional, volumetric retail rate structure. With DER, specifically DG, utilities are faced with customers that wish to export unused energy to the grid. The most common approach to this challenge is Net Energy Metering (NEM). Under NEM, customers are credited for energy exports at a volumetric rate. In some cases, this rate is less than the retail rate for energy purchases; however, “full retail NEM” is extremely common. NEM benefits from its simplicity, which has led to its adoption in 43 states [2].

There are three relevant consequences of NEM programs. One, under volumetric rate structures with NEM, customers receive price signals that do not reflect the system costs and benefits of electric energy. This is conclusion is consistent with the discussion on externalities provided in the previous section: there are time- and location-specific externalities associated with all DER. Two, under these rate structures, customers are incentivized to maximize their value (i.e. total energy generated with PV), not value for the system. Three, distributed PV can “cost-shift” from PV adopters to non-adopters, leading to inconsistent rate structures.

As an alternative to NEM volumetric rate structures, a DDS market should overcome these consequences.

Note, although NEM has been highlighted, it is not the only approach to accommodating DER. Other approaches include feed-in tariffs and tiered rate structures. This thesis does not provide a detailed evaluation of these approaches; instead the general conflict between DER adoption and traditional rate structures is used to motivate the development of the DDS market design.

The adoption of DG by customers has also impacted standard load profile assumptions. When energy exports are accepted, the most extreme cases are Zero Net Energy (ZNE) customers. In these cases, a customer exports a quantity of energy equal to their grid imports. Most commonly, the customer's exports result from daytime PV generation that exceeds their energy demand.

ZNE facilities, also referred to as Net Zero Energy Buildings (NZEB), are an increasing target in the "green construction" industry, garnering attention for organizations such as the US Green Building Council and the International Living Future Institute. However, when coupled with full retail NEM, ZNE customers disrupt the collection of fixed costs within volumetric rate structures. As shown in Figure 5 below, a customer has the potential to "zero out" their bill, despite maintaining a non-zero cost to serve.

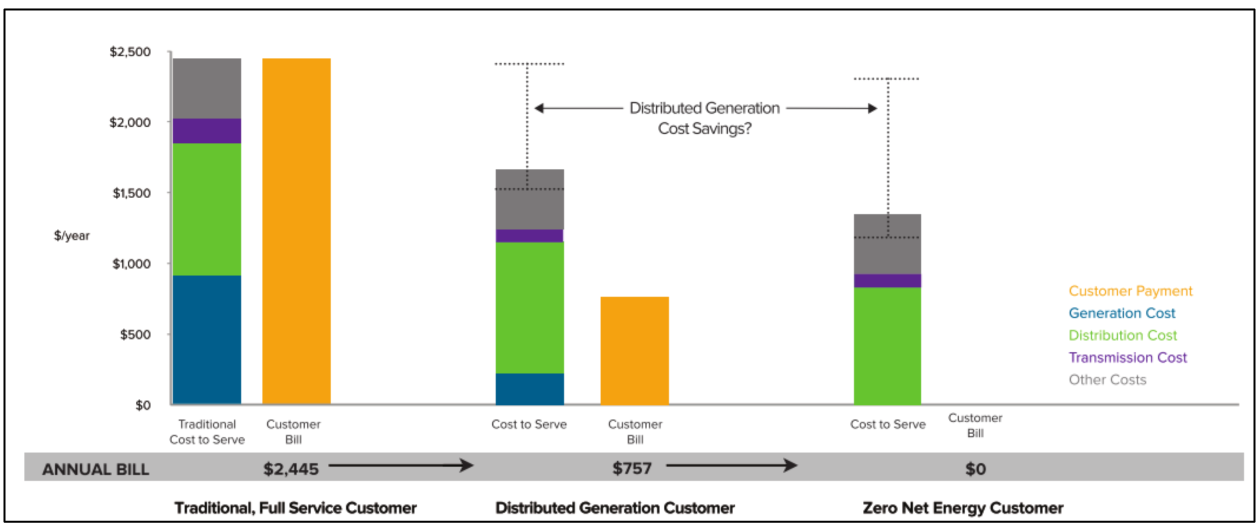


Figure 5: The Potential Impact of NEM and ZNE Customers on Capturing Fixed Costs [2]

This discrepancy between customer payments and cost to serve is a direct result of coupling fixed costs to bulk volumetric rates. RMI categories this impact into three "misalignments": cost allocation, in which customers are not appropriate charge or paid for the services they require or provide; equity, in which different customers are faced with different effective pricing; and operations, in which the grid's operational requirements are not adequately valued [2].

The equity misalignment, in particular, has led to industry consideration of the "utility death spiral". In this vision, DG adoption forces utilities to increase volumetric rates to recoup lost fixed costs. Customers without DG observe these higher rates, recognize improved DG rates of return, and, naturally, DG is adopted by more customers. Again, this leads the utility unable to recoup all fixed costs; the utility is forced to increase the volumetric rate yet again. The cycle continues, ultimately punishing customers unable to adopt DG, forcing the utility into a precarious financial situation, and disrupting the grid's future stability.

The scenario suggested by the "utility death spiral" illustrates only one concern with increased DER penetration under traditional retail business models. From a business model perspective, an improved distribution system model would address these three misalignments: cost allocation, equity, and operations.

This provides a set of goals for a DDS marketplace: costs must be properly allocated, customers should face equitable effective pricing, and the grid's operational requirements must be valued.

Based on these observations, adjustments to utility business models are required to support increased DER penetration. The challenges and misalignments with current business models inform the approach to business model changes.

RMI proposes that rate design "should ensure two things: that utilities receive adequate compensation to cover their prudent costs, and that those costs are distributed among customers equitably, so a customer's electricity bill is representative of the value of the services provided to, and by, that customer" [2]. When DER is introduced, existing rates for energy, demand, and customer charges may not meet these requirements. This would necessitate rate structure changes, to better align costs and revenues.

RMI then proposes evaluating rate structures on their ability to pay for operational services, capture and promote value to the system, and be implemented with flexibility [2]. Additionally, any proposed change must be technologically and politically feasible. To this end, any proposed rate structure should be as simple as possible, to improve customer understanding and gain approval from the public service commission.

These two criterion are useful in assessing a DDS marketplace proposal: a customer's electricity bill should reflect the value of services imported and exported by the customer; a rate structure should pay for services, promote value, and be flexible.

However, there are significant challenges faced by a utility when proposing a change to existing rate structures. There are many stakeholders in the retail electric energy sector, to include the public, utility shareholders, regulators. It is difficult to propose a change that provides benefits to all stakeholders. A proposed change by the San Diego Gas and Electric Company (SDG&E) illustrates these challenges.

In 2011, SDG&E proposed implementing a "Network Use Charge" on its residential customers [2]. This charge would specifically bill for use of the network, as a method to capture otherwise unrecognized costs associated with NEM households. As proposed, customers would face a charge tied to their absolute demand. In this construct, a customer exporting 1 kW (i.e. when PV generation exceeds household demand) is charged the same as a customer importing 1 kW (i.e. typical demand).

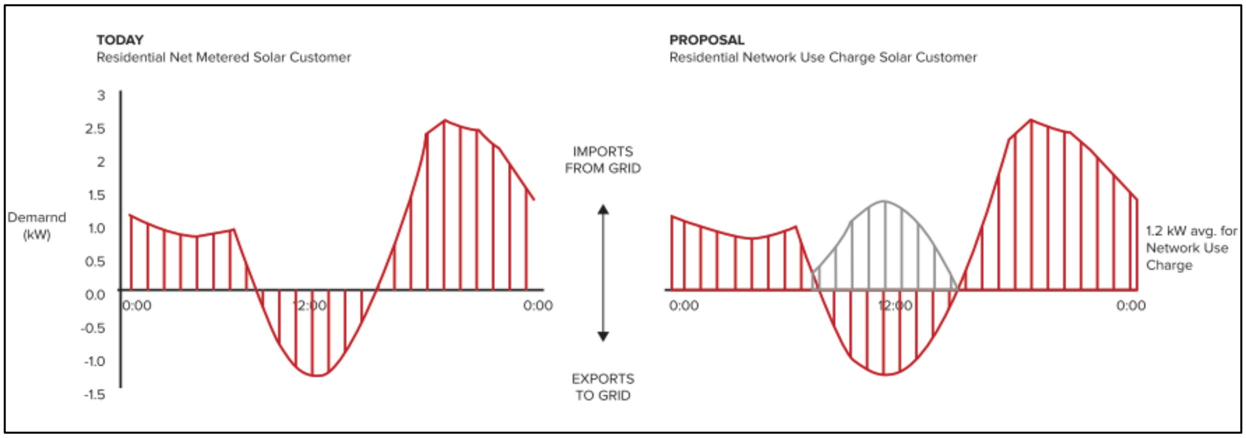


Figure 6: SDG&E's Proposed Network Use Charge [2]

This rate structure was specifically designed to ensure NEM customers pay their fair share of the costs associated with operating a distribution system, of which they are utilizing. It is worth noting this rate structure supplements, but does not replace, the traditional volumetric energy charge; customers would still be paid for exporting energy, albeit at a lower effective rate.

According to the RMI criteria, the SDG&E proposal could be consider an effective rate structure. It pays for operational services, by capturing the fixed costs associated with distributing both energy imports and exports. It captures and promotes value to the system, by recognizing high export quantities may drive up the costs of operating the distribution system. It can be implemented with flexibility, as the calculation formula is straightforward and can be scaled as needed. However, this proposal was met with strong opposition from groups including the solar industry, customer advocates, environmentalists, and NEM customers [2].

They argued the SDG&E proposal failed to capture benefits of DG, discouraged meeting California's renewable energy goals, and would result in price signals that do not reduce peak system demand. The highly variable costs and benefits of DER ensure that these claims are certainly true in many cases. However, the costs SDG&E attempted to capture in their proposal are also certainly valid in many cases. It is extremely difficult to describe these costs and benefits within a simple rate structure to be applied to all use cases. The strong opposition also illustrates some requirements when engaging with stakeholders.

Ideally, a rate structure proposal would benefit all stakeholders. However, it cannot be assumed this is always possible; so, at a minimum, newly captured costs should be specific and defensible. In application, this may also runs counter to a goal of simple, understandable rate structures.

This is a lesson for the design of a DDS marketplace: the marketplace should have simple, understandable rules with specific, defensible cost allocations.

Despite this difficulty, customer demand for DER continues and utilities are faced with integrating these resources into the grid. To address this challenge, the RMI panel considered alternative utility business models for a future with a higher penetration of DER and an increased customer focus on energy efficiency. The key question asked was "What are the functions that utilities will perform in the future and how should we create mechanisms to appropriately compensate utility companies for performing those functions?" [2].

As expected, there are many possible approaches to this question with even more permutations of utility business models. However, proposals can generally be separated into two approaches, which are not necessarily mutually exclusive: *Incentive Regulation* and *Network Utility*.

2.3.2 The Incentive Regulation Approach

The first category is the *Incentive Regulation* approach, in which the utility has a direct role in financing, deploying, and/or operating DER [2]. This is encouraged by regulator-approved revenue structures such as performance based earnings, shared savings, incentive rates-of-return, and DER deployment targets. This approach could parallel attempts to promote energy efficiency by decoupling utility revenues from total energy sales. The *Incentive Regulation* approach further establishes the electric utility as the primary driver in distribution operations; RMI refers to their role as a “network orchestrator and service provider”.

This approach would attempt deploy DER for the greatest value by leveraging the utility’s institutional knowledge. Additionally, the utility’s responsibility to serve all customers fairly remains intact, strongly limiting the possibility of customer inequalities. However, this approach requires regulatory changes and modified rate structures to support utility revenue requirements. With the stakeholder incentives previously described, this is not an insignificant challenge. Finally, this approach may necessitate the utility becoming more involved in customers’ behavior behind-the-meter, for example when selecting rooftop PV deployment locations.

Within the *Incentive Regulation* construct, there are multiple valid approaches. Again, an expert panel convened by the Rocky Mountain Institute provides insight into some of these approaches [6]. In their report “New Business Models for the Distribution Edge,” RMI proposes three potential utility business models. The first two, Integrated Distributed Resource Manager and Distributed Resource Finance Aggregator, can be categorized within the *Incentive Regulation* approach.

As an Integrated Distributed Resource Manager, the utility take responsibility for planning the deployment of DER. This is accomplished with fully transparent cost-benefit analysis and a goal of maximizing DER impact at the lowest cost. Incentive regulation would provide the utility revenue for acting within this role. During their cost-benefit analysis, the utility would welcome input from third parties and make its decisions in a peer-reviewed fashion. This approach would be relatively simple to implement, as it closely matches existing roles of the utility. However, the proposed goal of transparent and clear cost-benefit analysis may be difficult to attain; centralized decision making will inevitably lead to disagreement between stakeholders.

Alternatively, as a Distributed Resource Finance Aggregator, the utility takes an active role in managing the financing of DER. The utility enacts new tariffs, customized for customers participating in DER investment. Customers would engage with third-party “preferred service provider installers” to implement the DER investment. This provides a new revenue stream for utilities: facilitating business between customers and these third parties. Consequently, the third-party service providers are compensated by the utility for installing and managing the DER. This option is also attractive because it can be easily integrated into existing utility structures. However, RMI claims such a flow-through financing model could be difficult to implement.

2.3.3 The Network Utility Approach

Alternatively, the second category proposed by the Rocky Mountain Institute is a *Network Utility* approach, in which the utility provides customers with prices signals designed to incentivize favorable network behavior [2]. Instead of direct utility involvement, these price signals will induce investment by other service providers. In this approach, the utility’s role would include continued O&M of the grid, with the added responsibilities of creating markets, managing additional transactions, and interconnecting buyers and sellers. This would best parallel the current role of ISOs and RTOs at the wholesale level, but instead at the distribution and end-use level. Markets within the utility network would encourage optimal DER deployment; a well-designed market would include both incentivizing and dis-incentivizing price signals.

The *Network Utility* approach could provide the utility with new sources of revenue unconnected to volumetric energy sales, such as network operation and network facilitation. However, this approach is not without challenges. The exact nature of utility responsibility and revenues must be established. For the utility's new role as a market facilitator, regulatory benchmarks for performance must be established and enforced. Additionally, differentiated price signals may disrupt the traditional utility responsibility of customer equality. The regulatory implementation of a *Network Utility* approach should address the possibility of energy "haves" and "have-nots". Finally, this approach would require the creation of new platforms and protocols to take advantage of these differentiated price signals, perhaps making this a more significant shift than adopting the *Incentive Regulation* approach.

In a proposed business model described by RMI in "New Business Models for the Distribution Edge," the utility acts as an Independent Distribution Network Operator [6]. This can be classified within the *Network Utility* approach. As an Independent Distribution Network Operator, the physical distribution network remains a regulated utility with performance-based regulation. However, the role of the distribution utility now includes reducing system costs through pricing mechanisms. Not surprisingly, this parallels the role ISOs play in the wholesale market. In this aspect of the business model, the utility would be rewarded with regulatory incentives, such as "shared savings". RMI anticipates this model would result in some form of locational marginal pricing throughout the distribution network. Of the business models suggested by RMI in this whitepaper, this is suggested to support the highest levels of innovation in DER integration. By decentralizing the adoption of DER, the utility will enable creative use cases to maximize local value. However, this requires far-reaching structural changes, for regulators, utility, and end-use customers. One challenge is designing a market structure to captures all positive and negative aspects of DER and energy consumption. Another is revising the legal framework between actors within the distribution system, such as assigning responsibility and liability for the consequences of dynamic electrical characteristics.

From these two categories, the Dynamic Distribution System concept fits within this *Network Utility* approach. Under this approach, the DDS will allow many independent actors to pursue and capture local value. Prices within the DDS would describe the system value of energy and dynamic characteristics, exposing actors to new costs and opportunities. DER adoption would be incentivized by these prices, but ultimately dictated by the independent actors within the system. This observation is useful when framing the DDS proposal within existing framework proposals. Note the DDS concept, and this thesis, does not specify the business entity that functions as the Distribution System Operator (DSO). This could be the utility, to parallel the role described by RMI as the Independent Distribution Network Operator. Alternatively, this could be a new business entity altogether. This determination falls to the business model and regulatory aspects of the DDS concept and will not be addressed further in this thesis.

2.4 Electrical Energy Wholesale Markets

It is valuable to consider the market processes in existing energy markets, when developing a DDS marketplace. To this end, the design and operation of the wholesale electric energy market is considered, with many relevant conclusions. The primary resource for this research is Steven Stoft's "Power System Economics" [7]. From this text, five particularly relevant observations are made. These observations focus on: marginal costs and market clearing, market structures, designing and testing market rules, forward and spot markets, and locational pricing.

2.4.1 Marginal Costs and Market Clearing

As defined by *The MIT Dictionary of Modern Economics*, the marginal cost is "the extra cost of producing an extra unit of output" [8]. In our context, the unit of output is energy, generally expressed as kWh at the distribution level. Stoft's text focuses on wholesale energy markets; thus the marginal cost is that of power plants participating in the market.

Stoft states that, in a competitive market, energy producers will supply energy up to the quantity at which their marginal cost equals the market price. This is the *market clearing price*, at which supply equals demand [7]. A supply curve is thus each energy producer's marginal costs at each available quantity and the system's aggregated supply curve is the horizontal summation of all curve. In a wholesale market, it is generally assumed that the demand is fixed for a given point in time. Thus the market clearing price at the aggregated supply curve's marginal cost indicated by the current energy demand.

This concept, of marginal costs and market clearing points, can be applied to the DDS marketplace. In the DDS marketplace, market participants can be considered consumers, not suppliers. Additionally, rather than the demand quantity being fixed, the supply price can be considered fixed at the wholesale energy price. Thus, the DDS marketplace clearing point would be the quantity at which the aggregated demand curve matches the wholesale energy price. This will influence the overall design of the DDS market proposal.

Additionally, Stoft also defines a "price taker" as market participants that "take the price as given when computing their profit-maximizing output quantity" [7]. These are actors without market power. This same concept can be applied to consumers in the DDS marketplace.

2.4.2 Market Structures

Next, Stoft described the market structures used by the wholesale electric energy markets. In general, markets arrange trades between buyers and sellers. Trades can be "bilateral," directly between one buyer and one seller, or the exchange can be "mediated," through an intermediary. There are variations within these two categories and overall market may include components of each [7]. Bilateral markets are extremely flexible, but the flexibility is expensive and time-consuming. Wholesale energy markets are commonly mediated, either as an "exchange" or "pool" type, although long-term energy markets generally use bilateral forward markets.

An exchange, or auction market, provides security for market participants by acting as the counter party on all trades. If a market includes independent operation by buyers and sellers, their actions are linked by a "double auction". Exchanges can operate at much faster speeds than bilateral markets, making them useful for real-time markets [7]. However, bids into the exchange are limited to energy quantities and price. This may lead to bidders manipulating, or "game," their bids, to capture non-marginal costs, such as start-up or no-load conditions. This is a downside to simple exchange markets.

A pool is a modification of an exchange that includes side payments. In this market configuration, bids into the market include more information than the marginal cost curve. This information may include generator limits and start-up costs. When the centralized coordinator determines the market clearing point, it may issue side payments to "make whole" bidders that would otherwise lose money [7]. These pools typically use very complex bids, with an attempt to prevent bidders from gaming their bids.

Based on these observations, a DDS market may best be designed as an exchange, providing speed and simplicity. However, the simplicity of bids may requires bidders to "game" their bids, to encompass their risk and true costs; this will need to be considered by the market design.

The extent of possible DDS market structures is illustrated with the EcoGrid EU project [9]. In this pilot program, household DER was controlled with real-time markets, much like the DDS concept. For their deployment, the structure was a "'bidless' market with price announcements ex-ante". The settlement price was determined and published based on the anticipated response of dynamic, household-level devices; in other words, households did not communicate consumption preferences to the system operator. This method was selected to reduce the transaction costs predicted to be associated with small actors (i.e. the households) participating in the market. While this is not strictly a double auction, it is certainly a valid market approach. In contrast to the EcoGrid EU approach, this thesis will specifically address a double auction approach, in which all actors participate in the market as both buyers and sellers.

2.4.3 Designing and Testing Market Rules

Following the description on market configurations, Stoft describes methods for designing and testing market rules. The author's observations apply to wholesale energy markets, but there are key observations that also apply to a DDS marketplace.

One, a market should be designed to prevent gaming. Efficient trading occurs when all bidders submit their true costs and values into the market; this leads to efficient outcomes. A market mechanism that induces truth-telling is an *incentive-compatible* mechanism. The author states that single-price auctions are nearly incentive compatible [7]. In a single-price auction, the same price applies to all buyers and sellers, regardless of their marginal costs or values. This is an alternative to pay-as-bid auctions, in which buyers and sellers receive prices based on their marginal values and costs.

From this observation, a DDS market may work best with a single-price auction configuration. Additionally, the market should be evaluated on how well it incentivizes market participants to express their true costs and values.

2.4.4 Forward and Spot Markets

The next relevant lessons from the wholesale energy markets are the distinction between forward and spot markets.

Forward markets trade in energy futures. This includes nonstandard, long-term forward contracts. Additionally, the system operator will generally hold forward markets one day prior – the day ahead (DA) market – and often one hour prior to real-time operation. These formal process are also forward markets [7].

The author limits his definition of spot markets to the real-time (RT) market, although it is acknowledged that the term spot market is often used to also describe the DA and hour-ahead market. The relationship between energy purchased and delivered in the DA and RT markets is called the two-settlement system.

In the two-settlement system, deviations from forward contracts are assessed at the RT market price. The largest take-away from this process is that it maintains the same performance incentives for actors within the RT market, regardless of the forward market quantities and prices [7]. As a result, actors behave as if all transactions are made solely in the RT market. This results from the observation that forward markets become sunk fixed costs once established and market behavior is determined by marginal costs and values.

This is an extremely useful conclusion for modeling the behavior of a DDS marketplace. Even if the DDS marketplace include provisions for forward markets, RT market behavior can be modeled as if the forward markets never took place. As a result, all modeling in this thesis will only provide RT markets, but this does not preclude the existence of a forward market.

As an example of the relationship between DA and RT markets, the NY ISO market timeline is provided below.

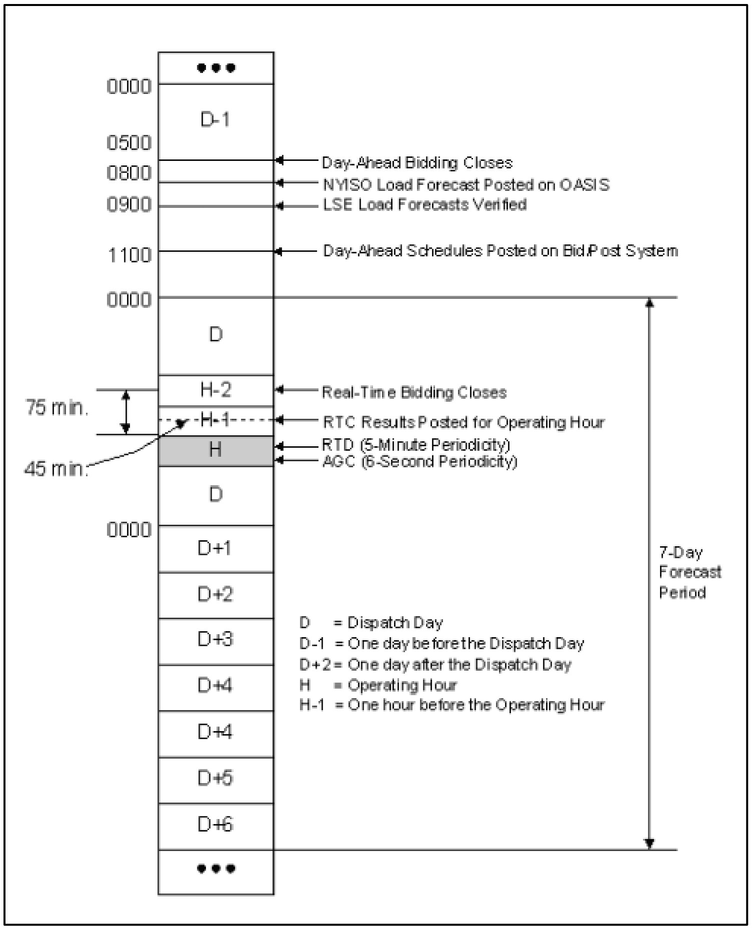


Figure 7: NY ISO Market Timeline [10]

2.4.5 Locational Pricing

Finally, wholesale markets provide insight on location-specific pricing. Up to this point, it has been assumed that the clearing price applies to all locations in the market. However, this is not the case. Pricing must reflect physical losses and may need to be adjusted due to capacity limits on system components.

Physical energy losses must be considered by the wholesale energy market. This can be accomplished through decentralized competitive loss pricing, which uses transmission markets, or through nodal loss pricing, which uses reference bus pricing [7]. Neither approach will be discussed here, but this illustrates the need of a DDS marketplace to consider system losses when determining prices.

Additionally, network capacity limits can prevent optimal market dispatch from being technically feasible. This drives *congestion pricing*, in which different locations in the network face different prices. These prices result in the optimal, economically efficient energy dispatch. Again, the details of this process will not be described, but this provides a precedent and foundation for location-specific prices, based on network limitations. A DDS marketplace could be expected to provide similarly differentiated price signals.

2.5 Dynamic Pricing and Price Elasticity of Demand

A DDS marketplace, by necessity, must incorporate customers' response to price signals. For insight on the expected customer response to participation in a DDS marketplace, the concept of dynamic pricing is examined. From this follows some general observations on the price elasticity of electricity demand and the role of hedging within a DDS marketplace.

Exposing customers to the volatility of the wholesale electricity market has been an on-going area of economic study for decades. This concept is described as dynamic pricing or, in some cases, real-time pricing (RTP). As described by Joskow and Wolfram [11], marginal generation costs fluctuate throughout the day: baseload power plants, with low marginal costs, are supplemented by power plants with higher marginal costs during periods of peak energy demand. When customers face retail rates that do not reflect this variation, they will consume too much when marginal costs are higher than retail rates and will consume too little when marginal costs are lower than retail rates.

Preventing this misallocation of consumption (i.e. seeking the economically efficient level of consumption) is the primary driver of dynamic pricing proposals.

Time-of-use (TOU) rates attempt to adjust for this discrepancy, but do not reflect the true marginal costs of real-time generation. These rates are set in advance, based on the anticipated generation costs for specific time periods; in practice, they only very roughly reflect the true marginal costs. Joskow and Wolfram list four traditional arguments against dynamic pricing: metering infrastructure would be too costly; complex rates would increase billing and metering costs; most customers would not understand or adapt to dynamic prices; and dynamic pricing would result in inequity of effective energy costs for customers with different consumption patterns [11]. They then claim that the first two claims are “largely irrelevant given current metering and billing technologies”. Further, pilot programs have shown that customers will adapt to dynamic pricing; although studies have been voluntary and limited, so they cannot necessarily be extrapolated to all customers. Finally, the authors recognize the potential for inequity as the largest impediment for wide-scale adoption of dynamic pricing. Numerous studies have attempted to model this impact, but results are not conclusive. As a result, it should be expected that dynamic pricing implementation would be cautious, with voluntary adoption by customers.

From these concerns, two more DDS marketplace design goals are identified.

One, the DDS marketplace, in support of the DDS as a whole, should seek to minimize the friction of adoption by customers. This is not a trivial requirement, yet it is unclear how one marketplace proposal fundamentally provides more or less friction than another proposal. Instead, this is primarily a concern for the DDS business model; the DDS marketplace should support the efforts of the business model’s customer experience goals.

Two, the DDS marketplace should provide opportunities for all stakeholders to benefit. As described by Joskow and Wolfram, when transitioning to dynamic pricing, there will likely be customers that higher effective energy prices. A DDS marketplace proposal need to address this directly. Instead, the marketplace should facilitate opportunities to seek and take advantage of new value propositions. The DDS marketplace should not be prescriptive: customers should have ultimate flexibility.

One important aspect of dynamic pricing is the anticipated response of customers to differing prices. This is described as the *price elasticity of demand*. Price elasticity of demand is defined as the percent change in quantity demanded, due to a one percent change in the price [8]. It is formally defined as:

$$\varepsilon = \frac{dQ/Q}{dP/P}$$

Generally, a given price elasticity of demand is negative. However, it is commonly expressed by its unsigned magnitude, although the actual negative value is presented in some publications. In this thesis, the unsigned magnitude will be used.

The price elasticity of demand is useful when illustrating a consumer's *demand curve*. A demand curve is a series of points representing desired consumption quantities – described by the horizontal axis – at various price points – described by the vertical axis. The price elasticity can range from perfectly inelastic to perfectly elastic. A perfectly inelastic demand (Ɛ = 0) does change quantities in response to price changes; this is represented by a vertical demand curve. Conversely, a perfectly elastic demand (Ɛ = ∞) can be represented as a horizontal demand curve, describing the logical limit of price sensitivity. For elastic demand between these extremes (0 < Ɛ < ∞), the quantity demanded will vary with the price; this is a descending demand curve. These cases are illustrated in Figure 8 below.

As a result of these elasticity limits, a demand curve is expected to be non-ascending. Note, there are goods in which the price elasticity of demand may be positive, resulting in an ascending demand curve; these are very special cases, and it is assumed electrical energy is never one of these goods.

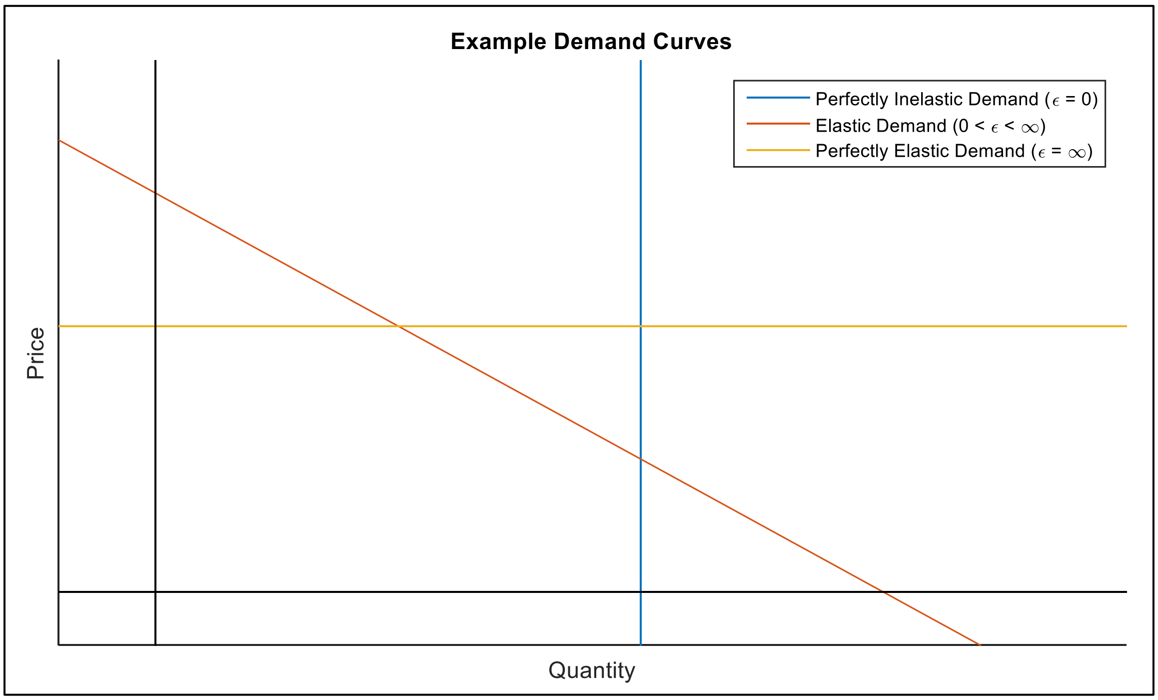


Figure 8: Impact of Elasticity on Demand Curves

Finally, a *supply curve* represents a producer's willingness to supply quantities of a product. A supply curve is also plotted on the quantity-price axes. As a demand curve is expected to be non-ascending, a supply curve is expected to be non-descending.

Returning to the price elasticity of electrical energy demand, in a meta-analysis of 15 dynamic pricing pilot programs, Faruqui and Serqici attempt to quantify the price response of residential electric energy consumption [12]. The range of experiments included many variables, including data collection and analysis methods, timing of the dynamic pricing event, and geographic location. As a result of these variables, the authors conclude that the measured elasticities of demand vary widely. In addition to the variables listed, the following factors influence the observed price elasticity of demand: availability of enabling technologies, ownership of central air conditioning, and the pricing rate structure itself. Ultimately, they find "substitution elasticities from the experiments range from 0.07 to 0.40 while the own price elasticities range from −0.02 to −0.10".

In "The Short-Run Effects of Time-Varying Prices in Competitive Electricity Markets," Holland and Mansur analyze the expected impact of real-time pricing on wholesale markets. In their analysis they use the following values for the price elasticity of demand: 0.05, 0.10, 0.20. In providing these values, the authors acknowledge that estimated price elasticities vary greatly, but state "this range is generally viewed as plausible" [13].

This analysis focused on the impact of RTP on the wholesale energy markets, not the households themselves, but these values provide an indication of the expected range of values. However, it is important to note that these values reflect the aggregated price elasticity of electric energy consumption; each individual customer, contributing to the aggregation, should be expected to have a wider range of variation.

Insights on the elasticity of each individual customer are gleaned from "The Long-Run Efficiency of Real Time Electricity Pricing" [14]. In this article, Borenstein develops a detailed model for customer energy demand, including elasticity. The author acknowledges that elasticities depend greatly on technology and are expected to increase in the long term: "the range -.025 to -0.150 illustrates the likely impact of RTP in the short run and under current technologies for demand response…in the longer run, however, real-time demand response will become easier to automate and larger elasticities might be expected, so I include results using -0.3 and -0.5 as well."

Additionally, Borenstein notes that elasticity should be expected to vary over the course of the day. In the model used, the author varies elasticity linearly across demand levels: 50% of the original demand elasticity is used the lowest demand; 192% of the original demand elasticity for the highest demand level. The author states: "these boundaries were chosen so that the demand-weighted average elasticity is equal to the original demand elasticity in order to allow some comparability to the previous simulations." Note, for the largest elasticity included in the model, -0.5, this would result in periods in which a customer's time-varying elasticity is modeled as -0.96.

Ultimately, these articles provide a large range of price elasticities values, with many influencing variables. This provides two conclusions: one, a DDS marketplace must be flexible; two, modeling the behavior of actors within a DDS marketplace has room for tremendous variability.

Finally, a review of dynamic pricing literature reveals the importance of hedging in RTP implementation. Hedging is a financial action designed to minimize the risk of changes in price. In a traditional retail electricity market, customers are (perhaps unknowingly) hedging against future wholesale energy prices. If a rise in fuel prices causes a spike in whole energy prices, retail customers will generally face the same retail rate. When a customer is exposed to real-time prices, they face higher risk.

As discussed by Borenstein in "Time-Varying Retail Electricity Prices: Theory and Practice," a RTP market introduces new risks to both utility retailer and end-use customers [15]. The author goes on to describe methods to provide hedging and risk mitigation. These observation can certainly be applied to the DDS Distribution System Operator (DSO) and customers in the DDS marketplace. In particular, Borenstein proposes:

“[The retailer could] offer to pass through the wholesale spot electricity price and would augment that offering with various price protection programs, such as a BYO baseline [build-your-own hedging strategy]. The retailer would then hedge its wholesale price risk in a way to match the retail price hedging that its customers have chosen to purchase. In essence, the retailer would serve as a broker of risk hedging services.”

Determining the specific hedging strategy for utility retailer is beyond this thesis. However, these comments provide an addition insight into the design of a DDS market. The market must support hedging strategies and risk mitigation by actors within the system. Ideally, the market would be flexible enough to support different strategies in different DDS implementations.

From this section of the literature review, the following conclusions apply to the design of a DDS marketplace.

One, dynamic pricing experiments have demonstrated that customers are willing to modify their energy consumption based on changing prices. However, the degree to which consumption changes – the price elasticity of demand – varies greatly across customers, time, and use cases. To support this observation, a DDS marketplace must provide end-use flexibility, in which a customer is able to modify their real-time consumption, based on their time-varying preferences.

Two, enabling technologies improve the price responsiveness of customers, so the marketplace and supporting infrastructure should allow for automated processes, based on observed and provided customer preferences.

Three, a proposed DDS marketplace must enable risk hedging by actors within the system. Ideally, it is flexible enough to allow actors to determine their optimal hedging strategy.

Finally, it is necessary to acknowledge that the concepts described in this literature review can be addressed under the umbrella concept of *transactive energy*. As defined by the GridWise Architecture Council, a DOE-sponsored organization, transactive energy is “a system of economic and control mechanisms that allows the dynamic balance of supply and demand across the entire electrical infrastructure using value as a key operational parameter” [16]. Further, some principles of transactive energy systems are: highly coordinated self-optimization; maintaining system reliability with enabling integration of DER; non-discriminatory participation; observable and auditable results; scalability and adaptability; and holding actors to an accountable standard of performance. These concepts will influence the development of the DDS market design.

This concludes the literature review. As described, there are externalities associated with DER penetration. These externalities necessitate changes to traditional retail utility business models. Proposed business models and studies on customer response to dynamic pricing, provide observations that are relevant to the development of a DDS market. These observations will be incorporated into the DDS market proposed in this thesis: the Dynamic Tariff Distribution Marketplace (DTDM).

3 Dynamic Tariff Distribution Marketplace

3.1 Overview

The Dynamic Distribution System (DDS) provides a solution to many of the concerns with wide-scale DER deployment, while still providing the many benefits. Additionally, implementation of a DDS can provide new business models and yield new efficiencies in the electric energy sector. As previously described, the DDS requires an internal energy marketplace, which provides discriminated price signals to actors within the system. This Distribution Marketplace can also play a role in managing the externalities associated with DER deployment.

The term Distribution Marketplace will be used to describe: the overall marketplace structure and rules; the communication processes and algorithms implementing this structure; and the actual act of customers and coordinators participating in the market. At its core, a Distribution Marketplace facilitates communication that determines an energy clearing price. This price is then communicated to actors, upon which they base their energy consumption. Following energy consumption, payments are calculated in accordance with marketplace rules. A Distribution Marketplace should, at a minimum, reflect the locational and temporal cost of energy, compensating for physical system energy losses.

The proposed Dynamic Tariff Distribution Marketplace (DTDM) relies on dynamic tariffs to capture externalities, incentivize favorable system behavior, and disincentivize unfavorable system behavior. Imposed tariffs follow a pre-established structure and convention, but update parameters based on real-time or anticipated system considerations. For example, a Prediction Tariff would be structured to incentivize accurate prediction of future loading. The actual incentive impact is driven by a pricing parameter, which is updated dynamically: if accurate prediction is currently valuable, the pricing parameter is high; if accurate prediction is not currently valuable, the pricing parameter is low.

Thus, the cost of energy at any point in the system would reflect three components: the upstream unit cost of energy, energy loss adjustment, and the impact of all intermediary dynamic tariffs.

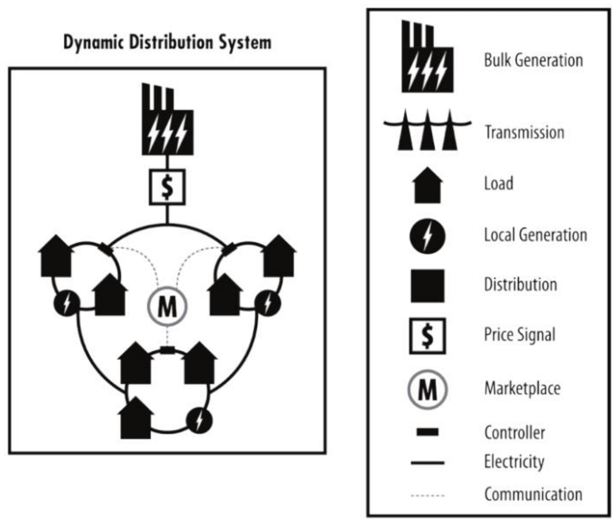


Figure 9: Dynamic Distribution System with Marketplace [1]

3.2 Design Objectives

Development of the proposed Dynamic Tariff Distribution Marketplace (DTDM) was an iterative process. The design objectives included end goals, guiding principles, and market consideration. The DTDM does not attempt to directly solve all stated goals. Instead, the DTDM is a platform upon which further development and innovation can occur.

3.2.1 End Goals

The end goals are based on supporting the implementation of a Dynamic Distribution System (DDS). To this end, the DTDM should support interconnection with a larger energy system, provide intra-system control capabilities, quantify externalities, and support islanded operations.

First, the DTDM should provide the external network with confidence in interfacing with the Dynamic Distribution System. The external network could be the wholesale electric energy market, a traditional electric utility, or a larger DDS in which the DTDM is a component. In this context, confidence extends to both physical power quality and financial forecasting reliability. The DTDM does not purport to manage power quality or stability directly. Instead, it provides minute-scale control and influence on energy balancing. Additionally, it provides the Distribution Systems Operator (DSO) with additional tools to encourage acceptable power quality. To support intermarket financial forecasting reliability, the DTDM must not disrupt the aggregate load forecasting performed by the wholesale energy market. In this context, the DTDM is a success if it provides greater reliability and predictability to financial instruments in the electrical energy sector.

Second, the DTDM should provide the system operator with load control. This enables a new distribution-level capability in distribution operation, specifically when sizing infrastructure for peak loading conditions. Additionally, this supports the first stated goal, as demonstrable load control can enable new relationships with the larger electrical energy network. For example, a successfully deployed DTDM could participate in the wholesale market as a managed load, virtual power plant, or synthetic ramping reserve. These new use cases can support better energy rates for customers within the DTDM.

Third, the DTDM should quantify and mitigate externalities. As previously described, there are numerous externalities, both good and bad, associated with wide-scale DER deployment. A successfully implemented DTDM should capture these externalities and ensure they are borne by the appropriate system actors.

Fourth, the DTDM should support islanded operations. Not all practical DDS networks would be capable of islanded operation. However, a well-designed DTDM would provide flexibility for the systems that support such operation. With a large-scale network of autonomous actors, this involves preventing system power imbalance.

3.2.2 Guiding Principles

With these goals in mind, development of the DTDM was guided by four guiding principles: indirect control, scalability, flexibility, and minimal communication. These principles are certainly not requirements for a Distribution Marketplace. However, each was selected because it either supports real-world implementation or integrates with the larger goals of the Dynamic Distribution System.

The first principle is indirect control. This is an acknowledgement that, in nearly all cases, autonomous actors will be responsible for the end-use consumption of electrical energy. In the context of the DTDM, the indirect control is provided by setting prices, not dictating energy consumption quantities. By necessity, this provides additional risk to the system operator. However, the DTDM rules are designed to enable mitigation of this risk through well-planned implementation and hedging.

In contrast, a proposed Distribution Marketplace that relies on direct control is naturally limited by voluntary participation and available technologies. It is unreasonable to assume customers will provide full control of their energy consumption to any third party. However, a customer may elect to install a Home Energy Manger that will locally control energy consumption based on a price signal. The DTDM proposal conjects that this provides a greater range of potential consumption responses.

The second principle is scalability. A well-designed Distribution Marketplace should not be constrained to networks of a certain size or type. In the context of the DTDM, scalability enables the fundamental rules and configuration to apply from household-scale device management to campus-scale microgrids to neighborhood-scale electrical distribution networks.

The third principle is flexibility. This principle recognizes that a proposed Distribution Marketplace cannot anticipate, and thus solve, all potential problems. The design should allow system participants to innovate solutions to meet anticipated and unanticipated problems. As a result, the DTDM will focus on processes and protocols. Proposed interactions with the system will be illustrative, not prescriptive.

The fourth principle is minimal communication. A Distribution Marketplace, by definition, will include multiple independent actors. The information pass between actors, or to a third party, should be limited to that necessary to operate the market. This derives from related considerations for efficiency, reliability, and security. Efficiency is provided by streamlining communication processes. Reliability is provided by removing the need for an always-on third-party connection for market implementation. Security is provided by minimizing the information any customer communicates about their energy consumption.

3.3 Network Structure

The Dynamic Tariff Distribution Marketplace (DTDM) network consists of nodes, linkages, and tariffs. Nodes are locations are which energy flow can be potentially measured, linkages connect nodes, and tariffs are imposed on linkages.

The DTDM also specifies owners for each node, linkage, and tariff component. An owner can be considered a firm or actor that supplies, consumes, and/or transports energy within the DDS. It is expected that an owner is aware of all components within their control.

The DTDM assumes a radial physical network. As a result, nodes exist in a hierarchy. For a given node, the node higher in the hierarchy is its supernode. For a given node, any nodes lower in the hierarchy are its subnodes. A node may have multiple subnodes, but a node may have only one supernode. This is the single supernode convention.

One node in the DTDM has no supernode; this is the top of the hierarchy. This is the Top-Level Node. Similarly, many nodes will have no subnode. These are Bottom-Level Nodes.

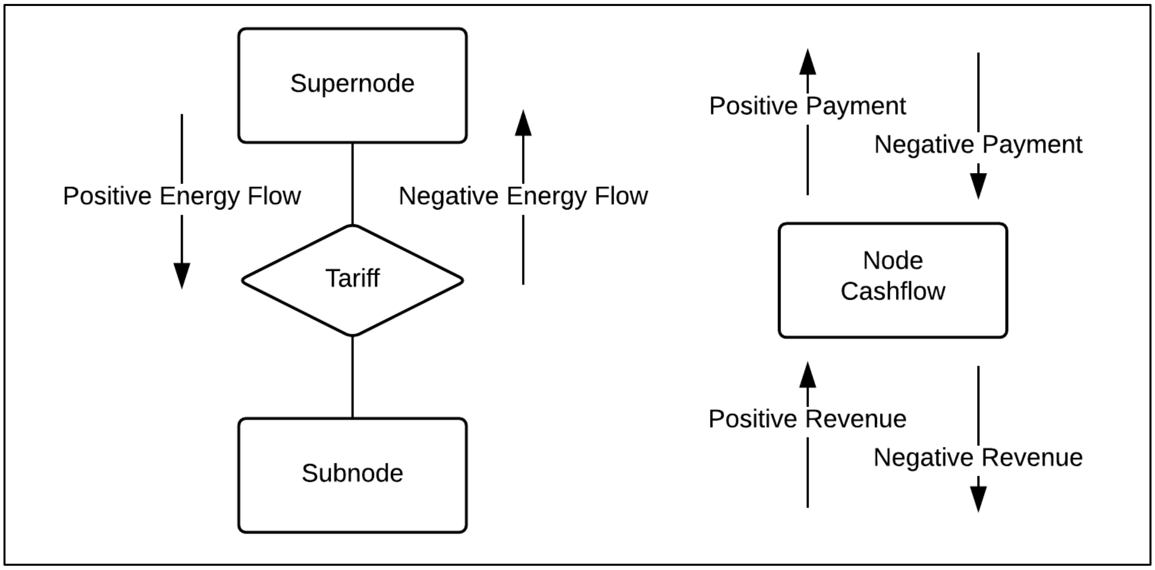


Figure 10: Energy and Cash Flow Conventions

The DTDM uses an energy demand convention. Positive energy flow is always from the supernode to the subnode. Energy flow from a subnode to supernode is also acceptable; this is expressed as a negative value. Cash flow convention moves up the hierarchy. Positive cash flow is a payment from a subnode to its supernode; this is also positive revenue from the supernode's perspective. Positive cash flow also occurs from a node to a tariff; positive revenue for a tariff is cash flow from the subnode.

Nodes are connected by linkages. A given linkage between a supernode and a subnode is owned by the supernode. Consequently, an actor within the DTDM establishes their scope at their most upstream node. Ownership then flows to all downstream linkages and nodes, unless interrupted by a new owner.

A tariff can exist on the linkage between any subnode and its supernode. Multiple tariffs may exist on any linkage.

Like linkages, the supernode owns the tariff. In this context, the owner collects all tariff revenues and is responsible for setting the parameter values. However, the subnode bears the tariff; its owner is responsible for making tariffs payments and adjusting its demand based on the tariff

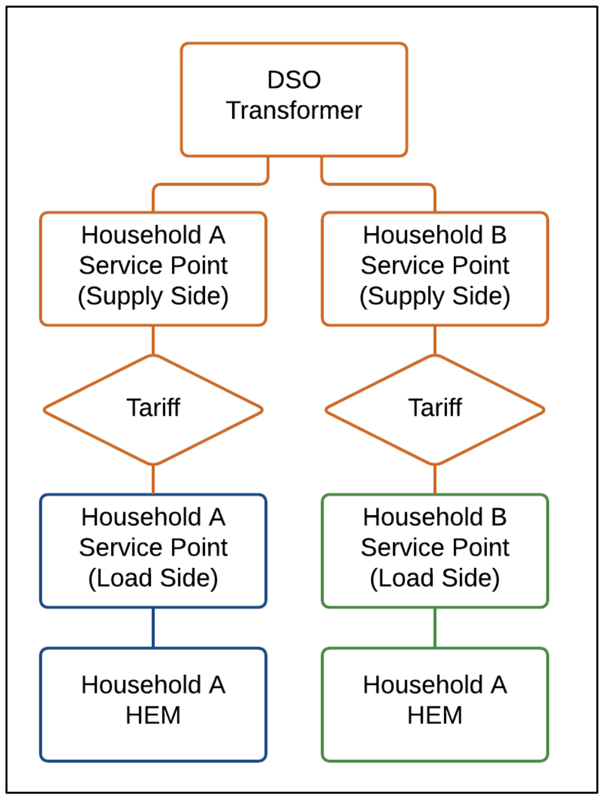


Figure 11: Node, Linkage, and Tariff Ownership Illustration

parameters. Thus, when a subnode submits its demand curve to its supernode, it is expected that the subnode has adjusted its "true" demand curve based on any tariffs.

Tariffs will commonly occur at the interface between two owners. However, a tariff may also occur on a linkage where the subnode and supernode have the same owner. In this case, the same owner sets the tariff parameters and adjusts the subnode demand accordingly. However, there is no cash flow from the owner's perspective, as the payments from the subnode would equal the revenues of the tariff. This would be particularly common for Capacity Tariffs, where the owner does not want to exceed the limitations of a physical component within the network.

3.3.1 Physical System Representation

The first step in configuring the DTDM is to describe the physical system layout. In this context, a node is a physical location on the network at which energy flow can be measured. Physical nodes are connected by physical linkages which impose losses. These losses will be parameterized in each instance with a Linkage Loss Constant.

In a radial Dynamic Distribution System configuration, the top-level node is the point furthest upstream. In the context of the DDS, this is the point at which the Distribution System Operator interfaces with another entity, such as a regional utility firm or the wholesale energy market. This would commonly be a substation. Additionally, this could be the point at which a radial subsection of the electricity system transitions to the larger, networked electricity system. This larger, networked system may itself be a Dynamic Distribution System, with its own rules. In this case, the DTDM is limited to the functional cluster with a radial configuration. The physical system is further defined by proceeding downstream from the top-level node, until reaching a bottom-level node. Every linkage must be the sole path between a node and its supernode. For example, if the top-level node is the high-voltage bus on a substation transformer, its subnode would naturally be the low-voltage bus on the transformer. The linkage between these two nodes consists of the transformer losses. If two feeders leave the transformer, then the low-voltage bus node would have two subnodes. This is illustrated in Figure 13

While a linkage cannot split without first passing through a node, a linkage can represent multiple system components. For example, if a feeder services a single transformer with a single load-side circuit, all those components could be represented with a single linkage. This approach is valid as long as the linkage can still be accurately represented with a Linkage Loss Constant. This evaluation is the responsibility of the supernode owner.

Because each linkage represents the sole path between a node and its supernode, a DTDM network representation applies to only one phase. Practical three-phase distribution systems would be represented by three overlapping DTDM networks. Three-phase sources and loads are thus included on all three networks. These sources and loads should not be considered mutually independent; this is examined in the market design.

The physical system representation continues until reaching a bottom-level node. Bottom-level nodes can represent any size of energy consumption or generation. This can range from an individual device to a neighborhood. This only requirement is that the energy flow at the node can be measured and the demand can be predicted.

Finally, physical distribution systems may be structured as loop configurations but operated with open circuits that provide a practical radial configuration. In such networks, switching between different radial configurations would necessitate adjusting the DTDM network structure.

3.3.2 Non-Physical System Components

Once the physical system is represented in the DTDM network, non-physical system components are added. This includes functional nodes and tariffs.

A functional, or virtual, node exists because it serves a purpose in the DTDM market or aids in communication. For example, the Distribution System Operator may operate the marketplace function at the same physical location as storage dispatch. While this physical bus can be represented as one node, these functions may be separated into two nodes. In this case, the linkage between the two virtual nodes does not contribute to physical energy loss.

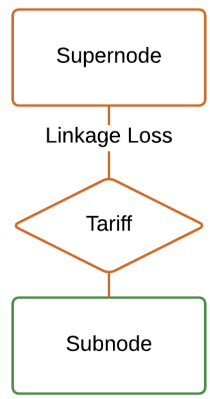


Figure 12: Linkage Loss and Tariff Ownership and Hierarchy

Virtual nodes are also useful for discriminating between ownership. In Figure 14, the service points for two households are illustrated. Although each service point is a single physical location, it is broken into a supply-side and demand-side node. This provides a clear ownership boundary between the supply- and demand-side systems. Notice that both nodes will measure the same energy flow, but this can be interpreted as each owner receiving the energy flow measurements. If the supernode owner does not measure this flow, or is indifferent to the measurement, this virtual node is not necessary. In this case, the service point would be limited to the Load-Side node, owned by the household.

Tariffs are additional non-physical system components added to the network. This is also shown in Figure 14, where tariffs are imposed at the service points, between the supply- and load-side owners. As stated previously, tariffs are owned by the supernode. Additionally, tariffs always occur below linkage losses in the network representation. This is because the subnode owner must bear the tariff, but does not have a direct relationship with the linkage loss.

Tariffs can be established wherever the system operator finds prudent. Typically, tariffs will appear in two locations. One, tariffs will be located at linkages with system-constraining capacity limits. Two, tariffs will be located at the interface between two owners. The location and impact of tariffs will be further examined in detail.

3.4 Marketplace Parameters and Process

With a DTDM network structure defined, the marketplace parameters and processes must be established. Marketplace parameters include the market duration, settlement interval, and tariff structures. The parameters are selected by the Distribution System Operator in advance of operating the DTDM. However, while marketplace parameters can change with each DTDM implementation, but the marketplace process is fixed. The marketplace process is designed to be robust and general, for successful implementation over a range of network structures and marketplace parameters.

3.4.1 Marketplace Parameters

The first marketplace parameter is the *market duration*. This is the interval in which the energy clearing market operates. For example, the NYISO real-time wholesale energy market generally runs every five minutes. There are advantages and disadvantages in setting this duration either short or long. The DTDM relies on an assumption that each customer's demand profile can be accurately described by the average energy flow over the market period. This is not a valid assumption for long market durations, where the "lumpiness" of actual energy demand is hidden. However, the DTDM operation also relies on individual components responding to price signals. If the market duration to too short, components may not have time to respond to price signals; consider an air conditioner expected to cycle from off to on every ten seconds.

The market duration also impact the method by which HEM and smart devices can anticipate the customer's demand preferences. However, this is also resolved by intelligent tariff and settlement design, so it may not be considered in selecting the market duration.

The second marketplace parameter is the *settlement interval*. The DTDM results in locational prices that are valid for the market duration. Settlement is the process of measuring energy flow and assessing payment at the locational price. The DTDM does not require that settlement occur only at the end of the market period. This parameter is the settlement interval, which may be less than the market duration. When assessing energy payments at a set price, this interval has no impact: (1 kWh + 1 kWh + 1 kWh)*($0.10/kWh) = (3 kWh)*($0.10/kWh). The only requirement is that the settlement interval evenly divides the market duration and aligns with the end of each market period. Tariffs and prices span market periods, so this is to prevent a settlement interval from "overlapping" two market periods.

However, this parameter can strongly impact the impact tariff settlements. Consider a tariff that is designed to incentivize non-volatility, within a DTDM with a 5-minute market duration. A 5-minute settlement interval would incentivize maintaining a constant average energy flow between 5-minute settlement intervals. However, a 1-minute settlement interval would incentivize maintaining a constant energy flow between 1-minute settlement intervals. This different would produce different optimal strategies for energy management algorithms.

Marketplace parameters also include *tariff structures*. Tariff structures are be established by the tariff owner and communicated to the tariff subnode owner. The structure indicates how energy is measured, how settlement is determined, and which tariff parameters are dynamic. These definitions must be done in advance to operating the DTDM. There are considerable requirements for tariff structures; this is described further in following sections.

Additional marketplace parameters include the actual timing of the market process, communication data formats, and precision tolerances. In general, these parameters will not be addressed in the general DTDM outline or specific case study simulations. Instead, they would be resolved during a practical implementation of the DTDM.

3.4.2 Marketplace Process

In execution, the DTDM process consists of three sequential steps: the market process, real-time actions, and settlement. These market process operates for each market duration, settlement operates at the settlement interval, and the real-time actions as always on-going. As a result, steps often run concurrently. Each step is outlined generally below. Specifics for each step and sub-step are highlighted in the Implementation section.

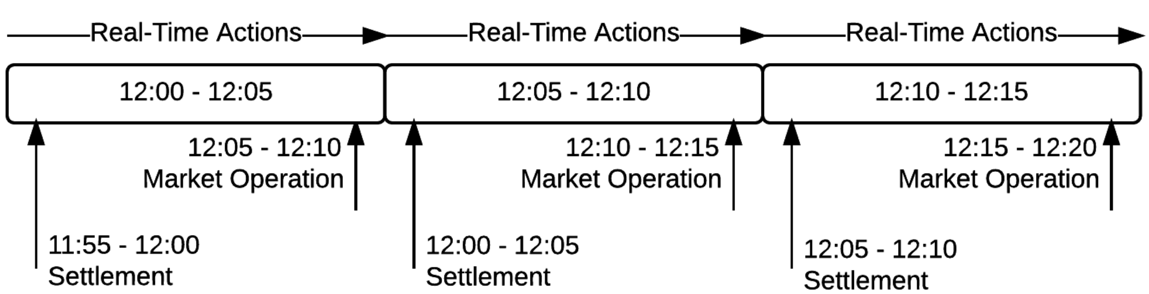


Figure 13: Marketplace Process for 5-Minute Market Duration and Settlement Interval

1. *Market Operation* sets location-specific energy prices that incorporates end-use expressed preferences, current tariff parameters, and matches the supply and demand quantities. Market Operation runs to meet the market duration. In this step, the following actions take place:
    a. Bottom-level nodes establish demand curves based on consumption preferences.
    b. Tariff owners establish and communicate tariff parameters for the market period.
    c. The marketplace node requests demand curves; the request propagates to all nodes.
    d. Nodes consider the impact of tariff parameters and submit demand curves; aggregated demand curves propagate to the marketplace node.
    e. The marketplace node sets a clearing price.
    f. Location-specific contract energy prices propagate to all nodes. This is the Node Marginal Price (NMP)

2. *Real-Time Actions* are the distributed energy consumption responses to location-specific prices, shifting preferences, and applicable tariff parameters. Note, in this context, consumption described both energy utilization (positive energy flow) and generation (negative energy flow). In this step, system response includes the following actions:
    a. Where applicable, tariff parameters updated in accordance with market results.
    b. Bottom-level nodes modify their energy consumption based on shifting real-time preferences, the Node Marginal Price, and tariff parameters. Bottom-level nodes may include dynamic energy-management algorithms that manage consumption during the market period. Alternatively, bottom-level nodes may simply use their NMP to drive a binary consumption decision for the market duration, or even ignore the price. Possible

consumption responses are categorized by shifts in the node's consumption preferences and its ability to dynamically measure and control its load:

i. Expressed Consumption Preference Unchanged / No Dynamic Load Control: Consumption reflects market submission.

ii. Expressed Consumption Preference Unchanged / Dynamic Load Control: Consumption reflects market submission.

iii. Expressed Consumption Preference Changed / No Dynamic Load Control: Consumption reflects updated preference; no adjustment due to price or tariffs.

iv. Expressed Consumption Preference Changed / Dynamic Load Control: Consumption decision balances updated preferences, market price, and tariffs.

c. Dispatch nodes monitor aggregated real-time consumption of subnodes; directly control a dispatchable subnode based on market price and tariffs

3. *Settlement* is the process of allocating payment and revenue for the preceding settlement interval, based measured energy flow, the node marginal price, and imposed tariffs. Settlement does not necessarily include the transfer of funds, which may occur periodically, based on aggregated settlement results. The settlement process includes the following actions:

   a. Total energy transfer for the settlement interval is recorded at each measurement point.

   b. Supernodes collect revenue from subnodes based on the subnode energy flow and NMP.

   c. Tariffs collect revenue from subnodes based on the subnode energy flow, the established tariff structure, and the tariff parameters over the settlement interval.

d. Subnodes make payments to supernodes and tariffs equal to the revenue calculated in 3.b and 3.c above.

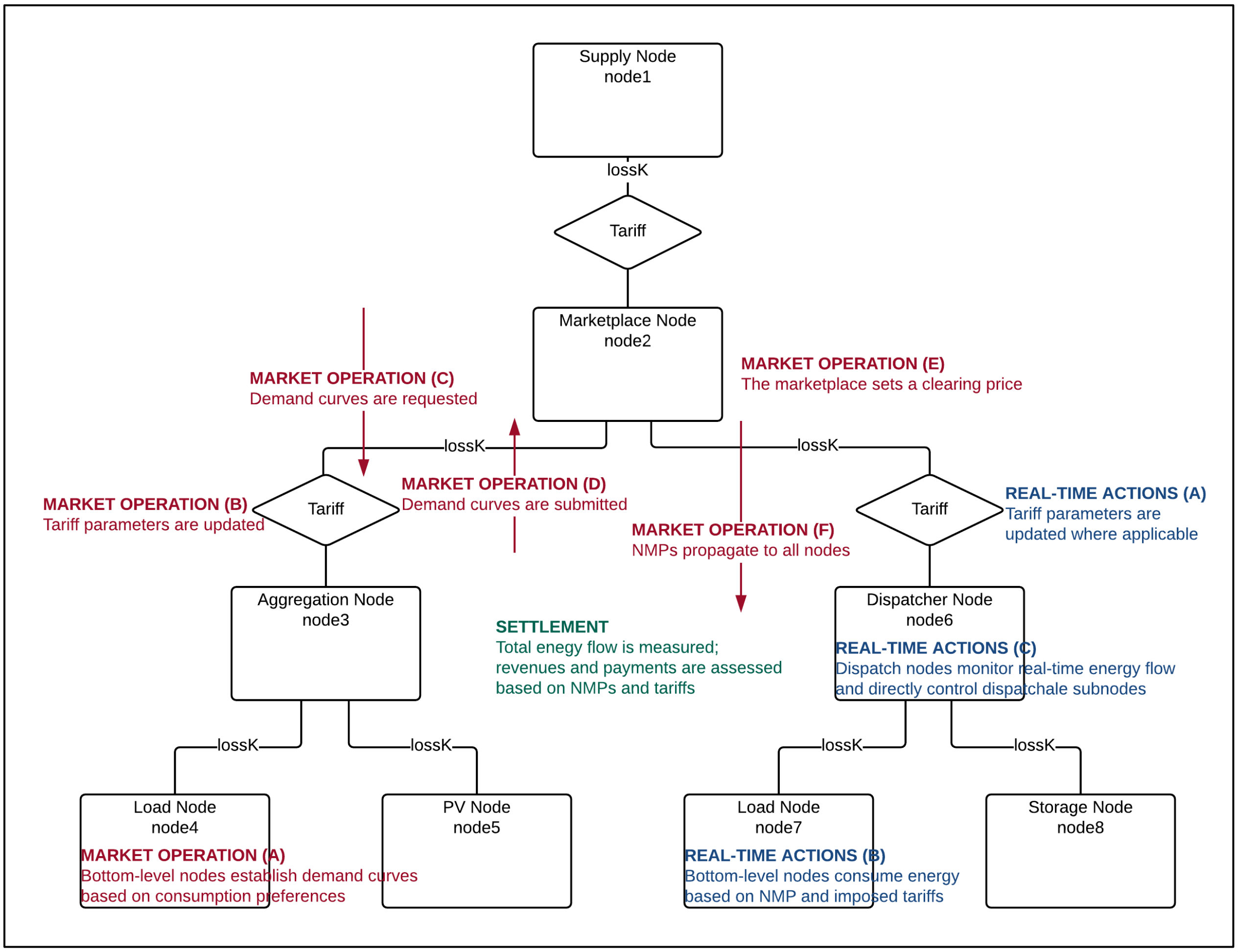


Figure 14: Marketplace Process

In addition to the steps outlined above, a specific DTDM may include a Forward Market, which would run in advance of the standard Market Operation. The Forward Market would balance long-term predictions for energy supply and demand, creating a baseline price for that time period. This process parallels ISO Day-Ahead and Real-Time markets. One benefit to this additional process is to incentivize forecasting and provide price hedging for customers. However, customer preferences in a DDS may be difficult to predict far in advance. Inaccurate prediction of future consumption preferences decrease the value of a Forward Market.

Additionally, as described in Section 2.4.4, forward markets do not impact optimal economic behavior during spot markets. For these two reasons – the difficulty in anticipating future customer-level consumption preferences and the lack of impact on spot market optimization – a Forward Market is not considered in the general description of a DTDM. This does not preclude a specific DTDM implementation for also including a related Forward Market.

3.5 Demand Curves

In the first step of Market Operation, bottom-level nodes establish demand curves based on consumption preferences. The concept is fundamental to the DTDM. A demand curve provides two functions. One, it expresses the node's consumption preferences. Two, it serves as a contractual offer to the supernode.

Serving both functions, a demand curve is a series of points, each representing a quantity (Q) and price (P). Each point is the node making the statement: "If the unit price of energy is P, then I would purchase Q units of energy". When plotted, the quantity is captured on the horizontal axis, while the price is captured on the price axis.

As a contractual offer, a demand curve can present any combination of points, insomuch as they meet the market's communication requirements. However, in microeconomic theory, the demand curve is a specific expression: the marginal utility observed in consuming one incremental unit. For example, consider a household consuming no energy. If the household is not at a comfortable temperature, one unit of energy could be put to use by the air conditioner; there would be high utility for this unit of energy. However, if the household is already at a comfortable temperature, one unit of energy would be put to another, less valuable purpose, such as operating a dehumidifier; there would be a low utility for this unit of energy. By associating price values to utility, a properly calibrated demand curve ensures the customer only purchases energy that is used for purposes of equal or higher value.

As described in Section 2.5, a customer's price elasticity of demand may vary, however a demand curve can always be assumed to be non-ascending.

Following this convention, a rational actor would choose to present their contractual demand curve as an expression of their marginal utility. In the DTDM, it is expected that this rational actor is a Home Energy Manager, translating the customer's consumption preferences. Thus, it can be assumed that demand curve submission reflect marginal utility at that node.

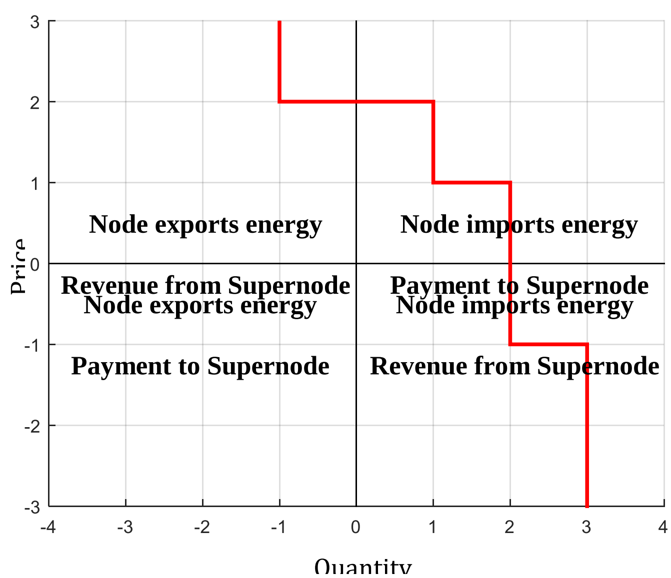


Figure 15: Demand Curve Quadrants

One consequence of the marginal utility convention is that the resulting demand curve is non-ascending. A rational customer will not value an incremental increase of energy more than any previous increment. Of course, a demand curve only represents marginal utility at any point in time; previous and current energy consumption will certainly impact the marginal utility of future consumption.

Similarly, a supply curve represents a node's generation preference and serves as a contractual offer. In this case, each point on the curve represents the marginal cost of producing one additional unit. For example, a diesel generator's supply curve would represent the fuel cost and unit efficiency at every possible output quantity. In contrast to non-ascending demand curves, supply curves is characterized as non-descending.

However, the DTDM uses a demand convention, which describes all load consumption preferences as a demand. To meet this requirement, supply curves are simply flipped along the vertical price-axis. For the diesel generator example, all operating points now reflect a price and negative quantity. Additionally, the flipped supply curve is now non-ascending. All generation in the DTDM will be expressed as a demand curve, using negative quantities.

A tariff imposes costs upon its subnode. These costs can be described by a supply curve, which the subnode uses to adjust their demand curve submission. Similarly, physical system losses necessitate a differing interpretation of a demand curve at each opposing ends of sub- and super-node linkage.

After demand curves are generated, they are communicated throughout the DTDM network and face aggregation, adjustment, and translation. The following section describes each step, while the technical analysis of implementation is included in Section 4.

3.5.1 Node Curve Generation

Theoretically, a bottom-level node can produce a demand curve of any shape, as long as it is non-ascending. In practice, different categories of bottom-level nodes produce demand curves with common characteristics.

A Load Node is a bottom-level node that consists of solely of electrical loads. This may or may not include a HEM that adjusts loads based on price signals. For Load Nodes with a HEM, the demand curve can be expected to have three components: inelastic demand, elastic demand, and maximum demand.

The inelastic demand portion of the curve represents consumption that will not respond to price signals provided by the Node Marginal Price (NMP). This is the left-hand limit of the demand curve. In Figure 16, this quantity is 0.03 kWh. For any NMP, no matter how large, this node will consume at least 0.03 kWh of energy.

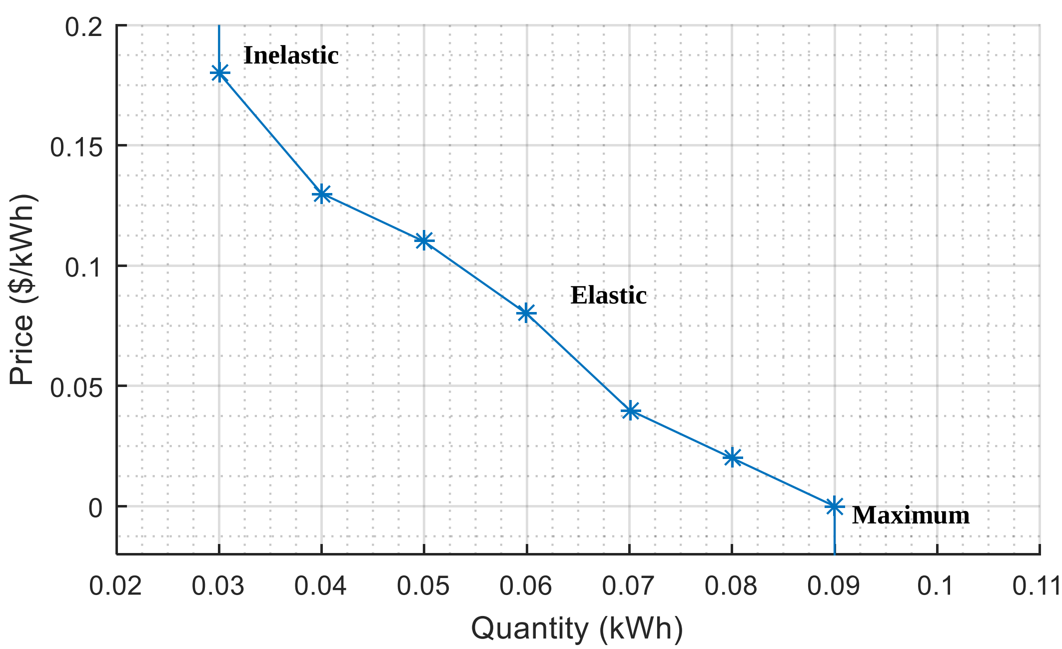


Figure 16: Typical Load Node Demand Curve

The elastic demand portion of the curve represents consumption that will respond to price signals. In Figure 16, this ranges from 0.03 kWh to 0.09 kWh. This is 0.06 kWh of potential load that can respond to price signals. If the NMP is above $0.18/kWh, the load will elect to consume none of the elastic demand. If the price is below $0.00/kWh, the load will elect to consume the entire available elastic demand. The shape of the elastic demand portion of the curve will depend on the HEM's translation of the end-use customer's preferences.

The maximum demand is the highest potential energy consumption. This is the right-hand limit of the demand curve. In Figure 16, this quantity is 0.09 kWh. Notice, for any price, no matter how small, this node will consume at most 0.09 kWh of energy. Note, the price at which the maximum demand occurs need not be $0.00/kWh. In general, this quantity represents every controllable device operating at full capacity. As a result, it can reasonably be expected that negative prices will correspond to the maximum demand; at those points the node is being paid to consume energy.

Note, a Load Node with HEM may also be described within the DTDM as an aggregation node representing the HEM with its subnodes representing the various household loads.

Photovoltaic generation is another common bottom-level node. A PV Node may represent a PV array of any size. The demand curve can be expected to have two elements: maximum generation and curtailment.

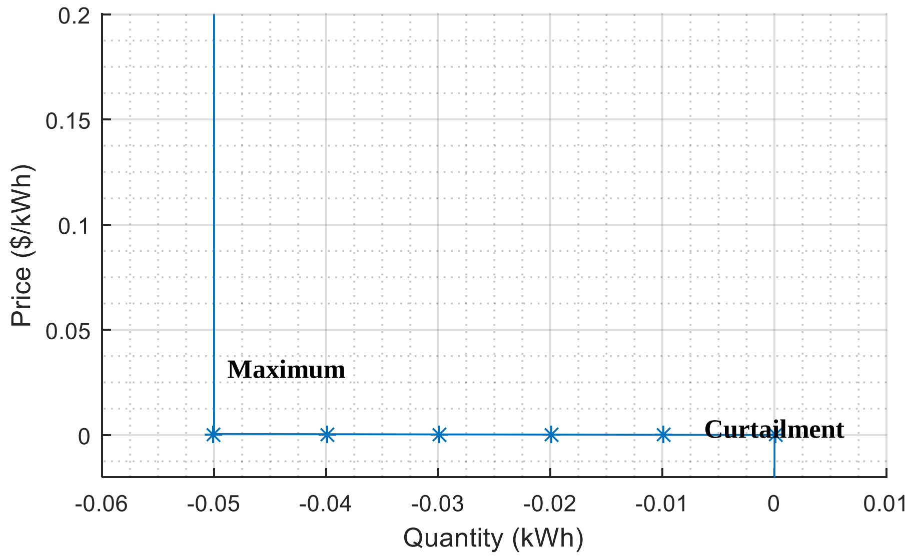


Figure 17: Illustrative PV Node Demand Curve

The maximum generation portion of the curve represents the expected energy generated from the non-curtailed PV array. This is the left-hand limit of the demand curve. In Figure 17, this is 0.05 kWh. Notice, this is a negative quantity, in keeping with the demand convention. For a five-minute market duration, this value corresponds to a constant PV output of 600 W. Note, this quantity value can incorporate predicted fluctuations in the PV output power. For example, the example maximum generation quantity may also represent three minutes at 1 kW and two minutes at 0 kW.

Recall that a supply curve represents the marginal cost of generation. Neglecting any O&M costs, which are not tied to volumetric generation, a PV array does not have any marginal generation costs. Appropriately, the PV Node demand curve provides maximum generation for any NMP above $0/kWh.

However, if the NMP falls below $0/kWh, the PV Node will face negative revenue for generation. Assuming adequate control technology, there is no costs for curtailment, so the PV array will elect to curtail generation. This point occurs at Q = 0, P = $0/kWh. This is the right-hand limit of the demand curve.

Energy storage is another common bottom-level node. A Storage Node may represent any form of dispatchable energy storage, but can commonly be considered stationary batteries. The demand curve can be expected to have four elements: maximum discharge quantity, discharge threshold price, charge threshold price, and maximum charge quantity.

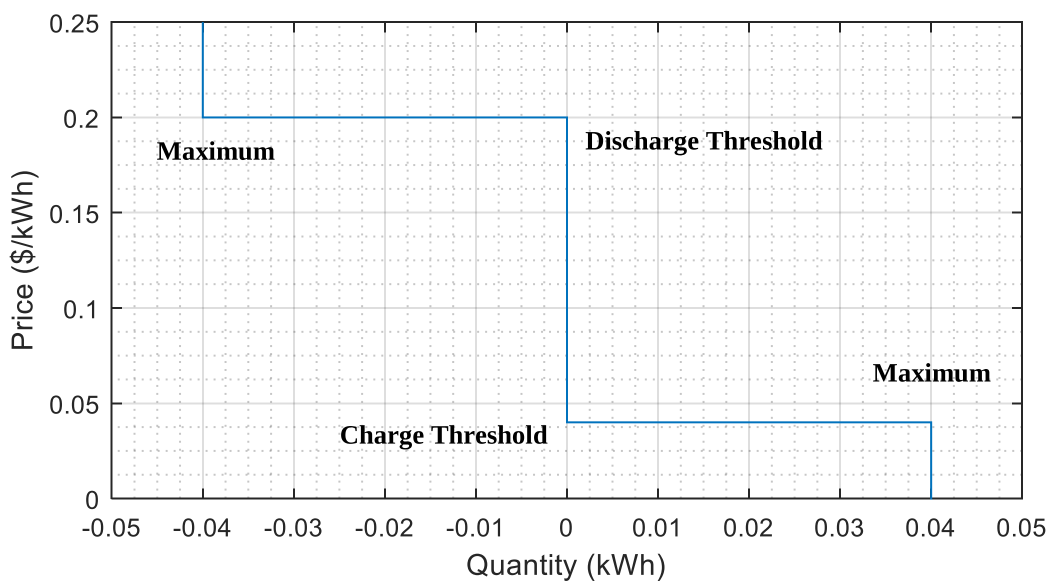


Figure 18: Illustrative Storage Node Demand Curve

The maximum discharge quantity represents the maximum energy the storage device can discharge during the market duration. This represents the left-hand limit of the curve. In Figure 18, this discharge quantity is -0.04 kWh, which is negative per the demand convention. This could represent the 2.4 kW power discharge over a one-minute market duration. Notice, it is not possible to discharge any additional energy, regardless of the NMP.

Similarly, the maximum charge quantity represents the maximum energy the storage device can charge during the market duration. This represents the right-hand limit of the curve. In Figure 18, this charge quantity is 0.04 kWh. In this example, the storage device has the same power rating for charging and discharging.

Note, both the maximum charge and discharge quantities should be limited by the current storage state of charge. For example, if the storage device only contains 0.01 kWh of available energy, the maximum discharge quantity should be limited to -0.01 kWh. The same principle holds if the storage unit is near its capacity limit.

The discharge threshold is the price above which the storage node will discharge. In Figure 18, this is $0.20/kWh. For any NMP above $0.20/kWh, the storage unit will discharge at full capacity. This is the NMP at which price arbitrage makes it “worth it” to sell energy into the DTDM.

Similarly, the charge threshold is the price below which the storage node will charge. In Figure 18, this is $0.04/kWh. For any NMP below $0.04/kWh, the storage until will charge at full capacity. This describes the storage node’s attempt to purchase energy for storage at the lowest possible price.

Any NMP between the two threshold values will result in no energy flow. In Figure 18, this is depicted by a vertical line segment, connecting the two threshold price values.

The threshold values, and the difference between them, would be based on the desired energy price arbitrage and cycling frequency targets. This could be determined in advance of the Market Operation and manually set, or could be determined via a node-level predictive pricing algorithm. In general, a storage unit should be expected to select price thresholds that will maximize the revenue earned through price arbitrage.

Additionally, the Storage Node could elect to "ramp" its curve, instead of including horizontal (or nearly horizontal) sections. This would provide an advantage of more accurately targeting specific energy quantities through the NMP. In particular, this is useful for Storage Nodes used in dispatch. This topic will be further explored in Section 6.

Examination of general bottom-level node demand curves provides some general observations. One, the left-hand limit of a demand curve can be represented by a positive vertical line. Two, the right-hand limit of a demand curve can be represented by a negative vertical line. Three, a demand curve that reflects actual consumption preferences may include both horizontal and vertical segments.

Finally, additional bottom-level demand curves are certainly possible. For example, a diesel generator would have a curve consisting of entirely negative quantities, with prices representing the fuel cost and unit efficiency at every possible output quantity. As another example, a pre-arranged contract for purchasing energy at a set price could be represented as a horizontal line at that price. Any possible demand curve is potentially valid, as long as it is non-ascending.

3.5.2 Curve Aggregation

During Market Operation, each node submits its demand curve to its supernode. When a supernode receives multiple subnode demand curves, it must generate a consolidated curve that represents the contractual consumption preferences of its subnodes. Demand curve *aggregation* is the combination of two or more demand curves. Curve aggregation does not account for the impact of tariffs or physical system losses; those are addressed in curve adjustment and curve translation, respectively.

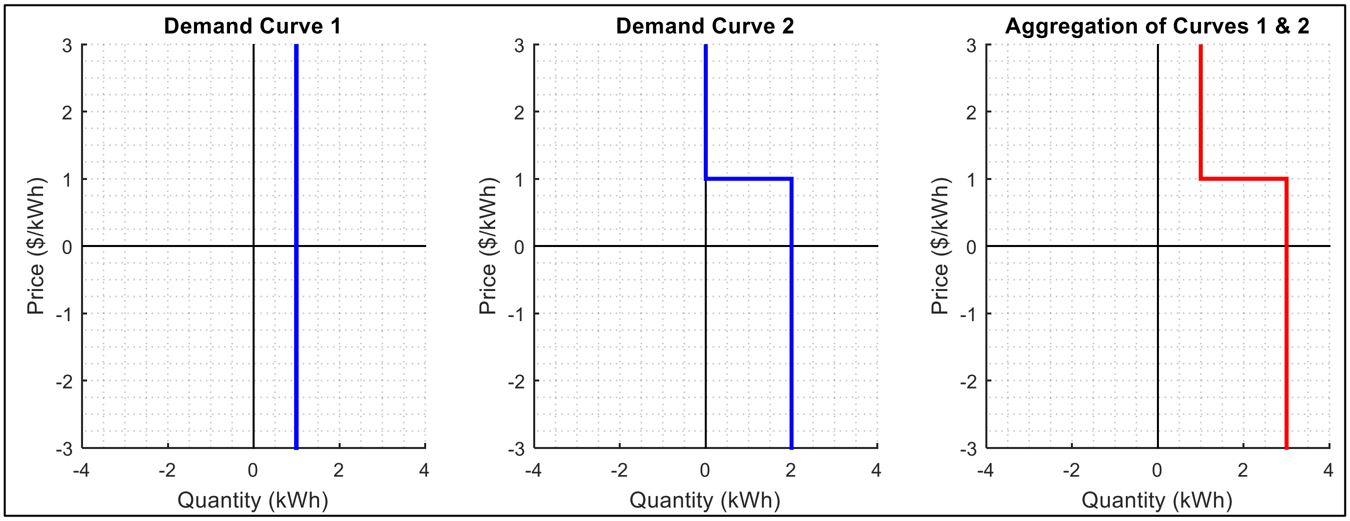


Figure 19: Example Demand Curve Aggregation

Curve aggregation will occur at any node with more than one subnode. If the node performs no functions other than curve aggregation and communication, it is considered an Aggregation Node. Most buses in a distribution system will be represented by an Aggregation Node. Alternatively, a node may perform functions in addition to curve aggregation and communication; examples include Marketplace and Dispatcher Nodes. Any node with multiple subnodes will perform curve aggregation during Market Operation.

Curve aggregation is accomplished by horizontal summation of the component demand curves. This treats price as the independent variable and quantity as the dependent variable. The process is illustrated in Figure 19. In this example, the combined consumption preferences of Demand Curve 1 and 2 are accurately described by the aggregated curve. For example, the aggregated curve indicates demand of 1 kWh for any price above $1/kWh; this corresponds to demand of 1 kWh for

Curve 1 and 0 kWh for Curve 2 at that price. Horizontal summation ensures similar observations can be made at any price point.

Curve aggregation can be accomplished for non-ascending demand curves of any size and shape. Additionally, rather than horizontal summation, demand curve aggregation can be considered the sorting of both demand curve's points into descending order. This observation is illustrative, but provides some practical difficulties, as described in Section 4.

3.5.3 Tariff Curve Generation

The purpose of tariffs are to capture externalities, incentivize favorable system behavior, and disincentivize unfavorable system behavior. A tariff is defined by its *structure*, which is established prior to DTDM operation, and *parameters*, with are set dynamically during Market Operation. A *tariff instance* is a tariff structure with specified parameters. The tariff instance is used to generate a *tariff settlement function* and *tariff curve*.

First, the tariff structure is defined. This describes how energy flow measurements will capture externalities, favorable system behaviors, or unfavorable system behaviors. The structure itself may be dynamic, relying on previous energy flow measurements. For example, a tariff could be designed to incentivize stable, unchanging energy flow. This tariff's structure could compare the measured energy flow in a market period to the measured energy flow of the previous market period. Any change between these two quantities would incur penalty payments.

Second, the tariff parameters are defined. These are set dynamically during Market Operation and serve to adjust incentives based on current locational and temporal considerations. In the example, the tariff parameter would be the magnitude of penalty payment for changing energy flow measurements. This parameter could change throughout the day: when stable energy flow is particularly valuable, this magnitude would be large; when stable energy flow is less valuable, this magnitude would be small.

A tariff instance is a defined tariff structure with specified parameters. A tariff instance is established by setting parameter values at the onset of Market Operation. From the tariff instance, a tariff settlement function is generated. For the example provided, the settlement function is as follows:

$$Settlement = (PenaltyParameter) * |Q_{marketPeriod} - Q_{marketPeriod-1}|$$

At the onset of Market Operation, the settlement function is established by updating the tariff parameters. Next, the tariff subnode must interpret this function and modify their consumption preferences accordingly. This is accomplished with a tariff curve. The tariff curve is the marginal cost imposed upon the subnode by the tariff settlement function. This is simply the derivative of the settlement function. For the example provided, the tariff curve is described by:

$$Price_{penalty} = -PenaltyParameter, \qquad Q_{marketPeriod} < Q_{marketPeriod-1}$$

$$Price_{penalty} = 0, \qquad Q_{marketPeriod} = Q_{marketPeriod-1}$$

$$Price_{penalty} = PenaltyParameter, \qquad Q_{marketPeriod} < Q_{marketPeriod-1}$$

Each point on the tariff curve can be considered a price incentive at that quantity. For a given quantity, if the price is negative, the tariff is incentivizing additional consumption. If the price is positive, the tariff is incentivizing reduced consumption. The magnitude of the price is the magnitude of the incentive. With this convention, the tariff curve can be considered a supply curve: the tariff imposes costs upon consumption. There are two consequences of treating the tariff curve as a supply curve: fixed cost impact and tariff structure restrictions.

The first consequence is the impact on tariff fixed costs. As a supply curve, the tariff curve only represents the marginal costs imposed by the tariff settlement function. This removes the impact of any fixed costs or, alternatively, tariff penalties that are not based on the measured quantity of energy. As an illustration, consider two tariff settlement functions:

$$Payment_1 = (PenaltyParameter) * |Q_{marketPeriod} - Q_{marketPeriod-1}|$$

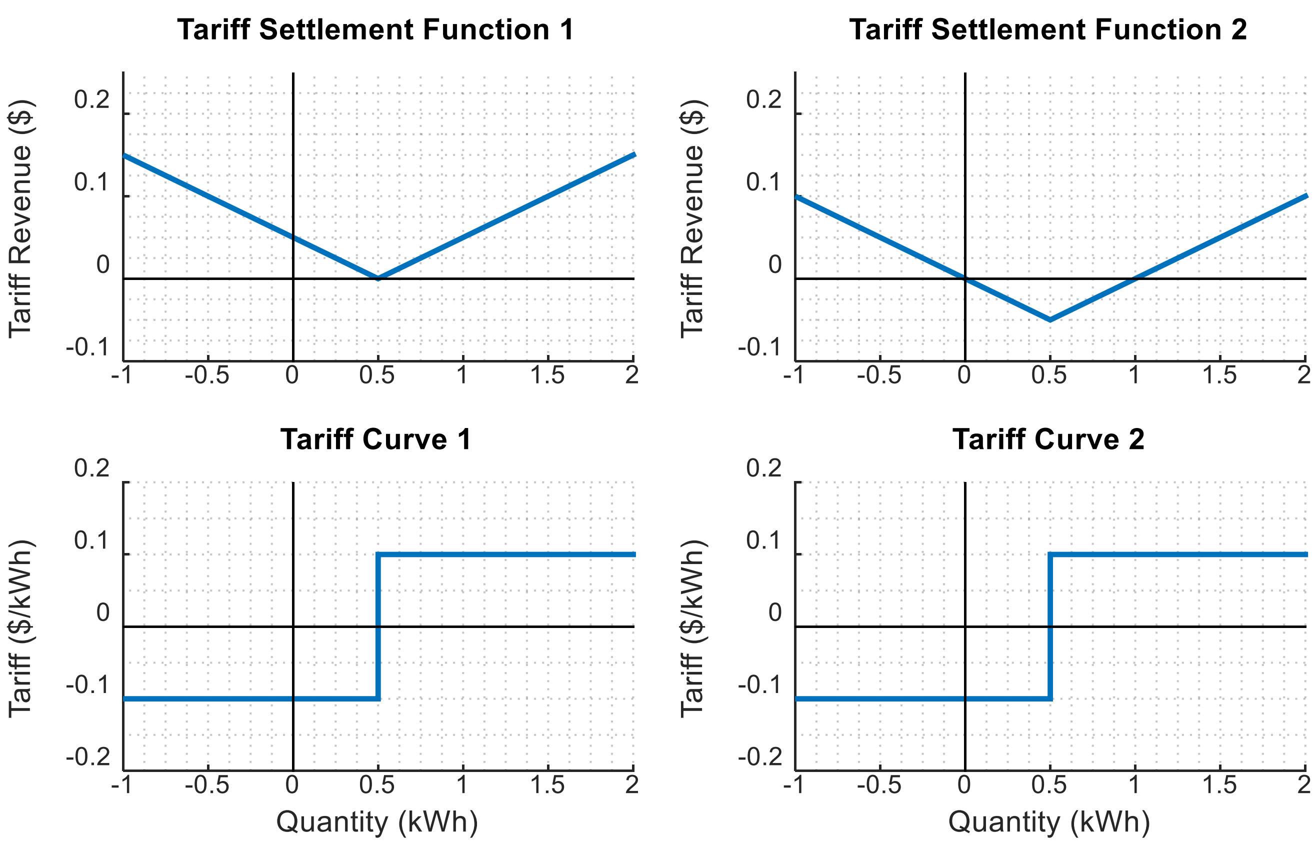


Figure 20: Example Tariff Settlement Functions and Tariff Curves

$$Payment_2 = (PenaltyParameter) * |Q_{marketPeriod} - Q_{marketPeriod-1}| - 0.05$$

The first tariff only penalizes customers for deviating from the previous energy measurement. Alternatively, the second tariff will reward customers (with a negative penalty, i.e. payment to the customer) is they maintaining constant energy consumption. Both tariffs share the same purpose, but the customer perspective of each will be different. However, both tariff settlement functions provide the same tariff curve. This is illustrated in Figure 20, with $PenaltyPayment = \$0.10/kWh$ and $Q_{marketPeriod-1} = 0.5\ kWh$. This is a useful result, as it allows DTDM system operators to design tariff structures that provide the desired system incentive and meet customer cash flow expectations. Additionally, tariff fixed costs can be ignore during simulation and forecasting; the simulation results can then be used to determine tariff fixed costs that meet cash flow requirements.

The second consequence is the implicit restriction of tariff structures. As a supply curve, the tariff curve must be non-descending. In Figure 20, the tariff structures incentivize consumption at 0.5 kWh; this encourages converging on a specified quantity. However, it would not be acceptable to encourage diverging from a specified quantity. For example, a tariff could not reward customers for avoiding zero energy flow with both positive and negative quantities. This would invert the settlement functions in Figure 20 and the resulting tariff curves would be descending, violating supply curve convention.

It is possible to contrive situations in which this limits the tariff owner or DTDM system operator. In these cases, the tariff owner must make a determination of the most valuable incentive and set the tariff parameters accordingly. However, as seen in demand curve adjustment, the benefit of treating tariff curves as supply curves outweighs this potential disadvantage.

From these observations, a tariff owner may elect invert the described process. Instead of beginning with the structure, the tariff owner would consider the desired tariff curve. This would be based on the desired price incentives and anticipated subnode demand curves. From this tariff curve, a settlement function can be established by integration. This would be used to define the tariff structure and parameters. This is a useful approach to tariff development; in fact, it will be the primary method for simulation and case studies.

The example tariffs provided in Figure 20 can be considered a “target quantity” tariffs. This general structure can be useful for many incentives: reducing ramp rate, reducing volatility, encouraging accurate prediction. What differs in these applications is the method in which the target quantity is established. For a ramp rate, the running average of previous energy quantity could be used. For volatility, the previous energy quantity could be used. For prediction, the quantity indicated by the NMP on the demand curve submission could be used; this would serve to encourage the subnode to adhere to their demand curve submission during Real-Time Actions, despite inaccurate prediction or changing consumption preferences.

The “Network Use Charge” rate structure proposed by San Diego Gas and Electric Company, as described in Section 2.3.1, could be implemented as a “target quantity” tariff. In that case, the target quantity is zero, with a tariff curve reflecting their proposed “Network Use Charge” rate.

Additionally, the target quantity settlement function need not be linear. For example, a settlement function with quadratic segments would result in a tariff curve with linear segments. This is useful for incentivizing small deviations more than large deviations. The same concept applies to any tariff type.

Other common tariff types should be expected. A capacity tariff disincentivizes exceeding a physical component's energy limitation. The tariff curve would be characterized by an extremely high price for any energy quantity above this limit. For energy export limits, this would be an extremely high negative price below the energy quantity limit. Additionally, the capacity tariff curve may "ramp" up to these limits, providing more cautious disincentives.

A subsidy tariff would serve to incentivize certain subnode loads. For example, a subsidy tariff could be placed above a PV node in the network, providing the PV owner payments for all PV generation. Such applications are possible where DER is incorporated into a utility's Renewable Portfolio Standard. Similarly, a subsidy tariff could be tied to carbon dioxide emissions from a generation source. This could essentially impose a local carbon tax to disincentivize carbon-intensive grid energy imports or fossil fuel generation within the DDS.

Tariffs could also be designed to encourage power factor correction or phase balancing. In these cases, additional information is incorporated into the tariff structure, but the tariff process follows the same rules.

Finally, in DTDM implementation, a subnode is responsible to interpret a given tariff instance into a tariff curve. The tariff owner establishes the structure and parameters, but the ultimately subnode bears the tariff and must adjust their demand accordingly. This does not impact the fundamental theory of tariff curves; however, it serves to add a degree of freedom for subnode owners. As an autonomous actor, the subnode owner can choose to hedge as they see fit. Their interpretation of the tariff settlement function will incorporate their risk preferences and anticipated prediction accuracy. In general, this concept will not be addressed in the DTDM overview or simulated case studies. Instead, all actors will implement tariff curves as described, without any hedging or further adjustment.

Next, it is necessary to describe how tariff curves impacts demand curves within the DTDM.

3.5.4 Curve Adjustment

A node uses tariff curves to adjust their demand curve submission. A node's curve submission to its supernode is a contractual offer. Thus, a node must incorporate impact of tariffs on the actual energy consumption preferences expressed by its demand curve. Demand curve *adjustment* is this process.

A demand curve represents the marginal utility, or benefit, of energy consumption. The tariff curve represents the marginal cost of energy consumption. The adjusted demand curve should be the net

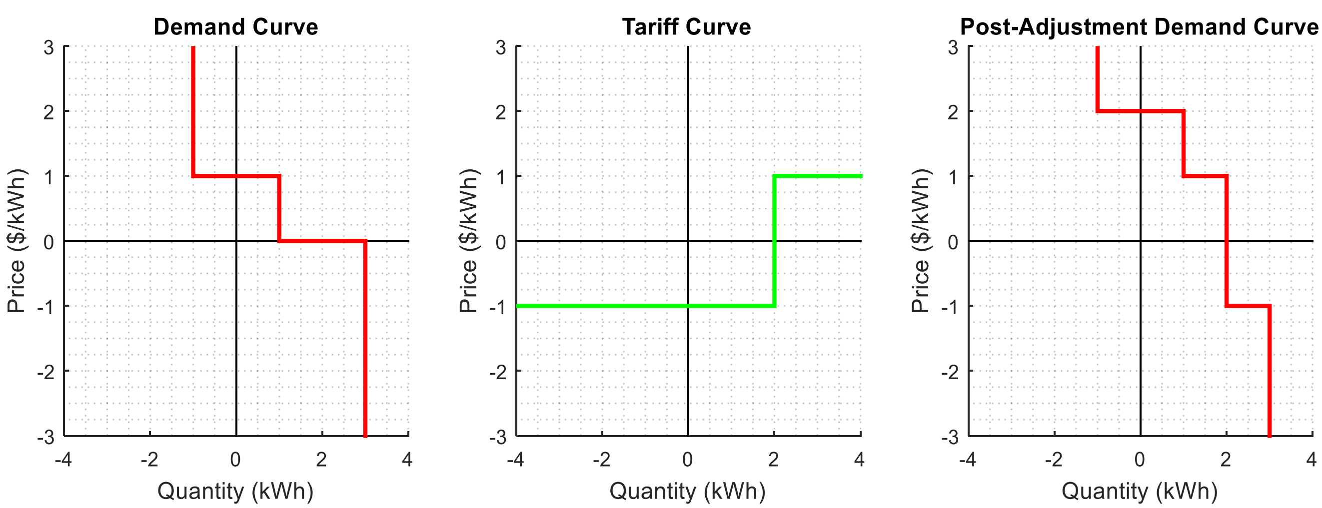


Figure 21: Example Demand Curve Adjustment

utility, when considering the tariff. Thus, the adjusted demand curve is simply the point-wise subtraction of the tariff curve from the demand curve.

Unlike curve aggregation, the adjusted demand curve spans the same domain of quantities as the original demand curve. However, the adjusted demand curve spans a larger range of prices.

Analytically, as a derivative of the settlement function, tariff curve are boundless. They do not have left- and right-hand vertical segments, like those in a demand curve. Thus, a tariff curve adjustment does not modify the quantity domain of the initial demand curve. However, some tariffs, such as capacity tariffs, may impose such strong penalties that the resulting tariff curve functionally has left- and right-hand limits. As a supply curve, the left-hand limit extend to negative infinity and the right-hand limit extends to positive infinity. In these cases, subtracting the tariff curve from the demand curve may provide an adjusted curve with a smaller quantity domain than the initial demand curve. This practical observation is utilized in Section 4 – Model Implementation and Simulation.

3.5.5 Aggregation and Adjustment Illustration

In summary, bottom-level nodes produce demand curves based on their preferences. Typical demand curves represent inelastic loads, elastic loads, PV generation, and energy storage. Demand curves are aggregated through horizontal summation. Tariffs impose costs based on their structure and parameters; this is described by the tariff curve. Demand curves are adjusted to accommodate these costs by point-wise subtraction of the tariff curve.

An illustration of these steps is shown in Figure 22. The following description serves to outline each step.

Node B is an inelastic load with a vertical demand curve at Q = 1. Node B will demand Q = 1 at any price. Node C is an elastic load with acceptable loads ranging from Q = 0 to Q = 2. If the price is more than 1, Node C will demand Q = 0; if the price is less than 1, Node C will demand Q = 2. These two nodes are aggregated at Node BC. Notice the resulting curve now ranges from Q = 1 to Q = 3. For prices more than 1, Node BC will demand Q = 1 (1 for Node B and 0 for Node C). For prices less than 1, Node BC will demand Q = 3 (1 for Node B and 2 for Node C).

Node A is a PV load, which will export energy its full capacity (Q = -2) at any price above 0. If the price is less than 0, the PV load will curtail, to prevent avoid paying for the privilege to export energy. Notice, while Node A is strictly supplying energy, its curve can be easily represented as a demand curve with negative quantities. Node ABC aggregates this curve with the curve observed by Node BC. For prices more than 1, Node ABC will demand Q = -1 (Node A exports 2, Node B imports 1, Node B imports 0). Similar observations can be made for any price point on the ABC demand curve.

Notice, the aggregation of demand curves is a horizontal sum, with the boundaries determined by summing the contributing curve's limits.

Tariff Z is a "target quantity" type tariff. Quantities above Q = 2 suffer a penalty of P = 1. However, quantities below Q = 2 suffer a penalty of P = -1; this is a payment to the subnode. Node ABCZ incorporates Tariff Z through demand curve adjustment. Points above Q = 2 subtract 1 from the

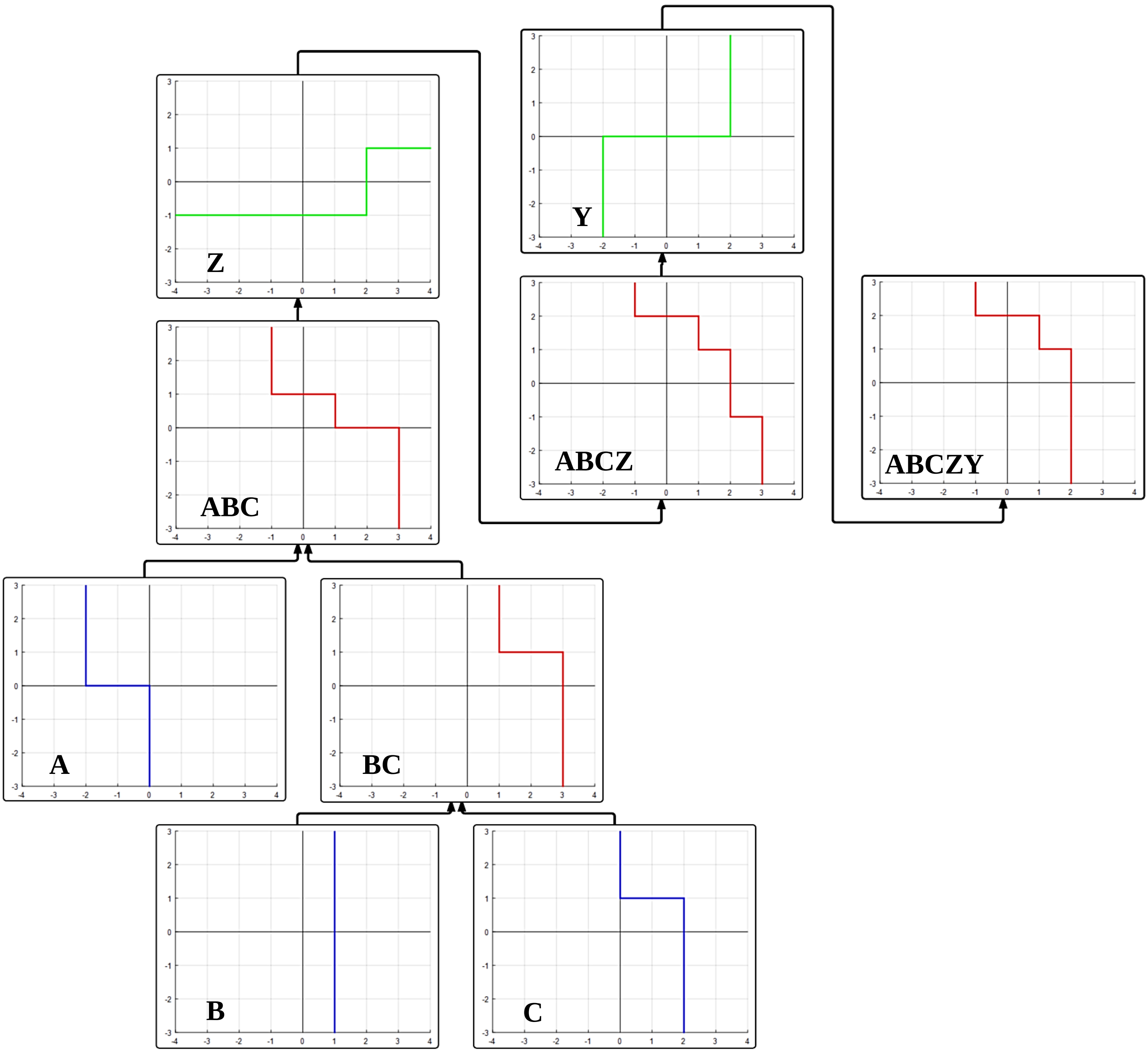


Figure 22: Combination of Demand Curves (Blue Demand, Green Tariff, Red Combination)

ABC demand curve. Points below Q = 2 subtract -1 from the ABC demand curve. This creates a price jump of 2 at the target quantity Q = 2. This reflects the incentive a "target quantity" tariff induces on demand.

Tariff Y is a "capacity" type tariff. This is interpreted as a system component unable to support energy flow above 2, in either direction. Quantities above Q = 2 face an extremely large penalty, represented by a rising vertical line. Quantities below Q = -2 face an extremely large negative penalty. Based on the demand convention, this quadrant specifies payments to the supernode; this is the same disincentive faced by quantities above Q = 2. Node ABCZY adjusts the curve of ABCZ based on Tariff Y. Notice the curve is not impacted between Q = -2 and Q = 2. However, adjustment for Tariff Y "trims" the curve for quantities above Q = 2. Those demand curve points are no longer favorable, based on the consequence of the strong capacity curve.

3.5.6 Translation

In the example above, no consideration was shown for the physical energy losses imposed by node linkages. Another process must be established to incorporate the impact of these losses. Demand curve *translation* is the consideration made for physical system losses.

A node expresses their demand curve in terms of local energy consumption. Due to physical system losses, this quantity is always less than the energy delivered from the perspective of the "upstream" supernode. A demand curve translation must address this difference in quantities. Additionally, revenue is not subject to physical losses and a node's demand curve correlates prices to the local measurement of energy. As a result, demand curve translation must address this difference in prices.

For example, consider the following. A household is willing to purchase 10 kWh at a rate of \$.10/kWh. This is a specific point on their demand curve. The household is the sole customer on a long rural feeder; its supernode is a distant distribution transformer. From the supernode's perspective, system losses require transmitting 11 kWh of energy to provide 10 kWh of energy at the household. From their demand curve, the household is offers to spend \$1 on this quantity of energy (10 kWh x \$.10/kWh). From the supernode's perspective, this is equivalent to P = \$0.909/kWh for Q = 11 kWh. This is the translation of a specific point on the demand curve: 10 kWh at \$0.10/kWh becomes 11 kWh at \$0.909/kWh.

The key to curve translation is an understanding of the impact of physical losses on energy transportation. This process also illustrates the need to represent every physical network junction with a DTDM node; curve translation requires describing the energy flow on each linkage as the consumption at the subnode.

3.6 Market Operation

Market Operation is initiated by a Marketplace Node. First, demand curves are requested. This triggers processes for demand and tariff curve generation, aggregation, adjustment, and translation. The Marketplace Node receives the consolidated demand curve and uses it to set a local clearing price. This is then propagated throughout the DTDM as Node Marginal Prices (NMPs).

3.6.1 Marketplace and Supply Nodes

Commonly, the DTDM is connected to a larger energy network, such as a larger electrical distribution network or the transmission system. This connection is represented by a Supply Node, which is the DTDM top-level node. The Supply Node's only subnode is the Marketplace Node, from which the rest of the DTDM network extends.

The Supply Node is used to represent any energy purchase agreements between the DTDM and the larger energy network. The agreement is translated into a supply curve. For example, if connected to the wholesale energy market with a flat-rate energy agreement, the supply curve is simply an infinitely-long horizontal line at the flat-rate price. More complex financial agreements can be described in this way, to include net metering agreements. Note, tariffs may be imposed on the linkage between the Supply and Marketplace Nodes, capturing physical capacity limits and describing ancillary service agreements.

Alternatively, if a DTDM is islanded and does not connect to a larger energy network, then the Marketplace Node is the DTDM top-level node. The Marketplace Node serves to balance energy within the DTDM network, so generation and demand are equal at all points in time and the market is cleared for each market cycle.

A DTDM may also exist within a larger DTDM network. For example, a campus may elect to run a DTDM with a ten-minute market duration; this collects demand curves and sets prices for all facilities on the campus. In addition, a facility on the campus my elect to run its own DTDM with a shorter market duration. This will balance energy consumption within the prices set by the larger campus DTDM. The campus DTDM will always see the facility as a bottom-level node. However, the facility DTDM will observe the campus DTDM with two perspectives. During the campus DTDM Market Operation, the facility will observe the campus DTDM as a connected supernode. However, during Real-Time Actions, between discrete Market Operations, the facility DTDM will observe the campus DTDM as a Supply Node. This Supply Node will reflect the campus DTDM NMP, with any real-time tariffs imposed upon the linkage. This allows the facility to operate an intra-market DTDM. The same concept can extend to any devices that control consumption more quickly than the DTDM market period, such as Home Energy Managers and storage dispatch devices.

3.6.2 Market Auction

Whether connected to a Supply Node or operating as an islanded DTDM, the Marketplace Node seeks to clear the market. This is done with a double auction market. The market is said to "clear" at the price for which quantity demanded equals quantity supplied. Setting the market price at this

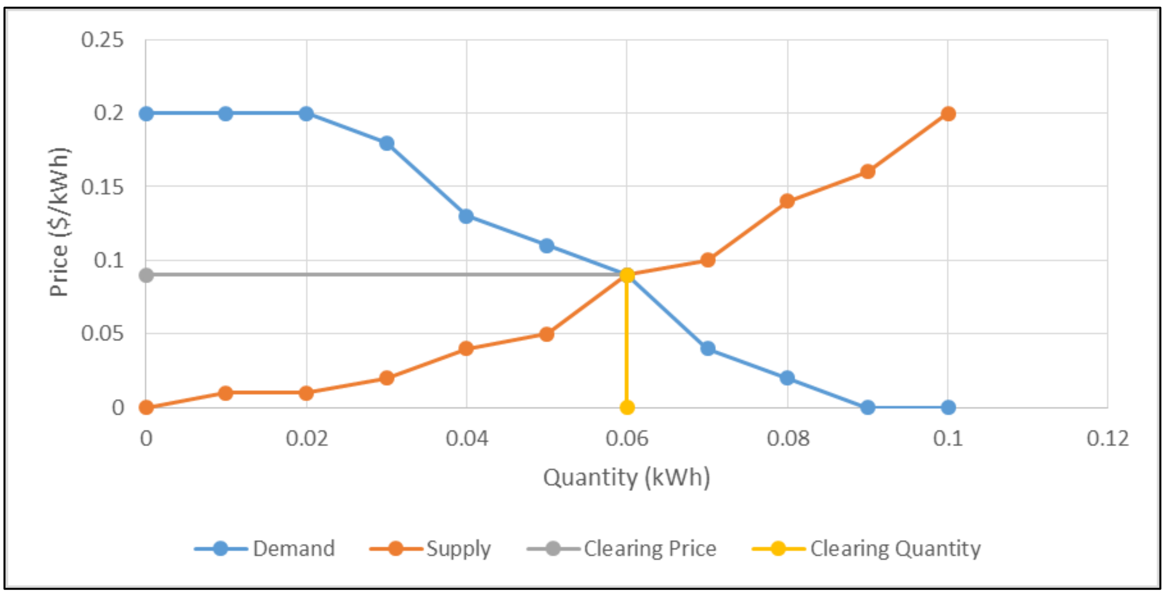


Figure 23: Example Clearing Price and Quantity

value elicits a response from both energy suppliers and energy buyers. Suppliers with offers below this price are willing to sell energy at the clearing price. Buyers with offers above this price are willing purchase energy at the clearing price.

In a single-price double auction, the price at which the market clears is the price for all buyer and all sellers in the market. There is no consideration made for the actual marginal costs and benefits described by the supply and demand curves: PV arrays that are willing to sell energy for $0.001/kWh will be paid at the same price as storage units only willing to sell energy for $0.09/kWh. Note, the clearing price does not establish a price for the entire DTDM, just at the Marketplace Node.

Note, as stated in Section 2.6.2, this is not the only possible configuration of a market auction. However, it's value is in its simplicity and the parallels it enables to existing wholesale energy markets.

An illustration of the single-price double auction is shown in Figure 23. In this case, a clearing price of $0.09/kWh elicits suppliers to generate 0.06 kWh of energy and elicits buyers to consume 0.06 kWh of energy. Because the supply equals demand, the market is said to clear. Notice the impact if the price was lower or higher. At $0.05/kWh, only 0.05 kWh is generated while approximately 0.07 kWh is desired; this imbalance is unacceptable. Similarly, at $0.15/kWh, approximately 0.085 kWh is generated while only 0.035 kWh is desired; this imbalance is also unacceptable.

Generally, when the Marketplace Node is connected to a Supply Node, clearing the market follows this process. In this case, the supply curve derives from the Supply Node financial agreement, adjusted for any linkage tariffs. The demand curve is the aggregated, adjusted, and translated curve received by the Marketplace Node; this represents the consolidated demand of the DTDM.

When in islanding mode, there is no Supply Node to provide a supply curve. In this case, there are two options. One, if the DTDM system operator has on-site generation, they may elect to describe the capabilities of this generation as a supply curve. Two, the Marketplace Node can simply set the clearing price where the consolidated DTDM demand curve crosses the vertical axis.

If the consolidated demand curve does not cross the vertical axis then the market cannot clear. Similarly, in Supply Node operation, if the supply and demand curves do not intersect, then the market cannot clear. This is market failure and must be avoided. In particular, this is avoided by ensuring adequate generation (or supply is available) and that an appropriate portion of loads are flexible. “Adequate” and “appropriate” quantities will depend on the DDS itself. One purpose of simulating the DTDM is to determine these quantities. Additionally, a DTDM system operator can prevent market failure by having on-demand load-shed or generation resources. This could be provided by a separate, local ancillary-services-type market. Such a market could also be used for power quality concerns that occur too quickly to be corrected by the DTDM process.

The double auction market provides an additional restriction placed upon demand curves within the DTDM. Previously, demand curves were only specified to be non-ascending. However, one goal of the DTDM is to provide transparency and deterministic behavior. Yet non-ascending demand curves may include both horizontal and vertical segments. When determining the clearing price and quantity, these segments can result in a range of acceptable prices or quantities. Consider a supply and demand curve with overlapping horizontal segments. In this case, the clearing price is clear, but the resulting quantity is unknown. This could lead to system imbalance. Alternatively, consider an islanded DTDM with a vertical segment on the demand curve at Q = 0. In this case, the clearing quantity is known, but there are a range of prices that elicit this response. Different actors within the system may have a different interpretation of the “correct” clearing price.

To compensate for these concerns, the DTDM requires all demand curves to be monotonic and strictly descending. Previously, the left- and right-hand limits of a demand curve were interpreted as vertical segments. These are now considered ever-so-slightly sloped. However, as previously described, there are occasions in which a "true" demand curve is not strictly decreasing, such as PV and Storage Nodes. The impact on these nodes' curve generation will be addressed in Section 6.

3.6.3 Node Marginal Pricing

Once the Marketplace Node determines a clearing price, it must be communicated to all bottom-level nodes within the DTDM network. Recall, demand curves undergo aggregation, adjustment, and translation before arriving at the Marketplace Node. Similarly, the clearing price undergoes a similar series of steps before arriving at each node. As a result, each node receives a price specific to its location within the DTDM network; this is the Node Marginal Price (NMP).

To propagate NMPs throughout the network, each node compare its NMP to its tariffs and the demand curves received from each subnode. In a process similar to, but simple than, that at the Marketplace Node, each node determine a clearing price and quantity. This provides the NMP it communicates to its subnodes. It is important to note: in this process, this only new communication between nodes is the subnode NMP. All information used to determine this price was provided earlier in the Market Operation process.

3.7 Real-Time Actions and Settlement

A Node Marginal Price (NMP), when received by a node, take effect for the duration of the next market period. During this market period, Real-Time Action processes occur. This begins with updating tariffs as appropriate, adjusting actual consumption, and coordinating dispatchable loads. After the market period end, the Settlement process takes place. During Settlement, internode payments are determined using measured energy flow, NMPs, and tariff instances.

The first step in Real-Time Actions is to update tariff parameters as necessary. In particular, tariffs designed to encourage accurate load predictions seek to incentivize consuming the quantity indicated by the NMP on a node's demand curve. This is a target quantity tariff. During Market Operation, a prediction had not been made (i.e. there had yet to be a demand curve submission). Thus the target quantity was unknown and, during Market Operation, the tariff parameters would indicate a null tariff instance. However, after Market Operation, there is now a target quantity: the quantity corresponding to the NMP on the demand curve submission. This is used to update the tariff instance for use during consumption and settlement. Similar tariff updates could be implemented between Market Operation and Real-Time Actions; the DTDM provides flexibility for solutions agreed upon by actors within the network.

The NMP and active tariff instances are designed to encourage efficient end-use energy consumption (and generation). However, for any length market duration, it can be expected that customer's preferences will shift in real-time. For example, human-triggered loads, such as a microwave, may have been unanticipated by the Home Energy Manager. Additionally, a bottom-level node may or may not have dynamic load control, to adjust for these changes. In general, a bottom-level node's Real-Time consumption can anticipated by these two component, providing four possibilities.

One, a bottom-level node may have unchanged consumption preferences and no dynamic load control. In this case, the demand curve submitted during Market Operation still accurately reflects the node's desired consumption. Thus, the NMP is expected to induce the quantity specified on the node's demand curve, even without dynamic load control.

Two, a bottom-level node may have unchanged consumption preferences and dynamic load control. Like the previous possibility, the demand curve submitted during Market Operation still accurately reflects the node's desired consumption. Thus, the NMP is expected to induce the quantity specified on the node's demand curve, and dynamic load control is unused.

Three, a bottom-level node may have changed consumption preferences and no dynamic load control. In this case, the demand curve submitted during Market Operation no longer reflects the node's desired consumption. For example, the bottom-level node may represent a household branch circuit. In this case, the customer's connected loads may not have been accurately predicted by the HEM. However, the HEM has no ability to control loads on this branch circuit; it cannot turn off the television. In this case, the node's actual consumption will reflect the updated preference, with no consideration made for the NMP or active tariff instances. In this case, a prediction-type tariff would provide revenue to the tariff owner, but would not have provided any impact on the actual energy consumption.

Four, a bottom-level node may have changed consumption preferences and dynamic load control. In this case, the demand curve submitted during Market Operation no longer reflects the node's desired consumption, but the node has some ability to shift consumption in real-time. For example, the customer may have modified the temperature setting on the air conditioner thermostat. With dynamic load control, this updated preference would be compared to the NMP and active tariff instances. The DTDM does not specify how this must be accomplished. However, existing DTDM protocols could be used: the node may update its demand curve based on the new customer preferences. By "replacing" its supernode with a Supply Node representing the NMP, the node can incorporate the NMP and active tariff instances in determining a new, local-only NMP-equivalent.

This NMP-equivalent would only serve to control energy consumption during the market period; during settlement the node would participate in the DTDM using its supernode-provided NMP.

This final case provides a method of bottom-level nodes to manage their actual energy consumption when preferences change during the market period. However, actors within the DTDM network may desire another level of control. Specifically, Dispatcher Nodes would measure subnode energy consumption in real-time and control the demand of specified Dispatchable Nodes based on these measurements, the NMP, and active tariff instances.

For example, the DTDM system operator may face strong tariff penalties for importing energy at a steep ramp rate. After clearing the market and propagating NMPs throughout the system, the DTDM system operator has no control over the actions of the bottom-level nodes. To hedge against the possibility of wildly-shifting preferences, the DTDM system operator may elect to have energy storage units connected at the DTDM network's connection to the larger grid. In real-time, the DTDM system operator measures the actual consumption of the DTDM network. These measurements are used to determine if, and when, to dispatch energy from the storage unit.

A Dispatcher Node may control generation, storage, or a load center. The node being controlled is considered a Dispatchable Node. This Dispatchable Node need to be a subnode of the Dispatcher; the only requirement is the Dispatcher Node can directly control the node's energy consumption with a known impact the total system energy consumption. With this approach, existing Smart Grid processes can be classified into this construct. For example, some utilities manage peak loads with external control over customer's air conditioners. However, one goal of the DTDM is to provide an alternative to such programs. Thus, the primary example of Dispatcher/Dispatchable Nodes will be energy storage located at a service entrance or change of ownership within the DTDM.

Upon the completion of a settlement interval, the DTDM begins Settlement. Settlement is the process of allocating payment and revenue for the preceding settlement interval, based measured energy flow, the node marginal price, and imposed tariffs. Recall, the settlement interval may be shorter than the market period. However, the end of a market period will always coincide with the end of a settlement interval. Settlement does not necessarily include the transfer of funds, which may occur periodically, based on aggregated settlement results. There are three steps to settlement "bookkeeping".

First, total energy transfer for the settlement interval is recorded at each measurement point. This information is passed to both the node and it supernode.

Second, the node must make a payment to its supernode based on this quantity at the NMP. This is revenue received by the supernode. Recall, as both energy quantities and NMPs may be negative, the payment by the node may also be negative. This is cash flow from the supernode to the node.

Third, the node must may a payment to each tariff imposed on the linkage between the node and its supernode. This payment is based on the measured energy flow, the established tariff structure, and the tariff parameters over the settlement interval. From the tariff perspective, this is revenue.

Commonly, a node will have the same owner as its supernode. Less commonly, but still possible, a node will have the same owner as the tariff imposed on the linkage between the node and its supernode. In these cases, the payment from the node will equal the revenue received by the supernode (or tariff). Thus, it is not required for an owner within the DTDM to calculate cash flow for every node within the network. In fact, settlement need only be accomplished for Edge Nodes, which are defined as nodes whose ownership differs from that of their supernode, any subnode, or borne tariff.

3.8 Implementation

The previous three subsections described the underlying principles of the DTDM. There are practical considerations in implementing such a system. Additionally, this marketplace makes some assumptions about a network upon which it would be imposed. Some of this considerations and assumptions follow.

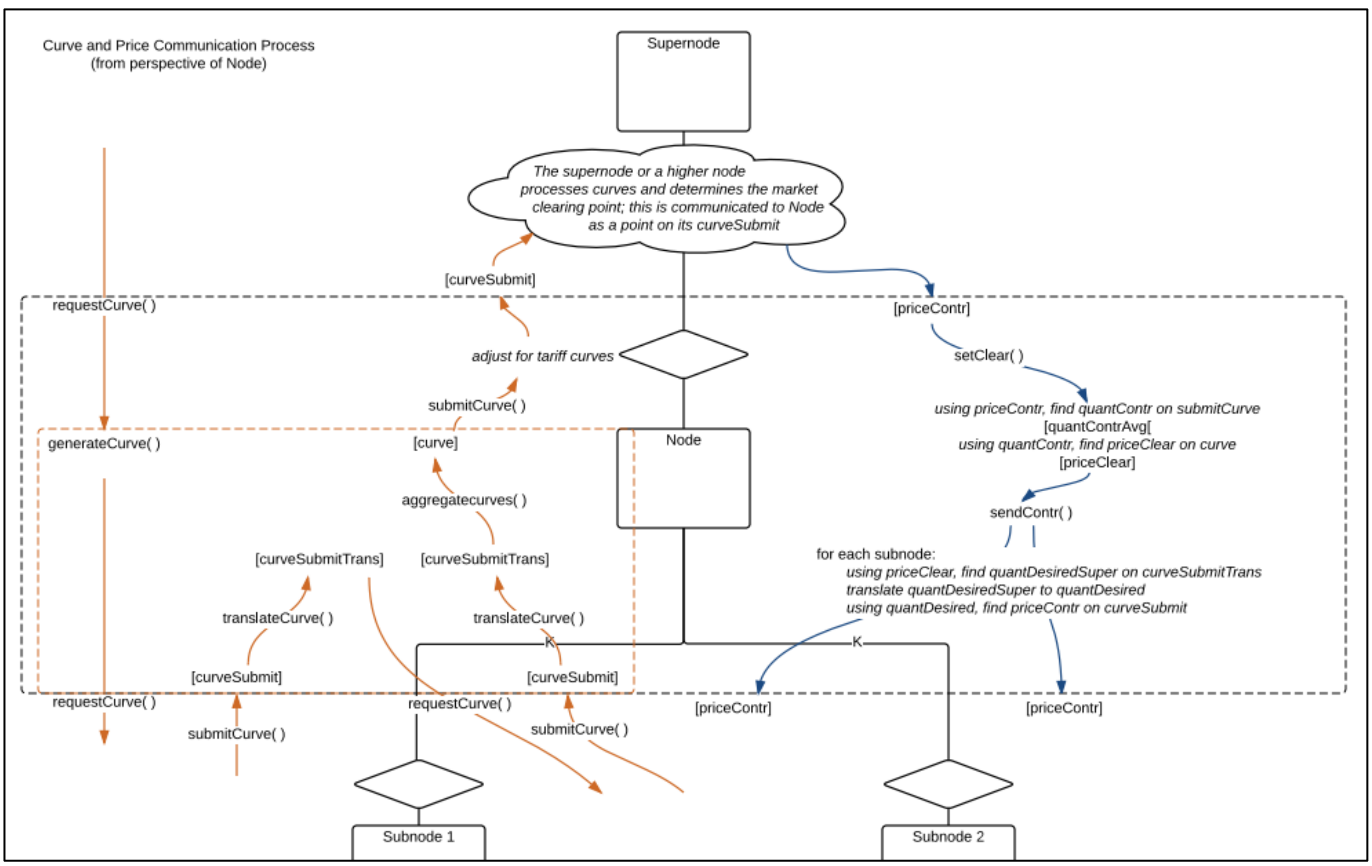


Figure 24: Example Communication Process

3.8.1 Practical Considerations

Implementation of this marketplace, within a DDS, would require deployment of resources to the distribution network. This may include home energy managers (HEM), to translate customer's consumption preferences and control loads, and advanced metering infrastructure (AMI) to measure and record power flow.

This market only applies to a single phase of electric distribution. In systems with all single-phase connections to the network, such as residential communities, this is not a problem: the DTDM system operator will simply run three parallel markets. However, in a network with connected three-phase loads, these markets will not be mutually exclusive. The topic of marketplace protocols and unit commitment is a topic for further study. As a starting point, it is conjectured that three parallel markets would operate, but a common clearing price would be set for all markets. This would necessitate a value judgement by the Marketplace Node, as it cannot be expected that the three consolidated demand curves match.

Communication between nodes would need to be established. This could be via radio frequencies, dedicated communication lines, over the internet, or in many other ways. Summarizing the process above, once the network configuration is established, there is limited communication required between nodes. This includes: tariff parameters, demand curves, node marginal prices, settlement revenues and payments. Of course, additional communication may be required for practical implementation.

Within a subsection of the network with a common owner, communication need not be between different physical devices. Instead an entire portion of the system could be represented virtually.

However, at the boundary of the owner's subsection, this representation ends. An individual owner does not necessarily have any understanding of the network beyond its connected supernode and subnodes. In fact, the market is designed to function optimally without requiring this level of understanding.

3.8.2 Limitations and Assumptions

The DTDM, as outlined, attempts to provide additional benefits and new opportunities to the electric energy sector. However, this proposal includes some assumptions and is not without limitations. Some of these assumptions and limitations are listed. This is not an exhaustive list; it is expected further refinement of the DTDM would occur before deployment.

One, it is assumed that enabling technologies, such as AMI and smart devices, are installed. This capital investment is a sunk cost and is not incorporated in the marginal cost/benefit calculations of the Distribution Marketplace. Certainly, recouping these capital expenses is a key component in practical implementation of a Distribution Marketplace. However, instead of incorporating this aspect into the market design itself, the results of a simulation should be used to provide the boundaries on capital investment. For example, the DTDM does not incorporate the amortization of PV investment and demand curves are expected to reflect marginal costs, when provided by a rational actor. The results of simulation will then provide insights on the revenue provided by the PV array. It is this return that should be compared to the amortization requirements.

Two, the Distribution Marketplace only balances clearing energy within the system. It assumes that all system components operate within nominal parameters during the market interval. For example, PV inverters are assumed to manage voltage at the appropriate level. Additionally, it is assumed that component responses to pricing signals will not have any adverse effects on system stability. Both of these assumptions would need to be addressed in real-world applications. For example, if cloud cover suddenly removes PV generation from the system, the price could increase. This increase could be observed by many energy storage units, which all begin operating the discharge mode immediately upon receiving the price signal. If the storage units are not in the same DDS subsystem as the lost PV – for example, if they are on a different feeder – this sudden injection of power could result in overvoltage. This specific example could be resolved in multiple ways: perhaps staggered price signals or storage inverters technical requirements. While not addressed in this proposed market, this aspect must be addressed in any practical implementation.

Three, it is assumed that customers will voluntarily participate in the Distribution Marketplace. This does not address the very real challenges inherent in customer engagement, adoption, training, and retention. It is assumed that customers will chose to join the marketplace. Additionally, it is possible to design the DTDM network such that non-participating customers are “firewalled” at pre-established, legacy energy rates. In this case, the DTDM supernode could estimate the load of this customer, based on historical data, and submit it to the market at an inelastic curve.

Four, by impacting a regulated utility, there will be valid concerns of cost allocation and customer equity. Both these concerns are valid, but not addressed in the marketplace design itself.

Five, reasonably accurate prediction and rational hedging are important for efficient marketplace function. With dynamic tariffs and energy costs, customers and system operators will be faced with challenges in estimating consumption and predicting future energy prices, which themselves are dynamic and tied to consumption estimates. The proposed marketplace, and subsequent simulation, does not address how this will occur, or how effective it should be expected to be. It is assumed that software and specialized algorithms will take advantage of the DTDM to optimize for their customers.

Six, for practical implementation, this system must provide benefits or opportunities for all existing stakeholders in the electric energy sector. This includes utilities, regulator, customers, and DER aggregation providers. The marketplace is designed to enable benefits for all stakeholders, but actual results will depend on each specific application. The marketplace design itself assumes this is not a barrier to implementation.

4 Data Sources

In the DTDM simulation, time-varying datasets will be needed. This data is provided by two sources. Baseline load and PV generation uses data from Pecan Street Inc. Dataport. Wholesale electricity spot pricing uses data from New York Independent System Operator (NYISO).

4.1 Pecan Street

For the load and PV behavior models, real-world demand and generation data is used as a baseline. The source of this data is Pecan Street Inc. Dataport [17], an energy research organization based in Austin, TX. Note, the Pecan Street data is used solely as a baseline for behavior models. In no way is this data fundamental to the DTDM concept of the general approach used in simulation or the behavior models.

Pecan Street, Inc. Dataport provides anonymized, historical minute-by-minute electrical energy demand for households, broken into specific appliance and general usage categories. Measurements were recorded by eGauge devices, providing the average power (in kW) over each one minute period. There are 1295 households in the Pecan Street dataset.

Each household has eGauge data in up to 66 categories. Of these 66 categories, *grid* comprises the household's power drawn from the electrical grid, *gen* comprises the power generated by a solar PV system, and *use* comprises the whole home power use (i.e. *use* = *gen* + *grid*).

The remaining 63 categories provide information on household demand. Categories specify appliances, such as *refrigerator1*, or circuits, such as *bathroom1*. The category *car1* denotes Electric Vehicle charging. However, not all electrical demand is allocated to a specific category. As a result, for a given household, the sum of categorized demand entries may be less than the *use* entry. This sum of entries will never exceed the *use* entry, however. Additionally, households may include loads that were not specifically prioritized in the eGauge system, despite matching an existing category.

Within the dataset, 710 households provide minute-by-minute data for the *use* category. The following table examines the frequency of each of these load categories within those 710 households. In addition to the numbers provided in the Pecan Street dataset, each category has been labeled with its anticipated elasticity type. As previously described, loads can be generally grouped into four categories, based on how price signals could impact the demand profile: Inelastic (I) when price signals will not impact the demand profile; Thermal (T) when thermal storage enables short-term demand shifting; Schedulable (S) when demand can be anticipated or delayed autonomously; and Behavior Adjustment (B) when customer could be reasonably expected to modify their demand profile to reduce their energy costs. These possibilities are denoted in the Elast column below. Note, all four categories could also be subject to increased or decreased total consumption based on price signals, in addition to load profile shifting.

Category	#	%	Elast	Category	#	%	Elast
air1	604	85.1%	T	kitchenapp1	381	53.7%	I
air2	86	12.1%	T	kitchenapp2	272	38.3%	I
air3	11	1.5%	T	lights_plugs1	177	24.9%	I
airwindowunit1	5	0.7%	T	lights_plugs2	84	11.8%	I
aquarium1	2	0.3%	I	lights_plugs3	35	4.9%	I
bathroom1	158	22.3%	I	lights_plugs4	13	1.8%	I
bathroom2	16	2.3%	I	lights_plugs5	4	0.6%	I
bedroom1	127	17.9%	I	lights_plugs6	2	0.3%	I
bedroom2	51	7.2%	I	livingroom1	174	24.5%	I
bedroom3	10	1.4%	I	livingroom2	13	1.8%	I
bedroom4	1	0.1%	I	microwave1	353	49.7%	I
bedroom5	1	0.1%	I	office1	62	8.7%	I
car1	92	13.0%	S	outsidelights_plugs1	37	5.2%	I
clotheswasher1	356	50.1%	B	outsidelights_plugs2	6	0.8%	I
clotheswasher_dryg1	63	8.9%	B	oven1	204	28.7%	I
diningroom1	35	4.9%	I	oven2	6	0.8%	I
diningroom2	2	0.3%	I	pool1	6	0.8%	S
dishwasher1	496	69.9%	S	poollight1	3	0.4%	I
disposal1	281	39.6%	I	poolpump1	32	4.5%	S
drye1	348	49.0%	B	pump1	8	1.1%	S
dryg1	101	14.2%	I	range1	233	32.8%	I
freezer1	23	3.2%	T	refrigerator1	525	73.9%	T
furnace1	557	78.5%	T	refrigerator2	39	5.5%	T
furnace2	59	8.3%	T	security1	20	2.8%	I
garage1	53	7.5%	I	shed1	4	0.6%	I
garage2	4	0.6%	I	sprinkler1	18	2.5%	S
heater1	7	1.0%	T	utilityroom1	11	1.5%	I
housefan1	3	0.4%	I	venthood1	44	6.2%	I
icemaker1	7	1.0%	S	waterheater1	116	16.3%	T
jacuzzi1	41	5.8%	B	waterheater2	4	0.6%	T
kitchen1	89	12.5%	I	winecooler1	8	1.1%	T
kitchen2	33	4.6%	I				

Table 1: Pecan Street Use Categories [17]

Next, as an illustration, one specific household will be examined. Minute-by-minute eGauge data for 2014 is analyzed and includes categories: *air1, dishwasher1, disposal1, furnace1, grid, kitchen1, livingroom1, range1, refrigerator1*, and *use*. Note, there are no loads categorized as "bedroom". This does not mean the apartment does not have bedroom loads, only that the eGauge did not specifically monitor those loads. This uncategorized component of *use* is denoted in the table.

Of the available categories, *air1*, *furnace1*, and *refrigerator1* are considered thermal elastic loads. Further, *dishwasher1* could be considered a schedulable elastic load. The remaining loads, by default, should be considered inelastic.

For each category the average power is the taken over the entire year; the maximum is the largest one-minute power consumption recorded. For the aggregate categories, the average and maximum are calculated using coincident power entries. As a result, the maximum coincident power provided to Thermal Elastic loads is less than the sum of maximum power provided to each Thermal Elastic load.

Additionally, for the elastic load categories, the average and maximum percentage of demand is calculated. This is based on the minute-by-minute contribution of each elastic category to the aggregate demand.

Category	kWh/year	kWh %	avg(kW)	max(kW)		
use	3868.70	100.0%	0.442	7.747		
air1	1503.45	38.9%	0.172	1.534		
dishwasher1	15.22	0.4%	0.002	0.772		
disposal1	0.43	0.0%	0.000	0.453		
furnace1	248.52	6.4%	0.028	4.622		
kitchen1	119.80	3.1%	0.014	1.983		
livingroom1	439.36	11.4%	0.050	0.396		
range1	258.92	6.7%	0.030	3.275		
refrigerator1	530.57	13.7%	0.061	1.331		
non-categorized	752.44	19.4%	0.086	2.499	avg(%kW)	max(%kW)
Inelastic	1570.95	40.6%	0.179	5.971	59.0%	100%
Thermal Elastic	2282.54	59.0%	0.261	5.866	40.7%	99.9%
Schedulable Elastic	15.22	0.4%	0.002	0.772	0.00%	93.0%

Table 2: Pecan Street Category Data - Example Household Annual Summary [17]

The loads in this example household indicate opportunities for thermal elastic load flexibility. One, these loads provide a large total portion of energy consumed in 2014. If a household HEM adjusted these loads to reduce their energy cost, the impact would be seen on the household's electricity bill. Two, the thermal elastic loads constitute a large portion of minute-by-minute power demand, both in raw quantities and as a percent of aggregate demand. In fact, there are instances in which thermal load demand constitutes nearly the entire aggregate demand. As a result, shifting these thermal loads can have a significant impact on the household's total energy demand and aggregate power demand.

However, in this example household there are limited opportunities for schedulable elastic load flexibility. This category is limited to *dishwasher1*, which has both a low proportion of energy and power. Despite this category constituting 93% of aggregate demand at some point in time, the average contribution to power demand is nearly zero. It cannot be expected that schedulable elastic loads, in this household, will have an impact on total energy demand or instantaneous power demand.

Due to the size of the Pecan Street dataset, simulation will use *air1* as the stand-in for all elastic loads. This category is used because it is very common and contributes strongly to both total energy and instantaneous power demand. Confining elasticity to this category will illustrate the impact of load flexibility, but it should be recognized that additional elasticity may be incorporated, both in simulation and application.

Consequently, the aggregation of inflexible loads will generally be described by subtracting *air1* from *use*. As stated previously, *use* does not include any household PV generation.

For PV simulation, the Pecan Street category *gen* will be used. No effort will be made to maintain a connection between a household's *use* and *gen* data. It is not expected that customers in the Pecan Street dataset modified their behavior based on their household PV generation. Further, although daily weather fluctuations impact PV generation and household energy consumption (e.g. air conditioner demand), the DTDM simulation does not delve to that level of detail. As such, these data will be considered fully independent.

Two adjustments are made to the data before implementing into the simulation. One, load data occasionally includes negative load values. This disrupts the Load Node behavior model, as described in Section 6.2. All negative load data quantities will be rounded to zero in the simulation. Two, PV data occasionally includes negative generation values. This disrupts the PV Node behavior model, as described in Section 6.3. All negative generation quantities will be rounded to zero in the simulation.

4.2 NYISO

For wholesale pricing, case study simulations use New York Independent System Operator (NYISO) data. NYISO provides of detailed and accessible information on its Markets & Operations website [18]. NYISO provides access to both Day Ahead Market (DAM) and Real Time Dispatch (RTD) prices. RTD prices reflect the dynamic marginal costs of energy, so those are used in the simulation. Additionally, NYISO provides Locational Based Marginal Prices (LBMP) for each of its eleven zones. Simulation uses pricing data from the CAPITL zone. Finally, the pricing data obtained spans calendar year 2014.

Each RTD LBMP data entry includes an End Time Stamp. This reflects the period for which the RTD price applied. In most cases, each RTD price applied for five minutes, which reflects NYISO's standard RTD market period. However, in some events the RTD price is updated within the five-minute period. This could be interpreted as attempts to compensate for immediate, unexpected supply-demand imbalance.

LBMP prices include energy, loss, and congestion components. As described previously, these components are important to wholesale dispatch of generation. However, to customers within the LBMP zone, the net LBMP is the effective price of energy; this is the price used in simulation. The presented simulations assume the DTDM is a price taker and has no influence on the wholesale market. Wide-scale DDS deployment and participation of DTDMs in the wholesale market would alter this assumption. This level of analysis is not provided in this thesis.

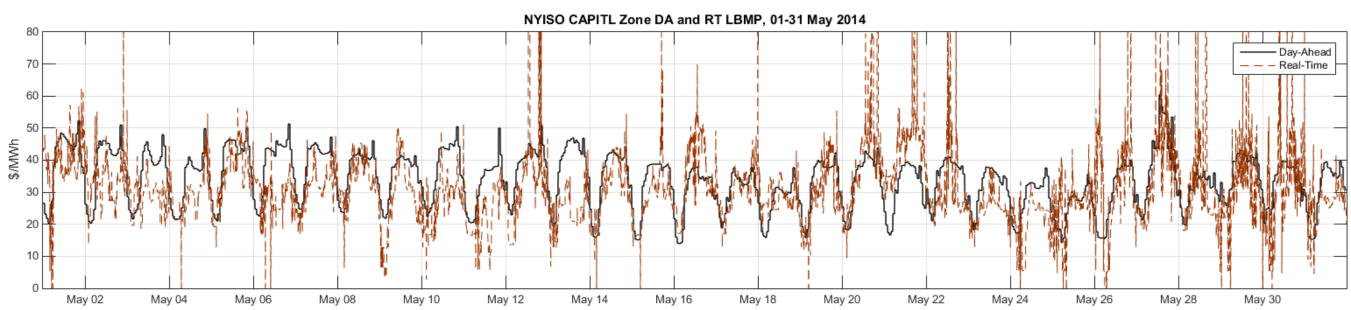


Figure 25: NYISO DA and RT LBMP Comparison [18]

RTD End Time Stamp	Zone Name	Zone PTID	RTD Zonal LBMP	RTD Zonal Losses	RTD Zonal Congestion
1/1/2014 1:55	CAPITL	61757	17.95	1.11	0
1/1/2014 2:00	CAPITL	61757	54.69	3.58	0
1/1/2014 2:05	CAPITL	61757	62.41	1.97	-32.73
1/1/2014 2:10	CAPITL	61757	51.85	2.34	-16.09
1/1/2014 2:15	CAPITL	61757	60.3	2.29	-25.25

Table 3: Example NYISO RT LBMP Data [18]

Figure 25 illustrates the volatility of the RTD prices. This is useful variation; the more dynamic the wholesale energy prices, the more opportunities for the DTDM to dynamically adjust energy consumption. Additionally, Figure 25 illustrates the relative stability of DAM prices; this reflects the hedging accomplished the forward markets. In a simplification, the RTD prices can be interpreted as being needed to provide energy unanticipated by the day-ahead forward market. The DAM prices will not be used in the DTDM simulation.

The NYISO LBMP does not include all costs associated with purchasing energy in the wholesale market and transporting it to the end-use customers. Significantly, it does not include transmission and distribution (T&D) costs or any distribution system operations and maintenance (O&M) costs. These may be fixed costs or variable, volumetric costs. However, the DTDM simulation's goal is to demonstrate how the network will respond to reasonably variable price signals. To this end, the NYISO LBMP is an appropriate stand-in for an effective real-time dynamic energy price. If the DTDM simulation was used to represent an actual DDS deployment, the LBMP values would need to be adjusted to reflect these additional costs.

5 Simulation: Matlab Implementation

5.1 Overview

There is value in modeling a Distribution Marketplace. Simulation enables comparing deployment locations and market configurations. To this end, the Dynamic Tariff Distribution Marketplace has simulated in Matlab.

The DTDM includes both energy flow and cash flow. It includes ownership of assets, network characteristics, market communication and decision protocols, and customer behavior models.

The DTDM simulation includes simplified behavior models for actors within the system. The design and limitations of these models is examined in Section 6. Additionally, the DTDM simulation does not implement all possible tariff configurations and system parameters. Instead, it proposes example configurations that represent common system goals.

As a result, the current DTDM simulation is a demonstration of the market protocols and an examination of the interaction between actors within the DTDM. It is not designed to prove the end results of DTDM deployment. Instead, the simulation should be considered a basic proof of concept and starting point for further research and development.

DTDM simulation in Matlab consists of initialization, simulation, and analysis. This process is outlined in chronological order below. Each of the steps will be examined in detail.

4.1 **Create DDS Object**

4.2 **Create Node Objects**

4.2.1 Construction and General Parameters

4.2.2 Demand Curve

4.2.3 Linkage Loss Constant

4.2.4 Power Factor

4.3 **Create Tariff Objects**

4.4 **Prepare Simulation**

4.4.1 Validate DDS

4.4.2 Define Relationships

4.4.3 Pre-Allocate Arrays

4.5 **Simulation: Market Operation**

4.5.1 Demand Curve Generation

4.5.2 Demand Curve Adjustment

4.5.3 Demand Curve Translation

4.5.4 Demand Curve Aggregation

4.5.5 Market Clearing Point

4.5.6 Node Marginal Price

4.6 **Simulation: Real-Time Actions**

4.6.1 Set Bottom-Level Node Consumption

4.6.2 Run Dispatcher Nodes

4.6.3 Determine System-Wide Energy flow

4.7 **Simulation: Settlement**

4.8 **Analyze results**

The DTDM is simulated in Matlab. Custom handle classes have been defined for DDS, nodes, tariffs, and owners object. Each object class has pre-defined parameters and methods. Some parameters are defined prior to simulation, others are updated during simulation. After initializing the system, most calculations and processes are completed by the object methods. However, additional functions, not included in the objects are methods, will be used. These are more general functions that apply in use cases outside of simulation.

The Matlab simulation has been designed to provide flexibility. However, some limitations have been imposed to reduce development time. For example, settlement interval and market duration must be equal. Additionally, simulation is limited to one-phase systems. Indication will be made when limitations beyond the DTDM construct have been applied to the simulation.

5.2 Create DDS Object

The DDS object contains the DTDM network representation, to include all node objects, tariff objects, and linkages. Additionally, the node and tariff objects include their unique parameters and recorded values. As a result, the first step in simulation is defining the DDS object.

```
DDS2 = classDDS;
```

The DDS object includes multiple parameters, which apply to the system and simulation as a whole. Like all Matlab objects, these parameters can be defined and accessed in the following format:

```
DDS2.Qbin = 0.0001;
DDS2.timeseries
```

DDS object parameters include: indexed object listing for system nodes, tariffs, owners, markets, and dispatcher; current simulation timestep; reference datetime values; Pcap, pFRE, Qbin, and Pmin; and total dispatch levels. A complete listing is provided in Appendix A.

Parameters that specify multiple objects, such as system nodes, can be access through indexing. Accessing the node object with index one is accomplished by:

```
DDS2.nodes(1)
```

Additionally, because the DDS object and all node, owner, and tariff objects are handles, indexing can be used to update parameters directly. Accessing the owner object for the node object with index one is accomplished by:

```
DDS2.nodes(1).owner
```

This method enables great flexibility is accessing and updating object parameters. Note, relationships between objects are included as parameters for both objects. For example, the following command will access the initial DDS object, DDS2:

```
DDS2.nodes(1).owner.DDS
```

Finally, objects in Matlab can refer to their own properties with *obj*. This is useful in methods within the class definition. For example, a method within the DDS object can access its node object with index one:

```
obj.nodes(1)
```

When initializing a simulation, three parameters must be immediately defined: *DDS.timeseries*, *DDS.Qbin*, and *DDS.Pmin*. The remaining parameters will be updated at the DTDM system is defined and simulated.

DDS.timeseries is the array of Matlab timeseries values that defines the system time during simulation. This parameter should be defined by the size of source data. The Pecan Street dataset provides minute-by-minute data for 2014. The corresponding definition of *DDS.timeseries* is:

```
DDS2.timeseries = (datetime('01-Jan-2014 00:00:00'):minutes(1):...
datetime('01-Jan-2015 00:00:00'))';
```

In general, this parameter is set for one-minute intervals. However, this is not a limitation in DTDM or the simulation code. This interval between timesteps could be any length, from seconds to hours. However, note any parameters referring to a duration value will need to be adjusted to compensate for the timestep value. All simulations described will use one minute timesteps.

The two additional initialization parameters, *DDS.Qbin*, and *DDS.Pmin*, apply to demand curves and will be described in Section 5.2.2.

Within the DDS object, nodes and owners are indexed in the parameters *DDS.nodes* and *DDS.owners*. However, it is helpful to search for a specific node or owner by the name specified by *node.name* or *owner.name*. This is accomplished with the DDS method *find*, which returns the node or owner object, with the specified string as its name.

```
DDS2.find('NameString')
```

The *find* method also stores the last found object as the parameter *DDS.recent*. This makes further updates to the found object faster. Additionally, the parameter *DDS.recent* is also updated when adding a node, owner, or tariff object to the DDS.

```
DDS2.recent
```

Finally, the DDS object methods *DDS.list* and *DDS.hierarchy* each provide a formatted display of the current DTDM network. *DDS.list* displays the nodes, tariffs, owners, markets, and dispatchers indexed in its object parameters. *DDS.hierarchy* displays all nodes and tariffs with indentations reflecting the sub- and supernode relationships. This method can also be called for any specific node, as *node.hierarchy*, to display only the relationships "below" the specified node.

5.3 Create Node Objects

5.3.1 Construction and General Parameters

After creating the overall DDS object, node objects must be added to the DTDM network. Each is added with the DDS object method *addNode*. For example, to add a node to the DDS object *DDS2*:

```
DDS2.addNode('NameString','OwnerString','SupernodeString')
```

This DDS method creates a node object and adds it to the *DDS.nodes* parameter. The newly created node must have a unique name, which also cannot match any existing Owner object names. If the OwnerString specified does not indicate an existing owner in the system, an Owner object is created with that name. If an Owner object exists with the name specified by OwnerString, the newly created node is added to its *owner.nodes* parameter.

Node ownership can be later be updated with the method *node.setOwner*. This will update the objects for the node, the previous owner, and the new owner.

The third input, *SupernodeString*, is optional. If it is not provided, the newly created node does not have a specified supernode; it will act as a top-level node. If *SupernodeString* is provided, the indicated two nodes are linked together: the new node is added to the supernode object's *node.sub* parameter and the supernode is set as the new node's *node.super* parameter.

A node's supernode can later be updated with the method *node.setSuper*. This will update the objects for the node, the previous supernode, and the new supernode. It will also execute *node.super.setPmin*, which will be described in Section 5.2.2.3.

The object parameter *node.type* specifies the type of node. The current simulation implementation limits this parameter to: Load, PV, Storage, Aggregation, Marketplace, Dispatcher, or Supply. Each node type has different relevant parameters and methods. For example, Load Nodes will reference *node.loadDataSource*, while PV Nodes will reference *node.pvDataSource*. The acceptable node types are either functional node types (e.g. Marketplace Nodes) or pre-established behavior models (e.g. Storage Nodes). Each will be examined in detail during Section 5.4.3 and Section 6.

By default, a newly created node is set to type 'Load'. If another node specifies the node as a supernode, then this is automatically updated to type 'Aggregation'. Assigning a bottom-level node to type 'Storage' or 'PV' requires updating the *node.type* parameter directly. Assigning a non-bottom-level node to type 'Marketplace' requires the method *node.setMarket* and the type 'Dispatcher' requires the method *node.setDispatcher*. The final type, 'Supply' is automatically assigned is the node's subnode is set to type 'Marketplace'.

5.3.2 Demand Curves

A node's current demand curve is stored in the *node.curve* parameter. While this parameter is established during Market Operation, its construction and format is fundamental to the DTDM process, so it will be described here.

Demand curves express nodes' anticipated consumption preferences. When a node submits a demand curve to its supernode, with a different owner, the demand curve is a contractual bid.

A demand curve is a series of points, each representing a quantity (Q) and price (P). Each point is the node making the statement: "If the unit price of energy is P, then I would purchase Q units of energy". When plotted, the quantity is captured on the horizontal axis, while the price is captured on the price axis.

In the Matlab simulation, demand curves are stored in the *classNode* property *curve*. The first column consists of quantities and the second column consists of prices. For a demand curve with n points:

$$node.curve = [Q_1, P_1;\ Q_2, P_2;\ \dots;\ Q_n, P_n]$$

Implementation of the DTDM, whether in simulation or practical application, places some restrictions on demand curves.

5.3.2.1 Qbin

Practical communication between actors and devices in the DTDM requires demand curves to be expressed as discrete points. Each discrete point includes a quantity, *Q*, and a price, *P*, value.

Q (kWh)	P ($/kWh)
.04	.110
.05	.090
.06	.085
.07	.080
.08	.060

Table 4: Example Demand Curve Values

The series of points that comprise a demand curve are controlled by a defined interval of quantities. This is defined as the *Qbin* interval. Each quantity value should be an integer multiple of the defined Qbin. Price values should only be provided for these quantities. For the example demand curve values in Table 4, Qbin = .01.

A demand curve cannot include multiple prices for any Qbin interval; this would violate the requirement that demand curves must be monotonic.

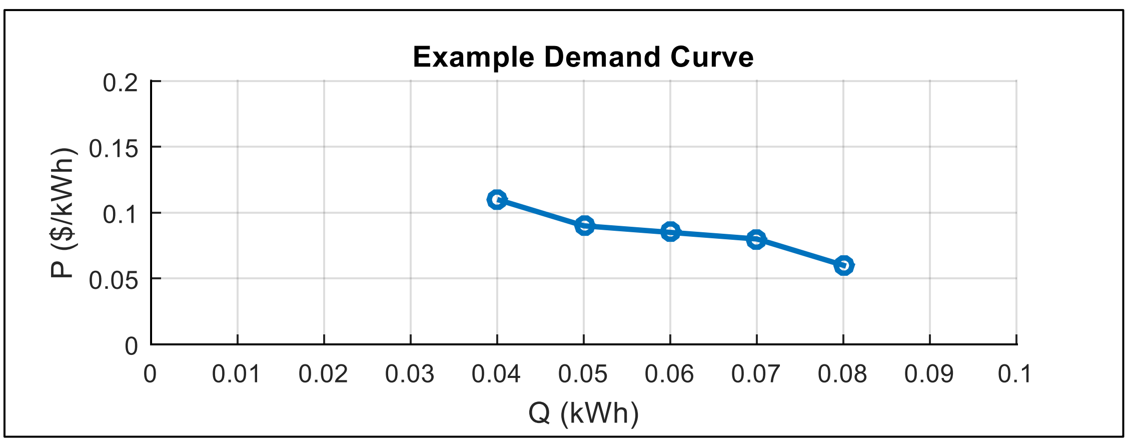


Figure 26: Example Demand Curve

The value of Qbin is limited by the ability of precision of measurement and control devices. As a practical guideline, Pecan Street data provides average power with a precision of 0.001 kW at one-minute increments. This corresponds to a precision of 0.00001667 kWh. A DTDM system using the same devices could not expect more precision than this value, so the DTDM should set Qbin >= 0.00001667 kWh.

Notice, the size of demand curve arrays scales proportionally to 1/Qbin. If this value is set unreasonably low, then there will be a processing burden with no discernable difference in results. With this in mind, Qbin = 0.0001 kWh will commonly be used in simulation. This is a DDS parameter that must be defined when initializing the DDS object.

```
DDS2.Qbin = 0.0001;
```

Demand between quantity-price points is estimated by linear interpolation. For the example demand curve values in Table 4, the node would be expected to consume .045 kWh at a price of $0.10/kWh. This convention should be known by all actors in the system. To limit misrepresentation by linear interpolation, Qbin should be set to the lowest value supported by device precision.

To promote consistent communication between nodes, the Qbin parameter is defined for the system as a whole. However, it is possible for two DTDM systems to interface, when each uses a different Qbin. Converting a demand curve from one system to another would require sampling, using the linear interpretation convention. This is not addressed in the DTDM simulation; in this case, Qbin is a system parameter that is used by all nodes and tariffs. For a DDS object *DDS2*:

```
DDS2.Qbin = 0.0001;
```

In the Matlab simulation, consideration must be given for floating point error, in particular when finding a specific Qbin interval. The following construction may lead to “missed” values:

$$find(curve(:,1) == nQbin)$$

Instead, the find function should look for values relatively near the target Qbin interval. A better construction would be:

$$find(abs(curve(:,1) - nQbin) < Qbin/1000)$$

5.3.2.2 Linear Interpolation

The values between Qbins are determined using linear interpolation. Because demand curves are expected to be both monotonic and strictly descending, it is possible to treat either price or demand as the independent variable.

For a given quantity, *Q*, which lies between adjacent quantities *Q1* and *Q2* specified by the demand curve, the corresponding price, *P*, is determined by:

$$P = P1 + \left(\frac{P2 - P1}{Q2 - Q1}\right)(Q - Q1), \qquad Q1 < Q < Q2$$

Similarly, for a given price, *P*, which lies between adjacent prices *P1* and *P2,* specified by the demand curve, the corresponding quantity, *Q*, is determined by:

$$Q = Q1 + \left(\frac{Q2 - Q1}{P2 - P1}\right)(P - P1), \qquad P1 < P < P2$$

In both case, the estimated point lies between the points [Q1, P1] and [Q2, P2].

5.3.2.3 Pmin

Demand curves are typically non-ascending, based on rational actor behavior models. However, DTDM rules require demand curves, when submitted to a supernode, to be monotonic and strictly descending. This is necessary to ensure deterministic results of the market clearing process. This prevents a demand curve submission from including horizontal segments, despite the fact that horizontal segments may reflect the actor's true preferences. For example, a hot water heater may turn on for at any price below \$0.05/kWh but remain off from any price above \$0.05/kWh. If the hot water heater's energy demand, while on, is 1 kWh, the resulting demand curve includes a horizontal segment at P = \$0.05 from Q = 0 to Q = 1.

However, the system dictates a negative slope between any two points on a demand curve. This slope is defined by the *Pmin* parameter. Pmin is defined as the minimum price difference between any two points on the demand curve. Thus, the minimum demand curve slop is defined to be -Pmin/Qbin.

For the example demand curve values in Table 4, Pmin ≥ \$.005 is satisfied. However, this curve does not meet the requirements for Pmin < \$.005. If the Qbin was smaller, for example .001, than the same curve (with sampling) would require a less strict Pmin, for example Pmin ≥ \$.0005. Thus, there is a relationship between the selection of Qbin and Pmin.

The goal of Pmin is to provide the system operator with the ability to select a clearing price that uniquely indicates a clearing quantity. It is for this reason that horizontal segments are unacceptable. However, because a customer's true preferences may be reflected with horizontal segments, the Pmin value should be as small as possible.

If horizontal segments reflect a customer's true preferences, it is that customer's responsibility to determine the best way to modify their true demand curve to meet the Pmin restriction. Examples are provided in Section 6 - Simulation Behavior Models. Additionally, horizontal segments often indicate an inability to consume intermediate quantities of energy. In the example above, the hot water heater may not have the inherent capability to consume quantities greater than zero but less than one. If a customer receives a NMP that indicates such a quantity on their modified demand curve, it is their responsibility to adjust consumption as they see fit. This may include: ignoring the deviation from their demand curve submission, implementing dispatchable balancing storage, or cycling their load during the market interval.

The Pmin value is based on the precision required by the system software and devices. For example, a Pmin of $1e-30 would provide a strictly descending demand curve; however, the price difference between two adjacent points may be too small to provide an actionable difference.

As a practical guideline, NY ISO publishes LBMPs with a precision of $0.01/MWh, which translates to $0.00001/kWh or $1e-5/kWh. If the wholesale energy price is used in determining the DTDM clearing price, then this should be considered a ceiling for the Pmin. A Pmin larger than $1e-5/kWh may result in incorrect correlation between the wholesale market price and the aggregated DTDM demand.

With this in mind, in simulation, Pmin = $1e-8 will commonly be used as the top-level node Pmin. This is set when initializing the DDS object.

```
DDS2.Pmin = 1e-8;
```

Pmin applies solely to the node at which it is established. Consider the aggregation of two demand curves, each at the minimum slope specified by -Pmin/Qbin. The aggregated curve will be expressed at the same Qbin interval. However, aggregation is the process of horizontal summation; the aggregated curve will have one half the slope of the components curves. This would result in an aggregate curve with a slope specified by -Pmin/2Qbin, which violate the Pmin parameter. To prevent this occurrence, a node with more than one subnode must specify a larger Pmin for its subnodes.

The Pmin value for subnodes is a function of the supernode's Pmin and the number of subnodes possessed by the supernode. This is expressed by:

$$P_{min}^{sub,1} = P_{min}^{sub,2} = \cdots = P_{min}^{sub,m} \geq mP_{min}^{super}$$

In simulation, the DTDM system Pmin applies to the top-level node. The Pmin parameter for all other nodes is determined by applying the above equation as equality. As a result, the addition of a new node results in an updated Pmin for all the nodes below new node's supernode. Specifically, when a node is created, it calls the method *setPmin* for its assigned supernode. This initiates an update below the assigned supernode:

```
setPmin(obj.super,obj.super.Pmin);

function setPmin(obj,newPmin)
...
obj.Pmin = newPmin;
if not(isempty(obj.sub))
      for n = 1:size(obj.sub,2)
            setPmin(obj.sub(n),newPmin*size(obj.sub,2));
```

```
        end
    end
    ...
    end
```

Consequently, establishment of the system Pmin for simulation should consider the overall size of the network and resulting Pmin at bottom-level nodes.

In practical implementation, the Pmin parameter for bottom-level nodes should not change when adding unrelated nodes to the DTDM. This is accomplished by exceeding the inequality in the above equation. For example, if a node has 15 subnodes and anticipates the addition of additional subnodes, it may elect to set the Pmin for all subnodes at 30*Pmin. This would satisfy the above equation for 15 additional subnodes, without impacting parameters for existing subnodes.

Finally, tariff curves are not restricted by Pmin. They are simply required to be non-descending and may contain horizontal segments. This exception recognizes that tariff curves serve only to modify a demand curve. If a demand curve meets the Pmin requirement, any adjustments based on non-descending tariff curves will also meet the Pmin requirement.

5.3.2.4 Pcap

A demand curve is a series of points, each representing a quantity (Q) and price (P). Each point is the node making the statement: "If the unit price of energy is P, then I would purchase Q units of energy". From this interpretation, it is expected the left-hand (LH) limit of a demand curve extends upward vertically and the right-hand (RH) limit of a demand curve extends downward vertically.

However, vertical segments, like horizontal segments violate the DTDM demand curve rules. Specifically, a demand curve with vertical segments is not monotonic.

To represent LH and RH demand curve limits, while remaining monotonic, these vertical segments are approximated with very large positive and negative prices, respectively. It is helpful for this price to be a system parameter: *Pcap*. When Pcap is a system parameter, nodes need not consider the LH and RH limits in their submission, only their minimum and maximum consumption.

The Pcap parameter should be specified for the DDS object during initialization. If not specified, the default value is 100.

```
DDS2.Pcap = 100;
```

For an expressed demand curve, two points are added to the demand curve:

$$[Q, P] = [Q_{min} - Qbin, Pcap;\ Q, P]$$

$$[Q, P] = [Q, P;\ Q_{max} + Qbin, -Pcap]$$

These additions are shown below, for the previously provided example demand curve values.

Q (kWh)	**P ($/kWh)**
.03	100
.04	.110
.05	.090
.06	.085
.07	.080
.08	.060
.09	-100

Table 5: Example Demand Curve Values with Pcap

The value of Pcap should be considerably large, relative to possible DTDM prices. This is to prevent bottom-level nodes from being expected to consume less than their minimum quantity or more than their maximum quantity. In the example above, it could be reasonable for the DTDM NMP to reach $2/kWh. From the linear interpolation equation above, this NMP can be expected to result in the following node consumption:

$$Q = .03 + \left(\frac{.04 - .03}{.11 - 100}\right)(2 - 100) = .039810$$

As expected, this quantity is very nearly the node's express minimum quantity of .04 kWh. In fact, it is not even 2% of a Qbin less than the minimum expressed quantity. Because the Qbin interval is expected to measure energy consumption with precision, this is an acceptable error. Notice the percent error, relative to Qbin, is proportional to $(NMP - Price)/Pcap$. In other words, the optimal selection of Pcap does not rely on the actual value of Qbin, only the acceptable percent error.

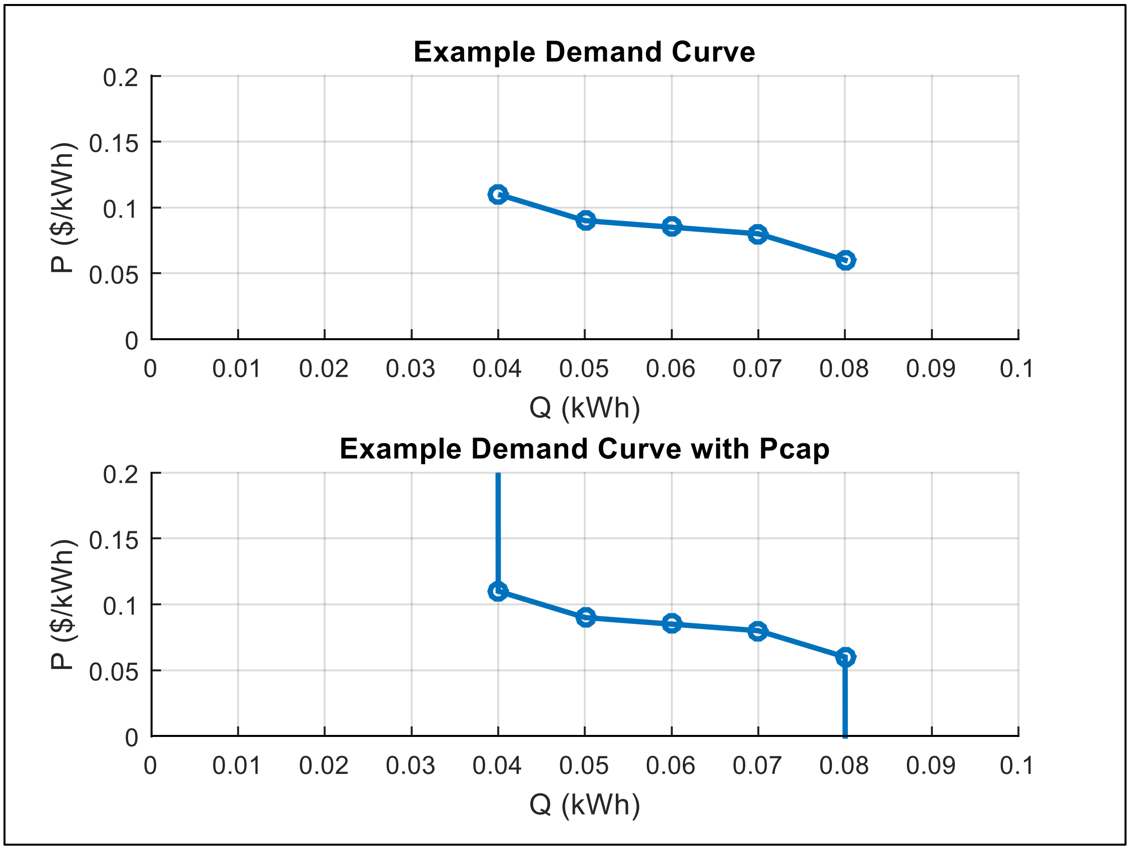


Figure 27: Example Demand Curve with Pcap at Limits

Pcap may also be used in other roles within the network. For example, approaching Pcap may indicate critical loading events, triggering automatic load-shedding functions. This may be useful when preparing for unscheduled islanding events. These uses of Pcap will not be examined further.

5.3.3 Linkage Loss Parameter

When a newly created node is linked to a supernode, the physical losses between the two nodes must be parameterized. This is the linkage loss parameter:

```
node.lossK
```

In the current simulation, this is a constant value, set during initialization. As will be seen, this parameter may be better served as a function of the subnode power factor. However, power factor is not tracked in the current simulation, so the linkage loss parameter must be estimated.

Additionally, this parameter relates to the linkage between a given subnode and supernode. The DTDM construct specifies this parameter should be estimated by the supernode, not the subnode. However, because the subnode only includes one supernode linkage, but the supernode may include multiple subnode linkages, this parameter is store with the subnode.

The linkage loss parameter should captures the impact of physical energy losses by translating a subnode's energy quantity (energy received) into a supernode's energy quantity (energy delivered).

Physical energy losses are a function of energy delivered. In general, this is described as

$$Q_{loss} = kQ_{delivered}^2$$

where k is an approximated linkage loss constant, *node.lossK*. Notice, this definition is defined by the quantity delivered to the subnode, as opposed to the quantity supplied by the supernode.

Practically, the linkage loss constant would be determined through empirical testing and modeling the specific network component over the expected operational range. However, this can also be estimated with the following method. For the following steps, actual units will be used.

Active power is expressed in terms of the magnitude or voltage, current, and the power factor

$$P = V * I * pf$$

Assuming nominal voltage, a power measurement provides the current

$$I = \frac{P}{V_{nom}pf}$$

Active power losses are provided by

$$P_{loss} = I^2 R$$

$$P_{loss} = \left(\frac{P}{V_{nom}pf}\right)^2 R$$

Assuming constant power, an energy quantity (in kWh) for a one-minute interval provides active power

$$P_{requested} = Energy_{requestedKWh} * \frac{60}{1000}$$

This is then substituted into the active power loss equation

$$P_{loss} = \left(\frac{Energy_{requestedKWh} * 60}{1000 * V_{nom} * pf}\right)^2 R_{pu}$$

Again assuming constant power during the market period interval, active power losses can be expressed as energy losses

$$\frac{Energy_{lossKWh} * 60}{1000} = \left(\frac{Energy_{requestedKWh} * 60}{1000 * V_{nom} * pf}\right)^2 R$$

Expressing energy in kWh with the variable Q and rearranging terms yields

$$Q_{loss} = \left(\frac{1000}{60}\right) Q_{requested}^2 \left(\frac{60}{1000 * V_{nom} * pf}\right)^2 R = \left(\frac{60 * R}{1000 * V_{nom}^2 * pf^2}\right) Q_{requested}^2$$

These terms are then matched into our definition for the loss factor

$$Q_{loss} = kQ_{delivered}^2$$

Providing and estimated linkage loss constant

$$k = \left(\frac{60 * R}{1000 * V_{nom}^2 * pf^2}\right)$$

This assumes nominal voltage, constant power and losses over the market period interval, and a constant power factor. Empirical testing will reveal if these assumptions are reasonable. If not, this parameter may need to be adjusted dynamically by the supernode. In particular, estimated power factor may need to be included with a subnode's demand curve submission.

5.3.4 Power Factor

The current simulation does not include any consideration for power factor. Pecan Street load and PV data do not indicate power factor, only active power. Additionally, wholesale energy prices are expressed in active power energy consumption. During simulation, it is assumed that all nodes operate with reasonable power factors.

However, as described, the linkage loss parameter is a function of the subnode's power factor. For this calculation, a reasonable power factor is assumed, such as 0.9. This value is

When the DTDM is practically implemented, if the power factor cannot be reasonably estimated with a constant value, then the subnode could be required to provide estimated power factor values over the range of quantities in their demand curve. This has not been implemented in the current simulation, but is considered here.

In addition to a price value, every point on the demand curve would now include a power factor. This value must also indicate leading or lagging. Alternatively, the tracked parameter could simply be θ, to prevent any sign confusion.

As a consequence, the translation of a demand curve is accomplished with a non-constant lossK parameter. However, the DTDM requires the translated demand curve to be monotonic and strictly descending. It is conjectured that, in most practical cases, a strictly decreasing demand curve with a varying power factor will remain strictly decreasing after being translated with a non-constant lossK value. This relies on the observation that the demand curve quantities will not be subject to step changes in power factor; the power factor of the next incremental quantity value by necessity depends on the previous power factor.

Note, with power factor being tracked within demand curves, it must also be corrected during aggregation and translation. This process is not considered at this time.

5.4 Create Tariff Objects

In addition to creating node objects, tariff objects may be added to the DTDM network. Each tariff object is added with the DDS object method *addTariff*. For example, to add a node to the DDS object *DDS2*:

```
DDS2.addTariff('SubNameString','TypeString','OwnerString')
```

This DDS method creates a tariff object and adds it to the *DDS.tariffs* parameter. The newly created tariff does not have a name; it is described by the subnode that bears it. This subnode is specified by SubNameString, which must be an existing node in the DTDM network.

If the OwnerString specified does not indicate an existing owner in the system, an Owner object is created with that name. If an Owner object exists with the name specified by OwnerString, the newly created node is added to its *owner.nodes* parameter.

The variable TypeString must match one of the pre-defined tariff structures. Currently, this is limited to 'capacity', 'rampSMA','rampEMA', or 'flat'. Additional tariff structures are not limited by the DTDM or the underlying code; they simply must be configured under a new tariff type.

The specified tariff type will determine which other parameters must be specified during initialization. For example, Capacity Tariffs require *tariff.capPwrLimit*, while RampEMA Tariffs require will *tariff.rampEMAnPeriod*. Tariff types and structures will be described in detail in Section 6.

5.5 Prepare Simulation

After defining the DDS object and adding the Node and Tariff objects, the simulation is ready to execute. Execution of the simulation is accomplished through a DDS object method:

```
DDS2.run(tRefStart,simDuration);
```

The input variable *tRefStart* provides the starting time value. This is an index value for

DDS.timeseries. The input variable *simDuration* provides the number of timesteps for which to execute the simulation. Recall, one minute timesteps are assumed; this is defined in *DDS.timeseries*.

When *DDS.run* is called, it performs a series of preparation steps before beginning the timestep simulation. These steps are validation, owner edge identification, and pre-allocation.

First, the DDS object method *DDS.validate* is executed. This method checks for common errors or inconsistencies that may have been made when configuring the DTDM network and initializing the simulation. This method should be updated as the simulation code develops. Additionally, *DDS.validate* may be used to enforce limitations that are not inherent in the DTDM market.

Currently, the validation process performs the following checks:

- Verify only one top-level node exists.
- Verify only one market exists and it is either the top-level node or its supernode is the top-level Supply node. This is not a limitation of the DTDM, but a limitation of the current simulation code.
- Verify dispatcher nodes have exactly one dispatchable node. This is not a limitation of the DTDM, but a limitation of the current simulation code.
- Verify dispatchable storage nodes have a dispatcher and the configuration is valid.
- Verify all DDS nodes and tariffs have owners assigned.
- Verify bottom-level nodes (i.e. Load, PV, Storage) do not have subnodes assigned.
- Verify *tRefStart* + *simDuration* does not exceed the size of *DDS.timeseries*
- Compare all node data source arrays to *DDS.timeseries*; sizes must match.

- Check data source entries for valid values: *loadDataSource* must be non-negative, *pvDataSource* must be non-positive. This is not a limitation of the DTDM, but a limitation of the Load and PV Node behavior models.
- Verify the DDS has never been run. Currently, no explicit method to "reset" the simulation has been implemented.

These checks are performed in sequential order. If a check fails, the code ends with an error message. If no checks fail, the simulation continues.

Next, the method *owner.edgeSet* is called for each owner in the system. This method defines each owner's "edge" nodes and tariffs, based on the final DTDM network configuration. An edge nodes and tariffs either indicate energy generation, energy flow to another party, or energy consumption. Determining these nodes will make it possible to easily determine each owner's net energy and cash flow when the simulation is complete; it ignore all energy and cash flow internal to an owner's subnetwork.

There are four classes of edge nodes, each documented in a different parameter. *owner.edgeNodesA* is populated with top-level nodes; this indicates possible energy generation. *owner.edgeNodesB* is populated with nodes with a different owner subnode; this indicates possible energy transfer. *owner.edgeNodesC* is populated with bottom-level nodes; this indicates possible energy consumption. *owner.edgeNodesD* is populated with nodes with at least one different owner subnode; this indicates possible energy transfer. The different owner subnodes are listed in *owner.edgeNodesDSubs*. *owner.edgeNodesE* is populated with nodes that bear one or more tariffs with a different owner; this indicates possible cash flow. The different owner tariffs are listed in *owner.edgeNodesETariffs*.

Finally, the all meter parameters are pre-allocated. These parameters are used to store the results of each simulation timestep. Only nodes and tariff objects with *obj.mFlag = 1* will record the results of each timestep; however, the default value for this parameter is 1. The parameter *DDS.meterTime* is filled with the values in *DDS.timeseries* indicated by the simulation input variables *tRefStart* and *simDuration*. Meter parameters for nodes and tariffs do not yet have values to record; these arrays are pre-allocated with zeros to speed up the Matlab code. For nodes, the following parameters are pre-allocated: *meterPriceContr, meterQuantContrAvg, meterPriceClear, meterQuantClearAvg, meterQuantActual, meterSubRevenue, meterSuperPayment, meterTariffPayment, meterRevenue, meterStoragePct*. For tariffs, the following parameters are pre-allocated: *meterQuantActual, meterRevenue, meterPriceEff*.

Meter parameters can be expanded to include are timestep-varying values of interest.

With validation, edge identification, and pre-allocation complete, the simulation can begin. This process simply performs the number of timesteps indicated by *simDuration*. Each timestep includes Market Operation, Real-Time Actions, and Settlement. A description of each follows.

5.6 Simulation: Market Operation

The DTDM rules do not specify when Market Operation occurs, relative to the market period for which it applies. For example, Market Operation for the 12:00 – 12:05 market period may occur at 11:55 or 11:45. Selecting the timing of Market Operation is the responsibility of the DTDM system operator. However, in the DTDM simulation, Market Operation always occurs immediately before the market period for which it applies. For the 12:00 – 12:05 market period, the DTDM simulation will execute Market Operation after the end of 11:59 but before any actions occur for 12:00.

Thus, the first step for any simulation time step is to determine if a DTDM Marketplace Node is scheduled to run. This requirement is stored in the Marketplace Node parameter *node.marketNext*. If this value matches the current timestep, which is stored in *DDS.t*, the Marketplace Node initiates the Market Operation method, *node.runMarket*. From the DDS object:

```
for n = 1:size(obj.markets,2)
      if (obj.t == obj.markets(n).marketNext)
            obj.markets(n).runMarket;
      end
end
```

The following subsections describe the process that occurs when *runMarket* is called. This includes: requesting demand curves; initiating bottom-node demand curve generation; initiating tariff curve generation; demand curve aggregation, adjustment, and translation; determining the market clearing point; and propagating a Node Marginal Price to all DTDM nodes.

If no market is scheduled to run for the current timestep, then the simulation immediately proceeds to Real-Time Actions. This is described in Section 5.6.

5.6.1 Demand Curve Generation

For Market Operation, the Marketplace Node seeks to determine its node-specific clearing price and quantity. The clearing price will then be propagated, with adjustments, to all nodes in the system: this is each node's Node Marginal Price (NMP) for the market period. To accomplish this, the

Marketplace Node must have a demand curve that reflects the system's energy consumption preferences.

The Marketplace node executes method *node.generateCurve* for the market period specified by input variables *marketStart* and *marketDuration*.

```
obj.generateCurve(obj.marketStart,obj.marketDuration);
```

The method *node.generateCurve* is generic for all node types. First, it clears the currently stored curve parameter:

```
obj.curve = [];
```

Next, *generateCurve* considers the node type.

Bottom-level nodes (i.e. Load, PV, and Storage Nodes) will generate a demand curve based on their specified behavior model. In these cases, *obj.generateCurve* calls methods *obj.setLoadCurve, obj.setPVCurve,* and *obj.setStorageCurve*, respectively.

Supply Nodes, while not bottom-level nodes, will also directly generate a demand curve within *obj.generateCurve* by calling *obj.setSupplyCurve*. However, these nodes are a special case in Market Operation and will be addressed later.

However, Aggregation, Marketplace, and Dispatcher Nodes do not directly generate a demand curves. These node types all have subnodes; each requires a demand curve that reflects the preferences of their subnodes. For these node types, *obj.generateCurve* initiates a demand curve request to each subnode. This is the method *obj.requestCurve*.

```
		for n = 1:size(obj.sub,2)
			obj.requestCurve(obj.sub(n),marketStart,marketDuration);
		end
```

In practical implementation, this request would be communication between two nodes. If both nodes share an owner, this would likely be accomplished with a virtual request. If both nodes do not share an owner, this would be a communicated request.

In simulation, the method *obj.requestCurve* simply calls *obj.generateCurve* for the specified node object.

```
	function requestCurve(obj,targetNodeObj,marketStart,marketDuration)
		...
		targetNodeObj.generateCurve(marketStart,marketDuration);
		...
	end
```

This represents the subnode receiving a demand curve request and running its internal demand curve generation function. As before, *obj.generateCurve* may result in either a direct demand curve generation (for bottom-level nodes) or further requests for subnode demand curve submissions (for non-bottom-level nodes). This process propagates until all bottom-level nodes receive a demand curve request.

When a bottom-level node receives a demand curve request, it generates a curve based on its behavior model. The behavior models included in the DTDM simulation are addressed in Section 6. For all behavior models, *obj.generateCurve* results in the setting the bottom-level node's demand curve parameter *obj.curve*. Next, the bottom-level demand curves will propagate back to the Marketpalce Node.

The Demand Curve Generation request process is illustrated in Figure 28 below. Figures 30 and 43 will illustrate the successive processes.

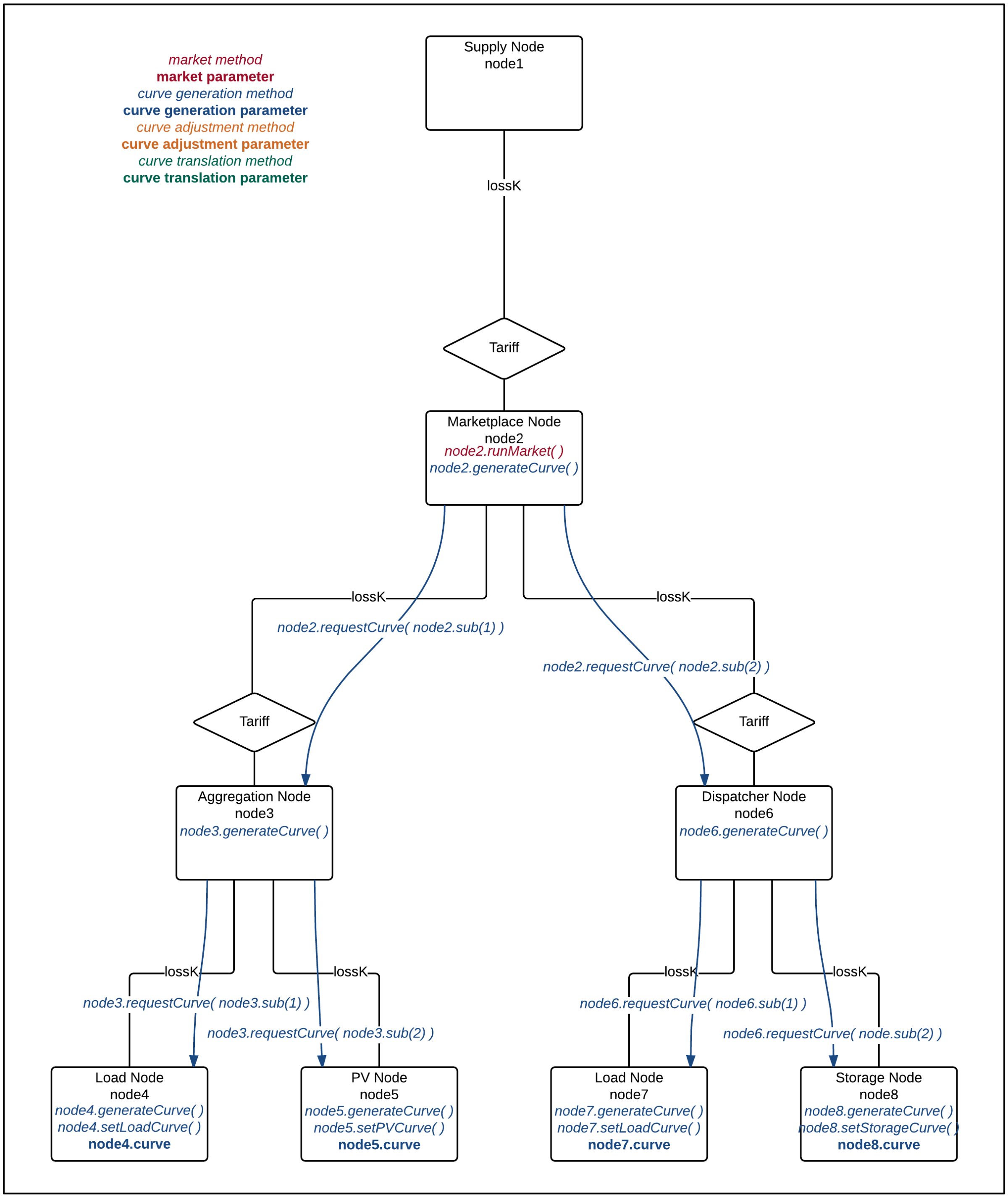


Figure 28: runMarket (1/3)

5.6.2 Demand Curve Adjustment

Next, the bottom-level node must submit a demand curve to the supernode. This is not simply the demand curve the node generated. The demand curve submission is a contractual offer. The bottom-level node must incorporate the impact of any tariffs in this submission. This is the process of demand curve adjustment. Demand curve adjustment is implemented in the method *node.submitCurve*. The process also applies when a non-bottom-level node submits a demand curves to its supernode.

In the DTDM simulation, it is assumed that generation of a demand curve is always followed by demand curve submission. Thus, the method *node.submitCurve* is included in the method *node.generateCurve*, after the node-specific demand curve generation function (e.g. *node.setLoadCurve*, *node.setPVCurve,* or, as later described, *aggregatecurves*).

The purpose of *node.submitCurve* is to adjust *node.curve* based on borne tariffs, with the resulting curve described by the parameter *node.curveSubmit*. This is the contractual curve that is submitted to the supernode.

If no tariffs exists between the node and its supernode, the demand curve submission simply reflects the node's actual demand curve.

```
curvePlaceholder = obj.curve;
...
obj.curveSubmit = curvePlaceholder;
```

If tariffs exist between the node and its supernode, the node must first request an updated tariff curve from the tariff object. This is accomplished by calling the method *tariff.generateCurve* for each tariff object. In most cases, this is accomplished as follows, from the node object:

```
tarifflist = obj.tariffs;
...
for n = 1:size(tarifflist,2)
      tarifflist(n).generateCurve(marketStart,marketDuration);
      ...
```

The method *tariff.generateCurve* develops a tariff curve based on its pre-definted node type. For example, Capacity Tariffs call *tariff.setCapCurve* and Flat Rate Tariffs call *tariff.setFlatCurve*. The specific tariff types included in the simulation are described in Section 6.

Once generated, each imposed tariff curve can be interpreted as a supply curve. Like the node demand curve, these tariff curves are a series of points at Qbin intervals. Curve adjustment is accomplished in two steps.

First, the output curve is "trimmed," or limited to the overlapping quantities of the two curves. This is based on the assumption that quantities not included on the demand (or tariff) curve are prohibitively expensive (i.e. are priced at Pcap). Using the convention that the first column of a curve is the quantity and the second column is the price:

$$curve_{placeholder}(1,1) = \max\left(curve_{demand}(1,1), curve_{tariff}(1,1)\right)$$

$$curve_{placeholder}(end, 1) = \min\left(curve_{demand}(1,1), curve_{tariff}(1,1)\right)$$

As a practical consideration, tariff curves can be infinitely large. For example, a target quantity tariff curve can be expected to extend horizontally to positive and negative infinitely. With this in mind, a tariff curve may not be generated until after the demand curve is known. Then the demand curve is only defined to be one Qbin longer than the demand curve, in either direction. The resulting adjusted curve is the same.

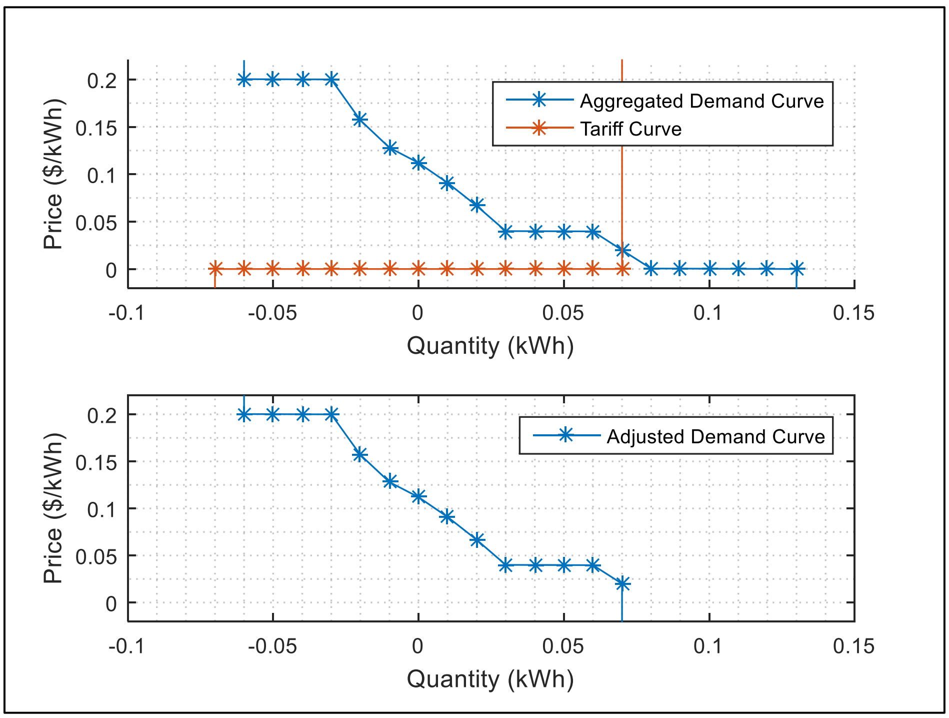


Figure 29: Example Trimming in Demand Curve Adjustment

Two, the adjusted curve is simply the difference between the demand curve and tariff curve, at each Qbin interval. For all values of *n* resulting from the "trimming" in step one:

$$curve_{placeholder}(nQbin, 2) = curve_{demand}(nQbin, 2) - curve_{tariff}(nQbin, 2)$$

When a node has multiple tariffs between it and its supernode, *obj.submitCurve( )* repeats this process for each tariff, iteratively adjusting the placeholder curve. The final result is stored in the parameter *node.curveSubmit*.

```
obj.curveSubmit = curvePlaceholder;
```

Note, during *submitCurve*, the simulation deviates from the DTDM rules in a few ways.

One, the node is requesting a tariff curve from the tariff object. However, in the DTDM construct, the tariff owner should only provide updated parameters; it is the node's responsibility to interpret the tariff instance into a tariff curve. This does not impact the simulation; all actors in the system would interpret the tariff instance in the same way. That being said, implementation or more advanced simulation would make this distinction.

Two, the simulation requires the node to request an update from the tariff owner. In practice, this update would be initiated by the tariff owner, in anticipation of Market Operation. This is a slight distinction; however, in implementation, the responsibility to update the tariff parameters should fall to the tariff owner, not the subnode.

Additionally, power factors are not tracked in the simulation. However, if they were, demand curve adjustment would not impact demand curve power factors. The process of adjustment only adjusts the price points on the demand curve, not energy quantities. Thus the power factor for each demand curve point would be unchanged.

The end result of *node.curveSubmit* and demand curve adjustment is an updated *node.curveSubmit* parameter. This is the node's contractual offer and is made available to the supernode.

The process of demand curve adjustment illustrated below, along with the forthcoming processes of translation and aggregation.

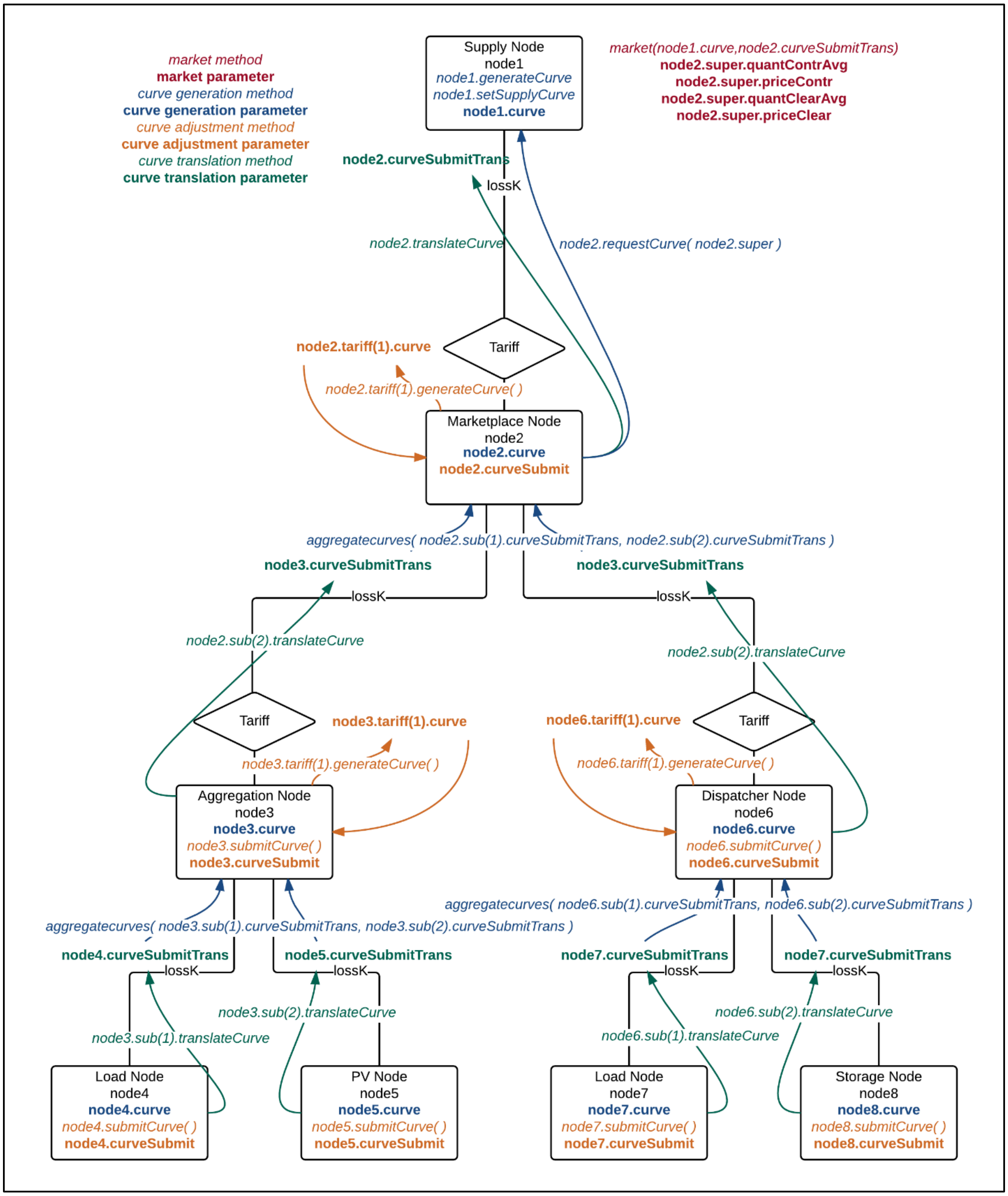


Figure 30: runMarket (2/3)

5.6.3 Demand Curve Translation

Once the supernode has received the demand curve submission, it must consider any linkage losses between itself and the subnode. This is demand curve translation. Demand curve translation will manipulate the subnode's demand curve submission, *node.curveSubmit*, into a demand curve that incorporates physical losses: *node.curveSubmitTrans*.

Note, the subnode's demand curve submission is its contractual offer. Thus, demand curve translation is the supernode's responsibility; they take on all risks in estimating physical losses. Loss translation assumes constant power and power factor over the course of the market period. These may or may not be a reasonable assumption. Additionally, loss translation requires making a reasonable estimation of the linkage loss parameters. If a practical implementation provides excessive variability for these considerations, the loss translation method may be impractical.

However, it is for this reason that the loss translation is accomplished of the supernode owner. It is expected that this system actor will have a better understanding of the parameters and approximations made during loss translation. As examples: a facility manager will have a better understanding of facility wiring than the buildings tenants; a distribution system operator will have a better understanding of the distribution linkages than individuals households. Not only do the supernode owners have a better knowledge of parameters, but they can be expected to be in a better position for hedging and risk mitigation. For example, if extreme weather results in an unknown impact on the physical loss parameters, the distribution system operator has the ability to adjust the loss translation calculation to minimize risk over the range of possibilities.

This last consideration, hedging, also applies to non-constant power and power factors over the market period. The supernode owner, when accomplishing loss translation, can choose parameters that reflect their desired level of risk. For example, if constant power is not as reasonable assumption, the owner can apply a contingency multiplier.

In this simulation, the translated demand curve is stored at as a parameter, *node.curveSubmitTrans,* for the subnode object. This is for data management convenience only. In practice, the translated demand curve would be stored by the supernode and may be unknown by the subnode.

Demand curve translation is controlled by the linkage loss parameter, *node.lossK*, which was defined in Section 5.2.3. As described in that section, the linkage loss parameter is a function of the subnode's power factor. Advanced implementation may include a specified power factor for every quantity point on the demand curve. However, for the current simulation, the power factor is considered constant. Therefore, *node.lossK* is considered constant for all points on the demand curve.

Loss translation captures the impact of physical energy losses by translating a subnode's expressed demand curve (in terms on energy received) into a curve to be used by the supernode (in terms of energy delivered). A demand curve is expressed as price points at a set interval of energy quantities; physical losses will impact both components.

Physical energy losses are a function of energy delivered. In general, this is described as

$$Q_{loss} = kQ_{delivered}^2$$

where k is an approximated linkage loss constant, *node.lossK*, as defined in Section 5.2.3. Notice, this definition relies on the quantity delivered to the subnode, as opposed to the quantity supplied by the supernode.

As a reminder, Q represents a quantity of electrical energy, while P represents a unit price for the electrical energy.

With the linkage loss constant established, a subnode's demand curve can be translated. Each point on the subnode's demand curve is an offer to transact energy. The subnode can be thought to say: "I will purchases Q units of energy at a price of P per unit of energy." This is also a commitment to purchase Q units of energy at a total cost of P x Q. The subnode is expressing demand entirely in terms of energy it receives.

First, it is assumed that the subnode demand is positive; energy will flow from the supernode to subnode. From the supernode's perspective, energy supplied must compensate for energy loss. Based on the demand convention, the supernode supplies energy

$$Q_{supplied} = Q_{delivered} + Q_{loss}$$

Using the previous Q_{loss} definition, this becomes

$$Q_{supplied} = Q_{delivered} + kQ_{delivered}^2$$

Using Q as energy delivered (the subnode quantity) and Q' as energy supplied (the supernode quantity), this becomes

$$Q' = Q + kQ^2$$

Further, the customer expressed their price in terms of energy delivered. Dividing their payment commitment (Q x P) by the energy supplied yields a new price point:

$$P' = \frac{QP}{Q'} = \frac{QP}{Q + kQ^2}$$

These two equations are then used to translate the subnode's demand curve from energy delivered to energy supplied. Modification of the demand curve will occur for each discrete point on the demand curve. The modification of each point can then be broken into two components: shift in Q and shift in P.

This shift is easily determined for quantity:

$$\Delta Q = Q' - Q = (Q + kQ^2) - Q = kQ^2$$

And with a few more steps for price:

$$\Delta P = P' - P = \left(\frac{QP}{Q + kQ^2}\right) - P = P\left(\frac{Q}{Q + kQ^2} - 1\right)$$

Expressing ΔP in terms of ΔQ provides

$$\Delta P = P\left(\frac{Q}{Q + \Delta Q} - 1\right), \qquad Q > 0$$

Notice, the change in Q relies entirely on the point's Q value. However, the change in P relies on both the point's P and Q values.

This equation is not valid when Q = 0, so an exception must be made for that case. Recognizing that there is no loss when Q = 0, the subnode's offer need not change. Therefore

$$\Delta Q = 0, \qquad \Delta P = 0, \qquad Q = 0$$

Next, consideration must be made for negative demand curve quantities; when energy flows from the subnode to the supernode. In these cases, the subnode is offering to export energy to the supernode. Physical losses are indifferent to the direction of flow and serve to reduce the exported energy received at the supernode. However, demand convention describes export energy as negative, so the losses are additive. The relationship between sub- and supernode energy quantity does not change:

$$Q' = Q + kQ^2, \qquad Q < 0$$

where Q still represents the subnode demand curve quantity and Q' represents the supernode's observed quantity. Notice, when Q is a negative quantity, the losses result in a Q' with a lesser magnitude than Q. Again, the translation in quantity is expressed as

$$\Delta Q = Q' - Q = (Q + kQ^2) - Q = kQ^2, \qquad Q < 0$$

Notice ΔQ will always be a fraction of the magnitude of Q. This prevents the loss tariff translation from shifting a quantity from the "exporting" to "importing" region.

Applying this equation to the price translation results in

$$P' = \frac{QP}{Q'} = \frac{QP}{Q + kQ^2}, \qquad Q < 0$$

$$\Delta P = P' - P = \left(\frac{QP}{Q + kQ^2}\right) - P = P\left(\frac{Q}{Q + kQ^2} - 1\right), \qquad Q < 0$$

$$\Delta P = P\left(\frac{Q}{Q + \Delta Q} - 1\right), \qquad Q < 0$$

Notice the price translation also does not change from the previous expression. Additionally, for negative Q values, the ΔQ expression always yields a positive value. However, negative Q values yield positive ΔP values. This makes sense: a supernode importing energy from its subnode must "pay" for the losses ignored in the subnode's offer.

In summary, the following expressions are used to translate a demand curve by a system loss component, *k*:

$$\Delta Q = kQ^2$$

$$\Delta P = P\left(\frac{Q}{Q + \Delta Q} - 1\right), \qquad Q \neq 0$$

$$\Delta P = 0, \qquad Q = 0$$

Finally, it is also possible for a demand curve to have points below the Q-axis (i.e. negative price values). In these instances, the above equations are still valid. The translation of points can be thus be generalized as follows:

	Negative Quantity	**Positive Quantity**
Positive Price	ΔP: Away from Q-axis (up) ΔQ: Toward P-axis (right)	ΔP: Toward Q-axis (down) ΔQ: Away from P-axis (right)
Negative Price	ΔP: Away from Q-axis (down) ΔQ: Toward P-axis (right)	ΔP: Toward Q-axis (up) ΔQ: Away from P-axis (right)

Table 6: Loss Translation Generalization

For practical values of k, when the initial curve is strictly decreasing, it is conjectured that the transformed curve will be also strictly decreasing in any quadrant. This has been examined empirically, but is not proven analytically.

When implementing the loss tariff transformation, there are practical limitations, based on the DTDM demand curve conventions.

First, it is possible for the transformed curve to violate Pmin limits for adjacent quantities. In particular, this would occur when a curve, already at the Pmin limit, crosses the P-axis at a negative value. The point on the P-axis (Q = 0) would not shift, but the next point would move up and to the right, potentially violating the Pmin constraint. This is a practical consideration that, while rare, must be monitored for when translating a subnode's demand curve. When this occurs, normal Pmin violation adjustments can be implemented.

A more significant difficultly arises when restricting the translated demand curve to the system defined Qbin interval. In particular, it is extremely unlikely that each point's ΔQ will equal an integer multiple of Qbin. However, each point on the translated curve must be at a Qbin interval. If the initial demand curve was generated by a known mathematical equation, the translated curve could be generated and sampled at the Qbin interval. However, the DTDM requires no such expectation, so a general approach is needed.

Using the expectation that the Qbin interval is small relative to the practical control of smart devices, the algorithm can force the ΔQ calculation to the nearest Qbin interval. This can be done with round, floor, or ceiling functions. Of these options, the round function provides the closest fit to the "true" curve transformation. Three examples follow. Notice the ceiling function is particularly inaccurate for values near Q = 0. The round function provides the best “averaged” error and is used for algorithm implementation.

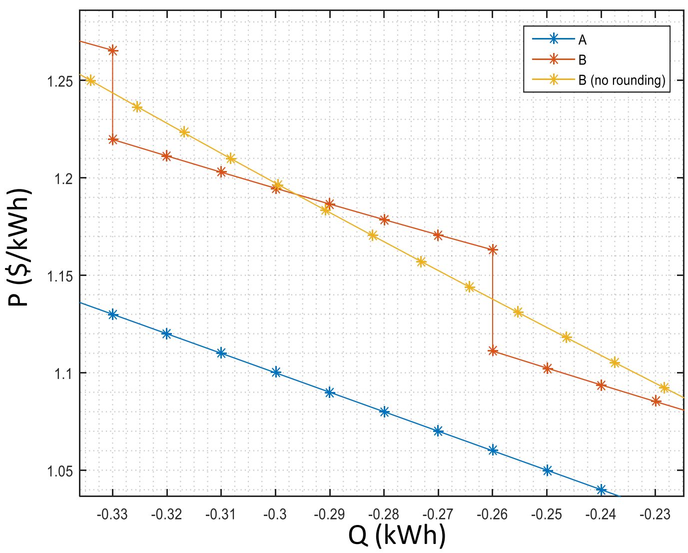


Figure 33: Translated Curve Segment (round

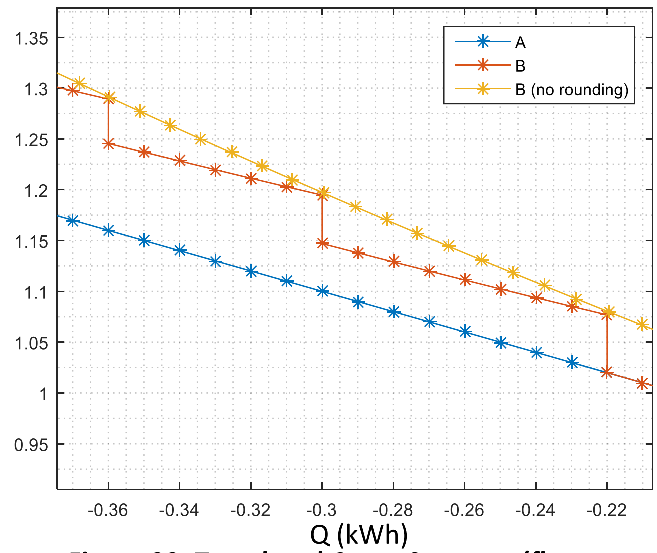


Figure 33: Translated Curve Segment (floor

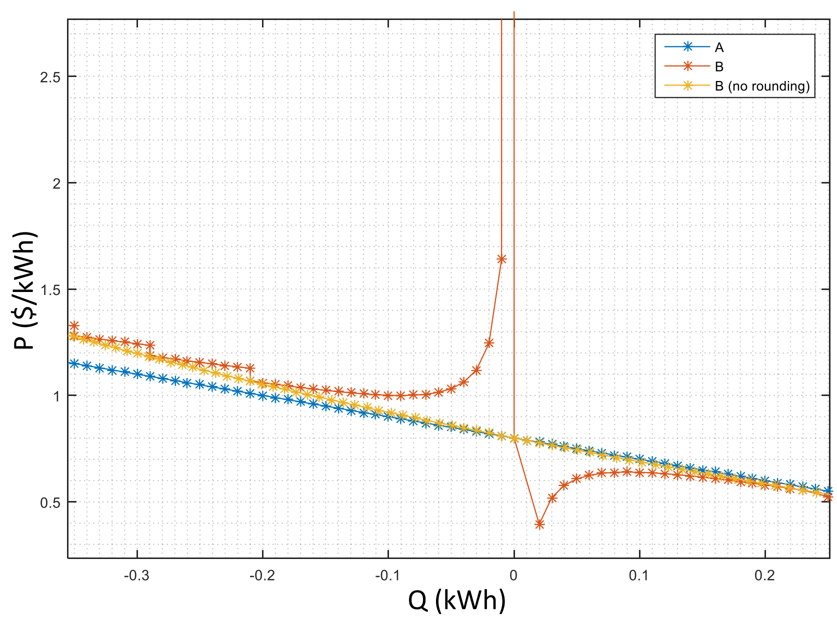


Figure 31: Translated Curve Segment (ceil function)

Rounding quantity values provides another challenge. The temptation is to use the rounded ΔQ when calculating ΔP. In particular, this would simplify loss tariff transformation in low-loss physical systems. When the approximated linkage loss constant, k, is very small, $\Delta Q = kQ^2$ may round to zero over the range of possible Q values. In this case, ΔQ would always be zero. Notice then, ΔP simplifies to

$$\Delta P = P\left(\frac{Q}{Q+0} - 1\right) = P(1-1) = 0, \qquad \Delta Q = 0$$

In these cases, the loss tariff will not provide any effect when translating the demand curve and the loss tariff transformation can then be neglected.

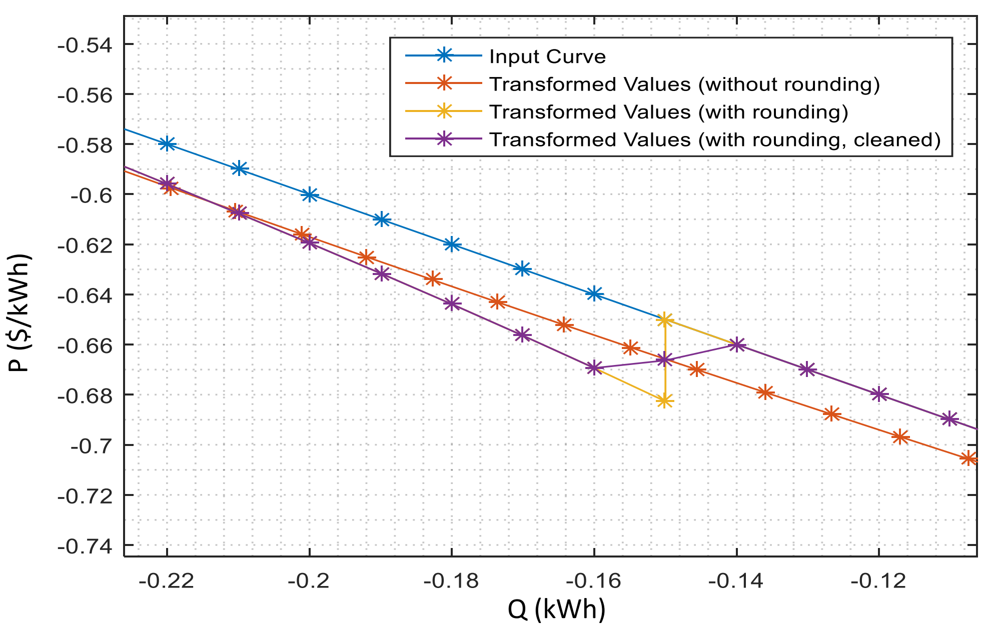


Figure 34: Translated Curve Segment (round Function) for Negative Prices

However, this method provides one significant disadvantage. Rounding ΔQ results in “jumps” when moving from one Qbin interval to another. For positive price values, the result remains strictly decreasing. For negative price values, the resulting curve may ascend where the jump occurs. An illustration follows. This ascending segments are solely due to rounding ΔQ; notice the unrounded curve is descending. This effect also occurs when "cleaning" the rounded values to prevent multiple points at the same Qbin value.

To avoid this consequence of rounding, the rounded ΔQ is used for a point's quantity value but the unrounded ΔQ value is used when determining ΔP for that point. This ensures the transformed curve remains strictly decreasing. It also results in a curve that more closely approximates the "true" transformed curve.

Two additional approximations must be made to "clean" the translation after rounding to Qbin intervals. One, for positive Q values, the ΔQ will be seen to "jump" when it rounds to the next largest Qbin interval. As a result, the translated demand curve will "skip" Qbins. These gaps are filled with linear interpolation. Two, for negative Q values, these "jumps" provide duplicate values for the same Qbin. These points must be replaced by a single point. As an approximation, the mean of the two price values is used. It is expected that, in practical applications, both Qbin and k will be small enough to prevent ΔQ rounding to result in more than two points at the same Qbin interval.

In conclusion, if the subnode demand curve is monotonic and decreasing, the transformed curve, with rounding as described, will also be monotonic and decreasing. Price values will appear at, and only at, Qbin intervals.

As a final illustration, a linear demand curve is translated in Figure 36. The initial demand curve includes Q values from -0.8 to 0.8 and P values from 1.6 to 0. Qbin is 0.01. No Pmin is specified, but the input curve has a P step size of 0.01 for every point. For illustration, k = 0.2 is used. Note, Qbin and k are both impractically large for the purposes of illustration. This illustration demonstrates the impact of using the unrounded ΔQ value when determining ΔP.

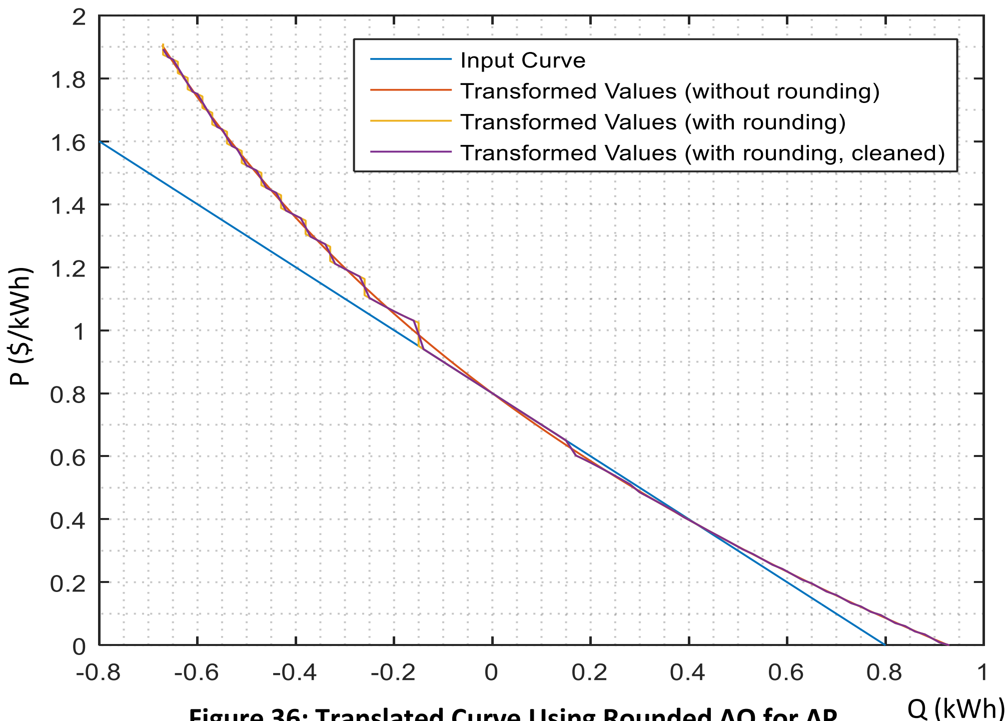


Figure 36: Translated Curve Using Rounded ΔQ for ΔP

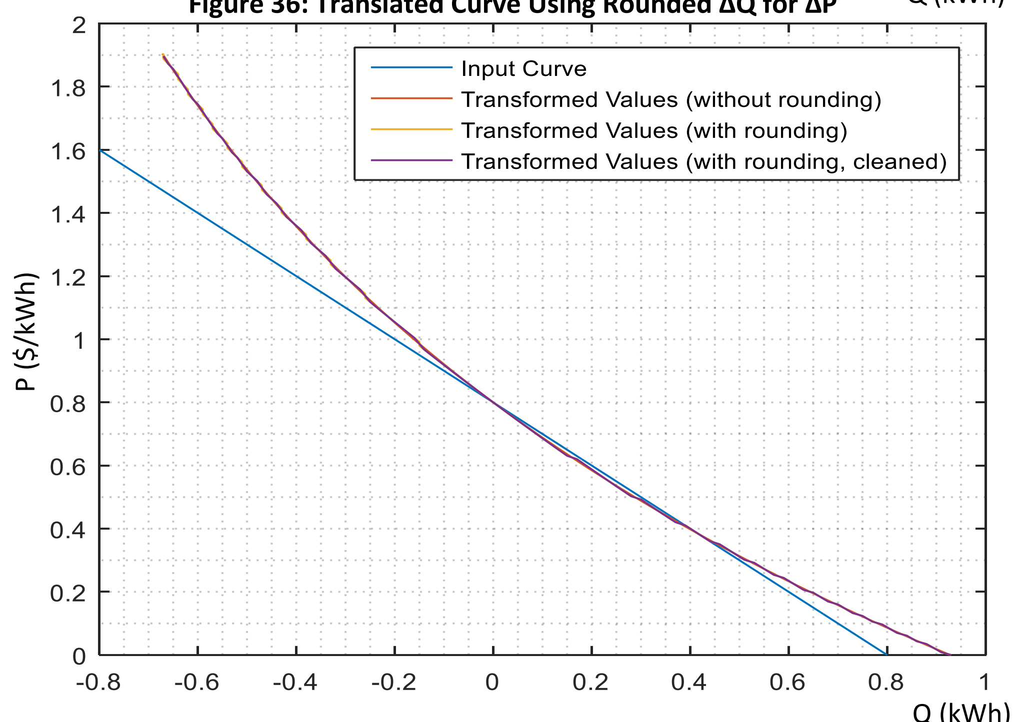


Figure 36: Translated Curve Using Unrounded ΔQ for ΔP

In summary, the method *node.translateCurve* serves to implement demand curve translation. This includes translating the quantity values, translating the price values, rounding the quantity values to the nearest Qbin, then removing duplicates and linearly interpolating missing values. The method is provided below.

```
function translateCurve(obj)
      if obj.lossK == 0
          obj.curveSubmitTrans = obj.curveSubmit;
      elseif isempty(obj.curveSubmit)
          obj.curveSubmitTrans = [];
      else
          Q = obj.curveSubmit(:,1);
          newQunrounded = (Q+obj.lossK*Q.^2);
          newQ = obj.Qbin*round(newQunrounded/obj.Qbin);
          P = obj.curveSubmit(:,2);
          I = (Q~=0);
          % index of nonzero Q values (zero Q value does not adjust
      P)
          newP = P;
          newP(I) = P(I)+P(I).*(Q(I)./newQunrounded(I)-1);
          if size(obj.curveSubmit,1) == 1
          % if initial curve is a single point, no interpolation
needed
              obj.curveSubmitTrans = [newQ,newP];
          else
          % else, curve is multiple points
          % first, check for translated points at the same Qbin
            (possible for Q < 0)
              if Q(1) < 0
                  dupI = find(newQ(1:end-1)==newQ(2:end));
                  for n = 1:size(dupI,1)
                      % for each identified duplicate Q entry in
newQ
                      index = dupI(n);
                      newP(index) = (newP(index)+newP(index+1))/2;
                        % average duplicate Qbin points
```

```
                    newQ(index+1) = []; % remove duplicate row
                    newP(index+1) = []; % remove duplicate row
                    dupI = dupI - 1;    % reduce index reference
                end
            end
            % second, interpolate for missing values
                (possible for Q > 0)
            if Q(end) > 0
                interpQ = (newQ(1):obj.Qbin:newQ(end))';
                interpP = interp1(newQ,newP,interpQ);
                newQ = interpQ;
                newP = interpP;
            end
            obj.curveSubmitTrans = [newQ,newP];
        end
    end
end
```

Finally, in this simulation power factor values are not tracked with the demand curve. However, if power factors are tracked with the demand curve, then translation would also impact the power factor values. To accomplish power factor translation, linkage components would need to be parameterized by both their active and reactive loss components. A more detailed analysis is not provided at this point.

The demand curve translation process is illustrated in Section 5.6.2.

5.6.4 Demand Curve Aggregation

The final process involved in the demand curve request is aggregation. Aggregation occurs when a supernode has available *node.sub(n).curveSubmitTrans* for all its subnodes. The node then calls the function *aggregatecurves* to develop its own demand curve. Thus, for a non-bottom-level node,

node.curve represents the adjusted, translated, and aggregated preferences of the subnetwork "below" the non-bottom-level node. This is the final step in the Marketplace node's *generateCurve* method.

Aggregation represent horizontal addition of demand curves. In effect, this uses price as the independent variable and quantity as the dependent variable. However, quantities are expressed at a fixed interval (Qbin), while prices are not. Additionally, the resulting curve must be expressed at the same fixed quantity interval. These restrictions pose challenges in implementation.

In the example below, curves A and B must be aggregated. The LH and RH vertical limits are not included. The different methods are illustrated in Figure 37.

A		**B**	
Q (kWh)	**P ($/kWh)**	**Q (kWh)**	**P ($/kWh)**
0	3	0	2
1	2.5	1	1.5
2	1.5	2	1
3	0.8	3	0.5
		4	0

Table 7: Example Curve Aggregation Methods Data

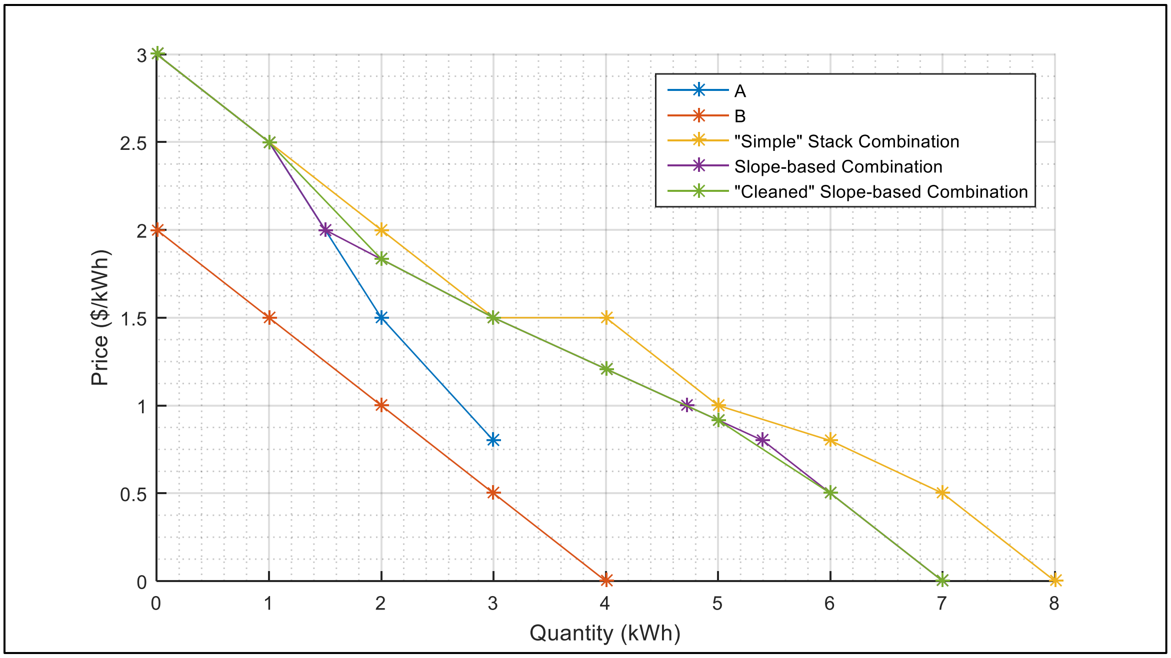


Figure 37: Example Curve Aggregation Methods

One approach to aggregating the curves is to simply sort the price points of A and B in descending order, with quantities based on the Qbin interval. This is illustrated as the "simple" stack combination. However, inspection reveals some problems with this approach. First, the aggregate curve implies a demand of Q = 8 when P = 0. However, the maximum demand of A and B are shown to be 3 and 4, respectively. This is an error equal to one Qbin. This cumulative error would be particularly concerning when aggregating many curves. Second, the resulting curve includes a horizontal segment at P = 1.5. This is because both A and B include a point that price in their demand curve. This is unacceptable; like the component curves, the desired aggregation must be monotonic and strictly decreasing. Third, visual inspection reveals that this curve does not actually represent the horizontal combination of A and B; it does not accurately predict the aggregate demand at most price points. This approach is unacceptable.

Instead, the curves should be aggregated with a slope-based combination. This is derived by combining the slopes of each demand curve in a piecewise fashion. It specifically takes advantage of the monotonic and decreasing nature of each demand curve. If the linear interpolation between points is to be trusted, this is the most accurate representation of the aggregate demand.

Specifically, slope-based combination is accomplished by examining the slope of each curve from the highest contributing price to the lowest contributing price. Each curve is considered to have a negatively infinite slope for prices above their maximum price and below their minimum price. The following equation is used to determine the combined slope of curves A and B:

$$combinedSlope = \frac{1}{\frac{1}{slopeA} + \frac{1}{slopeB}}$$

First, the aggregated curve is anchored at the sum of contributing curves' initial quantities, at the higher of the two initial prices:

$$[Q_1, P_1] = [\mathrm{Q}_1^{\mathrm{A}} + \mathrm{Q}_1^{\mathrm{B}}, \max(P_1^A, P_2^B)]$$

Next, *combinedSlope* is adjusted based on which curve provided the anchor price. Updating a slope from a given demand curve entry *n* is:

$$slope = \frac{P_{n+1} - P_n}{Qbin}$$

The combined slope is used to establish a candidate price. If either curve includes a price higher than this candidate price, an inflection point is created and that curve's slope is updated.

Otherwise, the combined slope establishes points at the Qbin interval. This process continues until both curves have exhausted all their points.

The result of slope-based combination is illustrated in Figure 37. Notice, points exist at each Qbin and at every price included in either demand curve. Additionally, this curve aggregation ends at the correct maximum load (Q = 7 in this example).

However, the aggregated curve cannot include points outside the Qbin interval. The most straightforward approach is to simply remove points that do not meet this criterion. This is illustrated as the “cleaned” slope-base combination. Notice this process introduces misrepresentations during linear interpolation. However, these errors are less than Qbin in all instances and are corrected by the next Qbin interval. This is considered an acceptable compromise. The “cleaning” of the aggregated curve must occur after the slope-based combination.

One problem with this method occurs when aggregating demand curves without overlapping prices. Examples are provided in Figures 38-40. Inspection demonstrates that, in the aggregated curve, a vertical line should be expected at prices below the Storage charging threshold ($0.04/kWh) and above the PV marginal cost ($0.00/kWh). However, this vertical line would make the demand curve non-monotonic, which is unacceptable. To correct for this, aggregation uses the higher price value at the duplicated Qbin interval. The result is shown below. This error is always equal to Qbin and occurs less often than the “simple” stack combination, so this method is still preferable. The impact of this error can be mitigated by used a small Qbin size. Additionally, this error is not cumulative. Once the error is introduced, the aggregate curve no longer has a vertical segment at that price range. Additional errors will not occur at that price range.

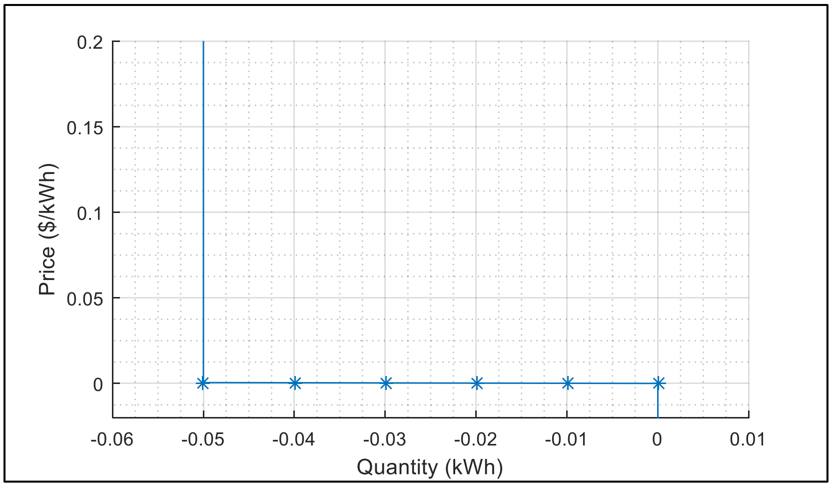


Figure 38: Example PV Demand Curve

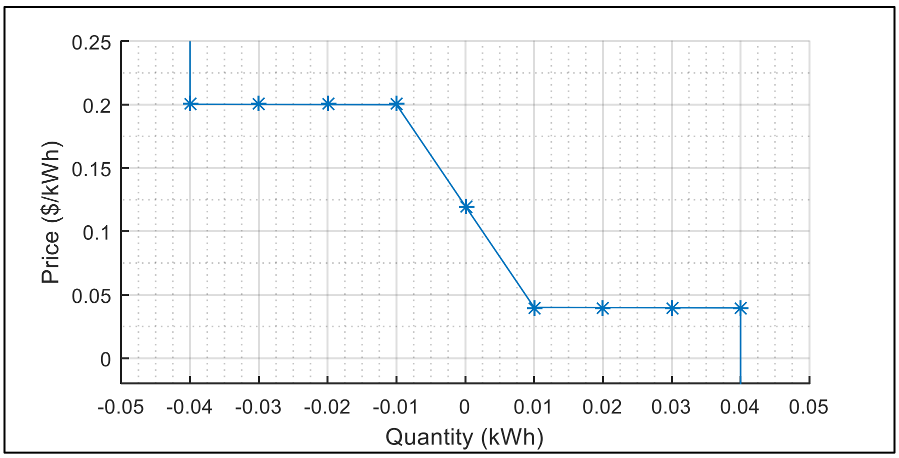


Figure 39: Example Storage Demand Curve

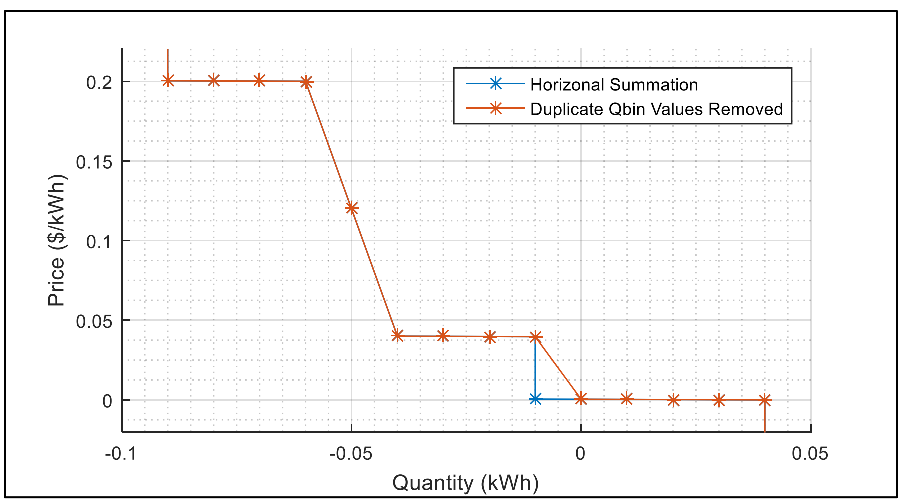


Figure 40: Example PV and Storage Curve Aggregation

In the simulation code, the aggregation process itself allows only two inputs. Aggregation of more than two curves will be performed sequentially, with intermediary aggregation curves. Alternatively, if the node has only one subnode, *node.curve* is simply *node.sub(1).curveSubmitTrans*, or the translated demand curve of the one subnode.

```
for n = 1:size(obj.sub,2)
      if n == 1
            obj.curve = obj.sub(n).curveSubmitTrans;
      else
            obj.curve = aggregatecurves(...
                  obj.curve,obj.sub(n).curveSubmitTrans,...
                  obj.sub(n).Qbin,obj.sub(n).Pmin);
      end
end
```

In this simulation power factor values are not tracked with the demand curve. However, if power factors are tracked with the demand curve, then aggregation would also impact the power factor values. To accomplish power factor aggregation, a process equivalent to phasor summation would occur for every quantity and power factor point. A more detailed analysis is not provided at this point.

The demand curve aggregation process is illustrated in Section 5.6.2.

Now that *aggregatecurves* has been described, every process has been outlined. As a result, the Marketplace Node has completed its *generateCurve* method. The Marketplace Node now has *node.curve* and can proceed with the market process.

5.6.5 Market Clearing Point

With its *node.curve* parameter updated through the *generateCurve* method, the Marketplace Node can proceed with *node.runMarket*. The next step is to determine the Marketplace Node clearing price and quantity. The market is said to clear at the price at which the supply quantity equals the demand quantity.

5.6.5.1 Clearing Point: Islanding Mode

In islanding mode, the Marketplace Node will be the top-level node. In this case, there is no supply curve. However, negative values on the demand curve can be considered supply. Thus, supply equals demand where the demand curve crosses the vertical axis: when Q = 0. Because the demand curve includes prices at every multiple of Qbin, there must be price at this point. This is the clearing price. If the demand curve does not cross the vertical axis, there is no clearing point.

```
if isempty(obj.super)
      index = find(obj.curve(:,1)==0);
      if isempty(index)
            error('Market cannot clear!');
      else
            obj.quantContrAvg = 0;
            obj.priceContr = obj.curve(index,2);
            obj.quantClearAvg = obj.quantContrAvg;
            obj.priceClear = obj.priceContr;
      end
```

There are four parameters updated after determining the clearing point: clearing price, average clearing quantity, contract price, and average contract quantity. The same definitions apply for all nodes, not just Marketplace Nodes. The clearing price, *obj.priceClear*, is the local price at which the market (or node) clears; this is a point on *node.curve*. The average clearing quantity, *obj.quantClearAvg*, is the corresponding quantity from *obj.curve*, divided by the market duration, *obj.marketDuration*.

However, the contract price is the price communicated to a node from its supernode: this is the Node Marginal Price. The contract price, *obj.priceContr* or NMP, is located on *obj.curveSubmit*. It is the price the supernode expects to be paid for energy delivered. The average contract quantity, *obj.quantContrAvg*, is the corresponding quantity from *obj.curveSubmit*, divided by the market duration, *obj.marketDuration*. Immediately following Market Operation, the *obj.quantClearAvg* will equal *obj.quantContrAvg*; however, they may differ during Real-Time Action for Dispatcher Nodes.

Note, the DTDM rules specify that only the NMP, *obj.priceContr*, is communicated after Market Operation. The other parameters are only used as intermediate steps and recorded for analysis. Specifically, *obj.quantContrAvg* is not contractually binding, other than in cases with a Prediction style tariff.

Finally, any Aggregation Node could potentially operate as a Marketplace Node in this manner. For example, a subsystem could continue to operate using DER when connection to the larger electrical grid is lost. This includes cases in which an Aggregation Node loses connection to its supernode in the DTDM network. The Aggregation Node can use the existing input from its subnodes and can make the market determination; it takes over as a Marketplace Node. Note, this transactive energy structure does not ensure the system will operate in this fashion, but merely provides an opportunity to do so.

The above clearing process applied only for top-level Marketplace Nodes. However, the Marketplace Node typically has a supernode: a Supply Node connecting to a larger energy network.

5.6.5.2 Supply Nodes

A Supply Node cannot have a supernode; it must be a top-level node. Like bottom-level nodes that are expected to consume energy (or generate negative energy), a Supply Node is expected to provide energy (or consume negative energy). Where a bottom-level node is a sink, the Supply Node is a source.

A Supply Node generates a supply curve, with similar conventions as a demand curve. Like a demand curve, the supply curve should have a series of quantity-price points, with a price for every Qbin interval. However, a supply curve is non-descending. Additionally, to support common financial arrangements, a supply curve is not required to be strictly ascending. For example, consider a connection to the wholesale market, with a flat rate energy price. The supply curve would be represented by a horizontal line at this energy price.

Tariffs may also be imposed between a Marketplace Node and its supernode Supply Node.

In all simulation cases, the DTDM is considered a "price taker" of the larger energy market. A price taker is an entity that has it has no influence on the market price being set, generally due to low market share. Thus, the DTDM network will have no influence on the price offered by the larger system, e.g. the wholesale energy market. If the DTDM and DDS concept is widely implemented, this assumption would need to be challenged.

Before a Marketplace Node can determine the clearing price and quantity, it must request a supply curve from its supernode. However, a Marketplace Node truly cares about the clearing point at its Supply supernode; there may be linkage losses and tariffs between the two. For example, if the Supply Node represents the connection to the wholesale energy market, the DTDM system operator seeks to clear the market at the price and quantity dictated by the wholesale energy market.

To this end, the Marketplace Node must adjust and translate its own demand curve. Note, because the Marketplace Node is connected to a supernode, demand curve adjustment was accomplished after aggregation during *obj.generateCurve*: *obj.curveSubmit* has already been established. Next, translation is accomplished by calling *obj.translateCurve*, which generates *obj.curveSubmitTrans*.

The Marketplace Node then requests a supply curve from its Supply supernode. This is accomplished, as before, with the method *obj.requestCurve*. Upon receiving this request, the Supply Node calls *obj.generateCurve*, which calls *obj.setSupplyCurve*.

When generating a supply curve in *obj.setSupplyCurve*, the simulation assumes an infinite capacity of node supply. It is expected that any capacity limits would be expressed as a capacity tariff between the Marketplace and Supply Nodes. Additionally, supply curves are assumed to be flat-rate price curve. This is not a limitation in the DTDM rules; it is an artificial limitation imposed by the current simulation code.

Thus, a supply curve is an infinitely horizontal function, at the flat-rate supply price. The flat-rate price is determined by referencing a dataset provided during initialization.

```
obj.supplyPrice          =          mean(obj.supplyDataSource(...
     refStart:refStart+(marketDuration-1)));
```

To limit the size of the supply curve, the Supply Node checks the length of the subnode's translated demand curve submission. As long as the supply curve exceeds the limits of the translated demand curve, the market clearing point can be determined correctly.

```
curveMin = min(curveMin,obj.sub.curveSubmitTrans(1,1));
curveMax = max(curveMax,obj.sub.curveSubmitTrans(end,1));
curveMin = curveMin - obj.Qbin;
curveMax = curveMax + obj.Qbin;
obj.curve(:,1) = [curveMin:obj.Qbin:curveMax];
obj.curve(:,2) = obj.supplyPrice;
```

This establishes the supply curve, *obj.curve*, which will be used when determining the Marketplace Node's clearing point. This step is included in the illustration provided in Section 5.6.2.

5.6.5.3 Clearing Point: Supply and Demand

The description of Supply Nodes above indicates that supply curves will always be infinitely horizontal. However, the DTDM rules and underlying simulation code do not make this assumption when establishing a clearing point from supply and demand curves. Thus, the general approach is described.

The following process is implemented in the function *market*, which requires an input supply and demand curve. The output of the function is the clearing quantity and price.

```
[clearQ,clearP] = market(obj.super.curve,obj.curveSubmitTrans);
```

To determine the clearing price and quantity from a supply and demand curve, there is a "standard" case and eight "fringe" cases. Each of the fringe cases occur when one of the curves intersects the other at its limit. In all cases, the demand curve is a series of descending prices at each Qbin interval and the supply curve is a series of non-descending (i.e. may be horizontal) prices at each Qbin interval.

For all cases, the curves are first aligned by their Qbin values. The description of curves does not include Pcap or any points beyond the first and last expressed quantity-price point. However, the resulting conclusion relies on the interpretation of demand and supply curve limits. Specifically, the left-hand (LH) limit of a demand curve begins a positive nearly-vertical line and the right-hand (RH) limit begins a negative nearly-vertical line. Conversely, the LH limit of a supply curve begins a negative nearly- vertical line and the RH limit begins a positive nearly- vertical line.

In the standard process, the demand curve begins above the supply curve, and the curves intersect. Specifically, the demand curve LH price is above the corresponding supply curve price and the demand curve falls below the supply curve before reaching either RH limit. The clearing price and quantity is the location where the two lines intersect. This location is determined with linear interpolation. Q_1 is the quantity at which demand price D_1 is above supply price S_1. The next Qbin interval is Q_2 at which demand price D_2 is below supply price S_2.

The following equations determine the clearing quantity and price when demand begins above supply and the curves intersect:

$$Q_{clear} = \frac{(Q_1 S_2 - S_1 Q_2)(Q_1 - Q_2) - (Q_1 - Q_2)(Q_1 D_2 - D_1 Q_2)}{(Q_1 - Q_2)(D_1 - D_2) - (S_1 - S_2)(Q_1 - Q_2)}$$

$$P_{clear} = \frac{(Q_1 S_2 - S_1 Q_2)(D_1 - D_2) - (S_1 - S_2)(Q_1 D_2 - D_1 Q_2)}{(Q_1 - Q_2)(D_1 - D_2) - (S_1 - S_2)(Q_1 - Q_2)}$$

In the first set of fringe cases, the demand curve begins above the supply curve, but the curves do not intersect. In these cases, the curve RH limits determine the clearing point.

In fringe case 1a, the supply curve ends before the demand curve. As a supply curve, the RH limit begins a positive nearly-vertical line. This nearly-vertical line intersects with the demand curve; this is the clearing point. Thus, the clearing quantity is the RH limit of the supply curve and the clearing price is the demand curve price at this quantity.

The following equations determine the clearing quantity and price when demand begins above supply, the curves do not intersect, and the supply curve ends before the demand curve:

$$Q_{clear} = Q_{supply}^{limitRH} \qquad P_{clear} = P_{demand}^{Q_{clear}}$$

In fringe case 1b, the same applies, but it is the demand curve that ends first. Thus, the clearing quantity is the RH limit of the demand curve and the clearing price is the supply curve price at this quantity.

The following equations determine the clearing quantity and price when demand begins above supply, the curves do not intersect, and the demand curve ends before the supply curve:

$$Q_{clear} = Q_{demand}^{limitRH} \qquad P_{clear} = P_{supply}^{Q_{clear}}$$

In fringe case 1c, both curves end at the same quantity. In this case, it is particularly valuable to consider each RH limit beginning a nearly-vertical line. If each line has the same slope, they will intersect at the price halfway between the supply and demand curve final price points. The clearing quantity is both curves' RH limit.

The following equations determine the clearing quantity and price when demand begins above supply, the curves do not intersect, and the supply and demand curve end at the same quantity:

$$Q_{clear} = Q_{supply}^{limitRH} = Q_{demand}^{limitRH} \qquad P_{clear} = \frac{P_{supply}^{Q_{clear}} + P_{demand}^{Q_{clear}}}{2}$$

In the second set of fringe cases, the demand curve begins below the supply curve and the curves do not intersect. In these cases, the curve LH limits determine the clearing point.

In fringe case 2a, the supply curve begins before the demand curve. As a result, when the demand curve begins with a positive nearly-vertical line, it intersects with the pre-existing supply curve. This is the clearing point. Thus, the clearing quantity is the LH limit of the demand curve and the clearing price is the supply curve price at this quantity.

The following equations determine the clearing quantity and price when demand begins below supply, the curves do not intersect, and the supply curve begins before the demand curve:

$$Q_{clear} = Q_{demand}^{limitLH} \qquad P_{clear} = P_{supply}^{Q_{clear}}$$

In fringe case 2b, the same applies, but it is the demand curve that begins first. Thus, the clearing quantity is the LH limit of the supply curve and the clearing price is the demand curve price at this quantity.

The following equations determine the clearing quantity and price when demand begins below supply, the curves do not intersect, and the demand curve begins before the supply curve:

$$Q_{clear} = Q_{supply}^{limitLH} \qquad P_{clear} = P_{demand}^{Q_{clear}}$$

In fringe case 2c, both curves begin at the same quantity. This result is similar to fringe case 1c. The clearing price halfway between the supply and demand curve starting price points. The clearing quantity is both curves' LH limit.

The following equations determine the clearing quantity and price when demand begins below supply, the curves do not intersect, and the supply and demand curve begin at the same quantity:

$$Q_{clear} = Q_{supply}^{limitLH} = Q_{demand}^{limitLH} \qquad P_{clear} = \frac{P_{supply}^{Q_{clear}} + P_{demand}^{Q_{clear}}}{2}$$

For the final case, the curves do not share any quantity values. These are non-overlapping curves. The Marketplace Node cannot provide a clearing price and quantity; the market does not clear.

Note, if both the supply and demand curves are a single point, at the same quantity, then both fringe case 1c or fringe case 2c apply with the same result. If the single-point-curves do not share the same quantity, then the non-overlapping curve case applies.

Illustrations of these cases are shown below. Note, the vertical lines shown at the LH and RH limits are not included in the supply and demand curve submissions; they are provided for illustrative purposes.

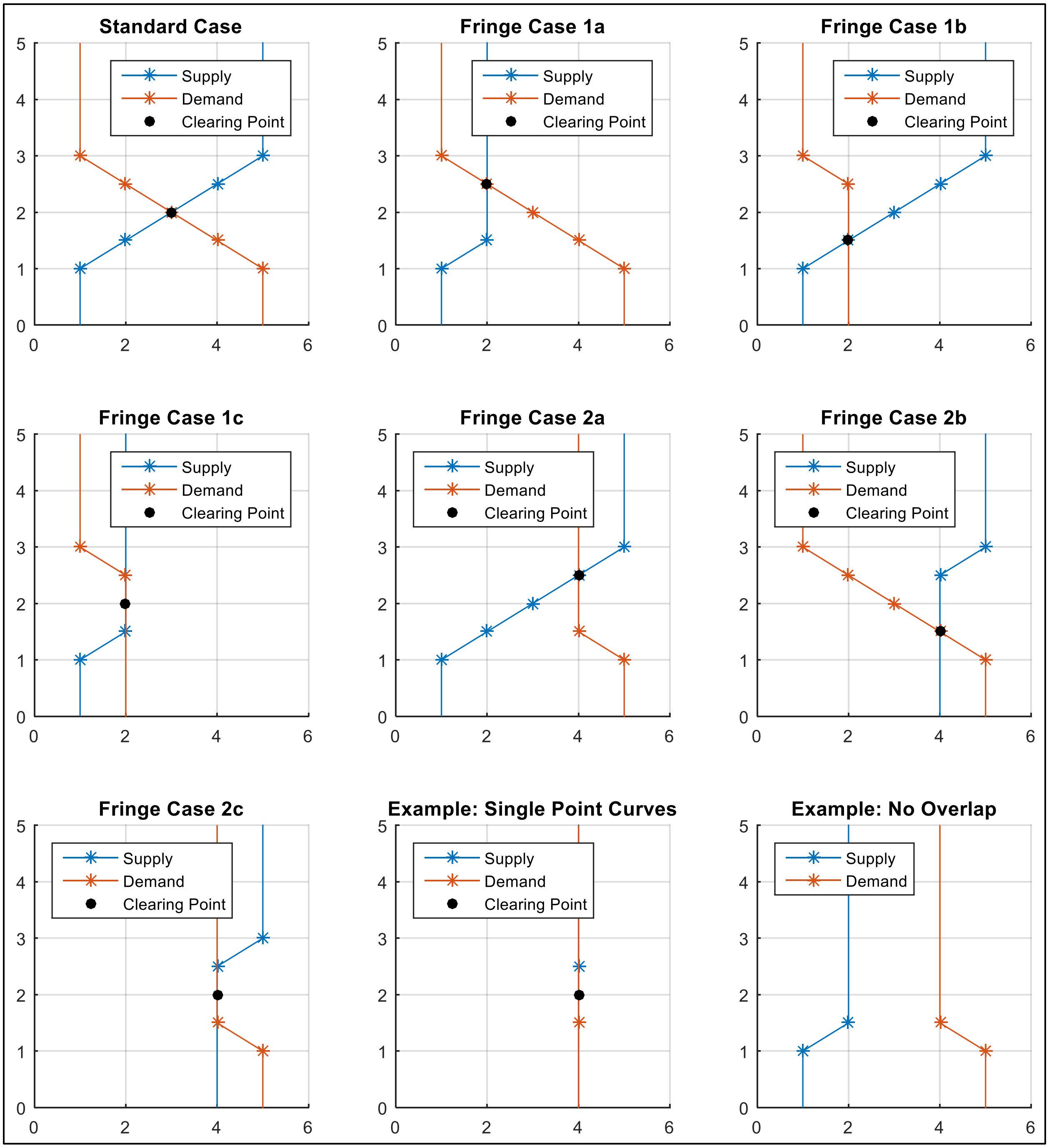


Figure 41: Market Clearing Point Cases (Q horizontal axis, P vertical axis)

The process described is accomplished in the function *market* and results in values for *clearQ* and *clearP*. The Marketplace Node calls this function on behalf of the perspective of the Supply Node. Thus, these values are assigned to the Supply Node. From the Marketplace Node object:

```
[clearQ,clearP] =
market(obj.super.curve,obj.curveSubmitTrans);
...
obj.super.quantContrAvg = clearQ/obj.marketDuration;
obj.super.priceContr = clearP;
obj.super.quantClearAvg = obj.super.quantContrAvg;
obj.super.priceClear = obj.super.priceContr;
```

For the Supply Node, there is no difference between the clearing and contract price and quantities; it bears no tariffs and its supply curve is unadjusted. This will not be the case for all other nodes in the system.

This is the final process illustrated in Section 5.6.2.

5.6.6 Node Marginal Pricing

After determining the local clearing price and quantity, the Marketplace Node must communicate a NMP to every node in the system. This is accomplished through the method *obj.sendContr*.

When a price is communicated to a subnode, it is not a clearing price, but a contract price. These subnodes will use the contract price to determine their local clearing quantity and clearing price. This is accomplished through the method *obj.setClear.* These clearing prices are then communicated to their subnodes as contract prices, and the process continues down to each end node. This process requires adjustment similar to demand curve translation.

Concisely: a clearing price is used by a node in its local calculations, while a contract price is sent between nodes.

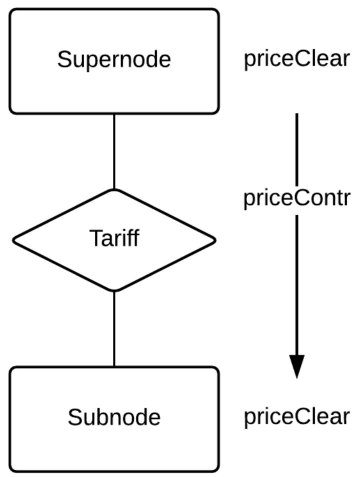


Figure 42: Clearing Price vs Contract Price

5.6.6.1 Clearing Price to Contract Price

Determining a contract price from a clearing price is accomplished through the method *obj.sendContr*.

To determine the contract price, NMP, a node must first locate the clearing price on each of its subnodes' translated demand curve submission. This provides a *quantSuperDesired* for each subnode, which is the desired contribution of each to the clearing quantity.

```
quantSuperDesired =
      price2quant(targetNodeObj.curveSubmitTrans,obj.priceClear);
```

The function *price2quant* finds the quantity corresponding to specific price on the input demand curve. This is possible because the translated demand curve submission is monotonic and strictly decreasing. There are three possibilities. One, the price exceeds the LH limit price; the corresponding quantity is the LH quantity. Two, the price is exceeded by all price values, including the RH limit; the corresponding quantity is the RH quanity. Three, the price lies between the LH and RH prices; the corresponding quantity is determined using linear interpolation of demand curve points.

```
row = find(curve(:,2)<price,1); % row of first P below price
if row == 1
% if price exceeds lefthand P, use lefthand quantity
      quant = curve(1,1);
elseif isempty(row)
% if price is less than every quantity, use righthand quantity
      quant = curve(end,1);
else
% else interpolate to find quant
      Q1 = curve(row-1,1);     P1 = curve(row-1,2);
      Q2 = curve(row,1);       P2 = curve(row,2);
      quant = Q1+((Q2-Q1)/(P2-P1))*(price-P1);
end
```

The value of *quantSuperDesired* is in terms of energy leaving the node for the supernode; it uses the translated demand curve submission. However, each subnode submitted an untranslated demand curve. The contract price communicated to each subnode should be based on the untranslated demand curve.

Thus, *quantSuperDesired* must be translated into a quantity received by the subnode: this provides *quantDesired* for each subnode. This process is the complement of demand curve translation for a single quantity.

If there is no linkage loss, i.e. *lossK = 0*, then the quantity leaving the node is equal to the quantity received by the subnode. E.g. *quantDesired = quantSuperDesired*.

For non-zero linkage loss, consider the *quantSuperDesired* as $Q'_{desired}$, where the superscript indicates measurement at the supernode itself. Similarly, consider the translated quantity *quantDesired* as $Q_{desired}$, where the lack of a superscript indicates measurement at the subnode. With a known Q' and k, the loss relationship equation can be used to solve for Q:

$$Q' = Q + kQ^2$$

$$kQ^2 + Q - Q' = 0$$

$$Q = \frac{-1 \pm \sqrt{1 + 4kQ'}}{2k}$$

For subnodes importing energy, $Q'_{desired} > 0$. Therefore $\sqrt{1 + 4kQ'_{desired}} > 1$. To ensure $Q_{desired} > 0$, the numerator is selected to be $-1 + \sqrt{1 + 4kQ'_{desired}}$.

Alternatively, for subnodes exporting energy, $Q'_{desired} < 0$. Therefore, $0 < \sqrt{1 + 4kQ'_{desired}} < 1$ for all real values. Note, practical boundaries on the value of *k* will always ensure $|4kQ'_{desired}| < 1$, providing real values. Based on this limitation, $Q_{desired} < 0$ for both possible numerator choices. However, the most realistic solution is that in which $Q_{desired}$ is closest to $Q'_{desired}$. This occurs when $-1 + \sqrt{1 + 4kQ'_{desired}}$ is the numerator.

In summary, when translating a quantity,

$$Q_{desired} = \frac{-1 + \sqrt{1 + 4kQ'_{desired}}}{2k}, \qquad k \neq 0$$

$$Q_{desired} = Q'_{desired}, \qquad k = 0$$

Or, in Matlab:

```
k = targetNodeObj.lossK;
if k==0
      quantDesired = quantSuperDesired
else
      quantDesired = (-1+sqrt(1+4*k*quantSuperDesired))/(2*k);
end
```

As will be seen, this translation is simply an intermediate step in determining a price to communicate to a subnode. Thus, rounding is not required for this step.

Finally, the node then uses the function *quant2price* to determine the location of each subnode *quantDesired* on their demand curve submission.

```
targetNodeObj.priceContr =
      quant2price(targetNodeObj.curveSubmit,quantDesired);
```

Similar to *price2quant*, the function *quant2price* finds the price corresponding to specific quantity on the input demand curve. This is possible because the demand curve submission is monotonic and strictly decreasing. The price is always determined using linear interpolation between a pair a price quanity-points. There are three possibilities for the pair of points, each of which his described in the Matlab code below.

```
row = find(curve(:,1)>quant,1); % row of first Q above quant
if row == 1
% if quant less than lefthand quantity, use Pcap
      Q1 = curve(1,1)-Qbin;    P1 = Pcap;
      Q2 = curve(1,1);         P2 = curve(1,2);
elseif isempty(row)
% if quant greater than every quantity, use -Pcap
      Q1 = curve(end,1);       P1 = curve(end,2);
      Q2 = curve(end,1)+Qbin; P2 = -Pcap;
else
% else interpolate to find quant
      Q1 = curve(row-1,1);     P1 = curve(row-1,2);
      Q2 = curve(row,1);       P2 = curve(row,2);
end
price = P1+((P2-P1)/(Q2-Q1))*(quant-Q1);
```

Note, before calling the function *quant2price*, the method *sendContr* verifies that the desired quantity either lies within the demand curve limits or within close proximity of either limit. This allows linear interpolation using ±Pcap and the LH/RH limit points.

At the conclusion of this process, each subnode has received a NMP, stored in the parameter *node.priceContr*.

5.6.6.2 Contract Price to Clearing Price

Determining a clearing price from a contract price is accomplished through the method *obj.setClear*.

To continue propagating NMPs throughout the system, each node must determine its local clearing point. Recall, the contract price, NMP, is based on the node's demand curve submission. The clearing price, however, is based on the node's actual demand curve.

First, the node must convert the received contract price to an equivalent contracted quantity, based on their demand curve submission. While this can be considered a contracted quantity, there may or may not be penalties for deviating from this quantity. Such penalties would be assessed by a Prediction Tariff.

```
obj.quantContrAvg = ...
    price2quant(obj.curveSubmit,obj.priceContr)/marketDuration;
```

Note, for convenience, this value is stored as an average quantity for each timestep in the market duration.

During Market Operation, the equivalent contracted quantity is the same as the clearing quantity. Note, this may not be the case during Real-Time Actions, especially for Dispatcher Nodes.

```
obj.quantClearAvg = obj.quantContrAvg;
quantDesired = obj.quantContrAvg*marketDuration;
```

These steps appear to be circular: *quantDesired* is simply *price2quant(obj.curveSubmit,obj.priceContr)*. However, by being deliberate in the transitional steps, there is opportunity for the node to perform nuanced risk mitigation. Specifically, a node may recognize the uncertainty inherent in the demand curve submission of its subnodes. That, coupled with the limitations of precision, may encourage a node to shift the *quantDesired* value. This is implemented in the simulation, in a limited fashion.

If the demand curve submission is "trimmed" by a tariff (e.g. a Capacity Tariff), then the node may risk large financial penalties for exceeding its demand curve submission, even by a very small quantity. To avoid this penalty to occur simply due to floating point error, the node adjusts its *quantDesired* slightly away from the tariff limit.

```
if (quantDesired-obj.Qbin)<obj.curveSubmit(1,1)
% if desired quantity is within a Qbin of curveSubmit LH limit
      if (obj.curveSubmit(1,1)-obj.Qbin)>obj.curve(1,1)
      % and if curveSubmit is more restrictive than curve
      % quantDesired should be increased slightly
            quantDesired = quantDesired + obj.Qbin;
      end
elseif (quantDesired+obj.Qbin)>obj.curveSubmit(end,1)
% if desired quantity is within a Qbin of curveSubmit RH limit
      if (obj.curveSubmit(end,1)+obj.Qbin)<obj.curve(end,1)
      % and if curveSubmit is more restrictive than curve
      % quantDesired should be decreased slightly
            quantDesired = quantDesired - obj.Qbin;
      end
end
obj.quantClearAvg = quantDesired/marketDuration;
```

More advanced risk mitigation steps may occur during this process, but are not included in the simulation.

Finally, with *quantDesired* updated, providing the local clearing quantity, the node can determine its local clearing price, *priceClear*. This is simply accomplished by finding the location of *quantDesired* on *obj.curve*.

```
obj.priceClear                                 =                                 ...
      quant2price(obj.curve,quantDesired,obj.Qbin,obj.DDS.Pcap);
obj.sendContr(marketDuration);
```

After determining the local clearing price, the node calls the method *obj.sendContr*. This repeats the process described in Section 5.5.6.1, propagating NMPs throughout the network. An illustration of this process is provided below. This is the final step in Market Operation. Next, the simulation will proceed into Real-Time Actions.

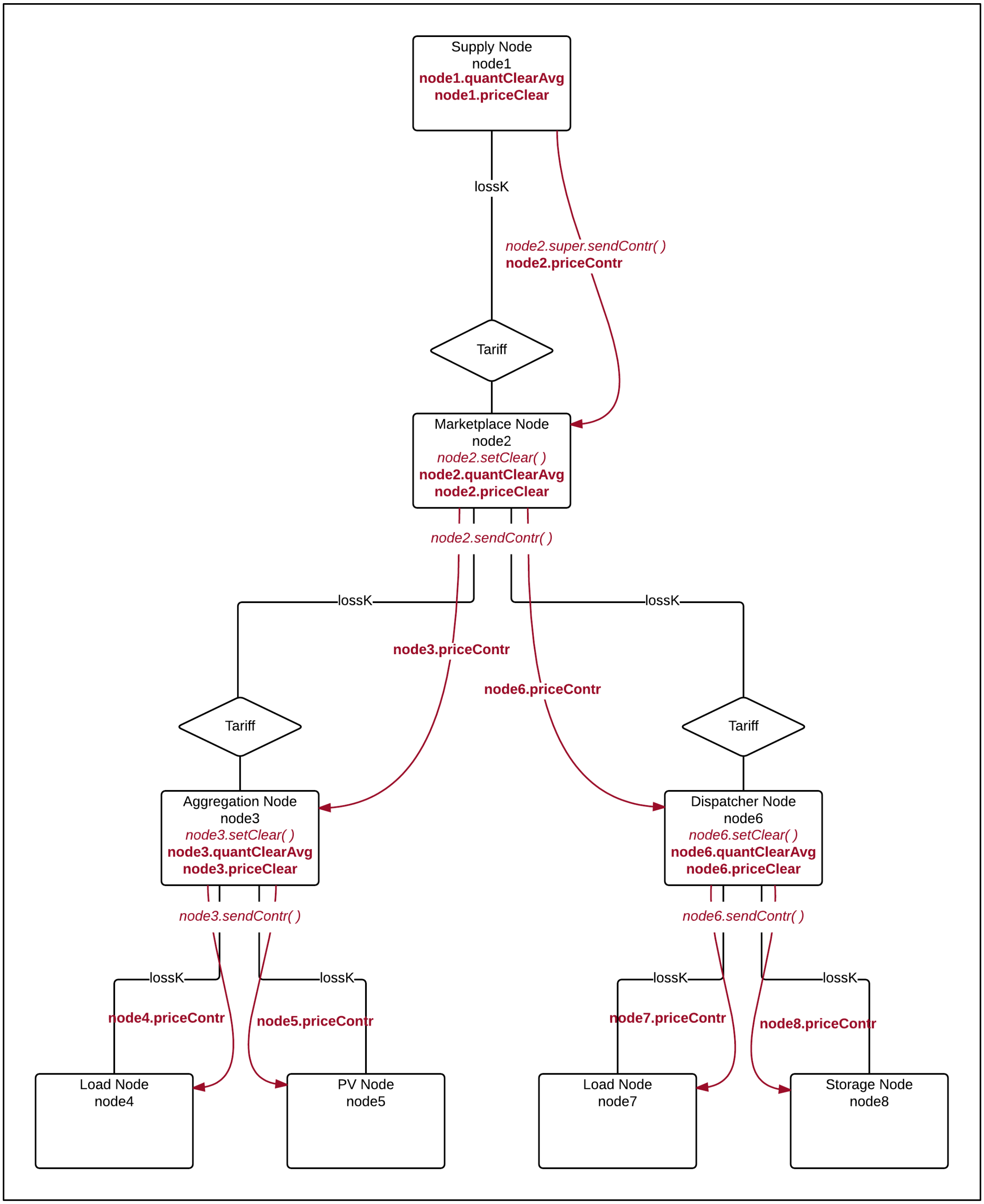


Figure 43: runMarket (3/3)

5.7 Simulation: Real-Time Actions

With Node Marginal Prices (NMPs) provided to every node in the system, the simulation can proceed into the Real-Time Actions phase. This consists of three steps: setting bottom-level node consumption, running dispatcher node functions, and determining system-wide energy flow.

Recall, the Market Operation process applied for the upcoming market duration. The market duration will often last for multiple time steps. The NMP established during Market Operation will apply for all of these time steps. However, the Real-Time Action simulation will be executed for each time step. This will continue until a Marketplace Node parameter *node.marketNext* equals the current time step; then Market Operation will execute before proceeding with Real-Time Actions.

The following processes will apply for every time step in the simulation.

5.7.1 Set Bottom-Level Node Consumption

First, bottom-level node consumption is determined by executing the method *node.setQuantActual* for all nodes in the system. From within the DDS object method *DDS.run*:

```
for n = 1:size(obj.nodes,2)
      obj.nodes(n).setQuantActual;
end
```

When this method is called by each node, its checks the *node.type* parameter. Bottom-level nodes (i.e. Load, Storage, PV Nodes) will use the NMP and their behavior model to determine their actual energy consumption for the market period. For details on proposed behavior models and actors' response to price signals, reference Section 6.

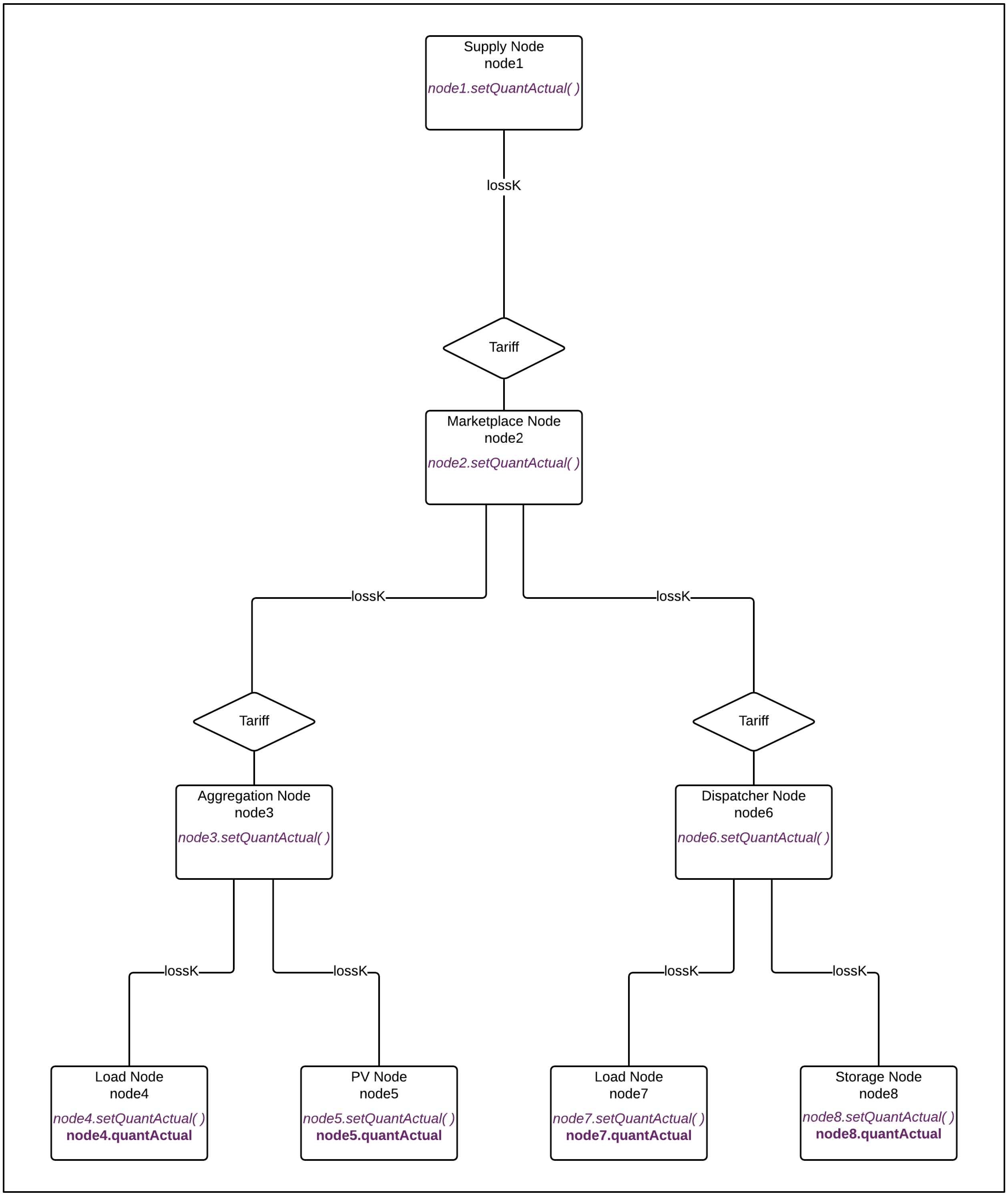


Figure 44: setQuantActual

The DTDM simulation overrides node behavior models in only one case: Dispatchable Storage Nodes. Storage Nodes include a parameter *obj.storageDispatchFlag*. If this parameter is true, the storage node is controlled by its Dispatcher supernode. As will be described in the next subsection, simulating dispatchable storage requires iterative simulation. To this end, any dispatchable storage node will set *obj.quantActual = 0* at this point in the simulation.

5.7.2 Determine System-Wide Energy Flow

After bottom-level node consumption has been established, the DTDM simulation performs a simplified estimation of system energy flow. This is accomplished with the node object method *node.simEstimate*, which estimates cumulative energy flow for its DTDM subsystem.

If no Dispatcher Nodes exist in the system, *DDS.top.simEstimate* is immediately called, providing the *quantActual* values for all nodes in the system. If Dispatcher Nodes exist in the system, iterative simulations are required. These simulations utilize *simEstimate*, so it is described here.

The method *simEstimate* polls the current node's subnodes. If a subnode's *quantActual* parameter is empty, the subnode calls *simEstimate*. If the subnode's *quantActual* parameter is not empty, it is translated to a supernode quantity, using the linkage loss constant definition.

$$Q' = Q + kQ^2$$

The sum of all subnode translated quantities provides the specified node's *quantActual*.

```
function simEstimate(obj)
      if isempty(obj.quantActual)
      % if quantActual is empty, poll subnodes and sum
            obj.quantActual = 0;
```

```
                for n = 1:size(obj.sub,2)
                % for all subnodes
                      obj.sub(n).simEstimate;
                      additionalDemand = obj.sub(n).quantActual + ...
                            obj.sub(n).lossK*obj.sub(n).quantActual^2;
                      obj.quantActual = obj.quantActual + additionalDemand;
                end
          end
    end
```

This method of estimating system energy flow has limitations. For one, the linkage loss constant used by the DTDM Market Operation is used here, unmodified. This prevents any analysis on the sensitivity of the linkage loss constant and its impact on the system. This is considered in Section 8.

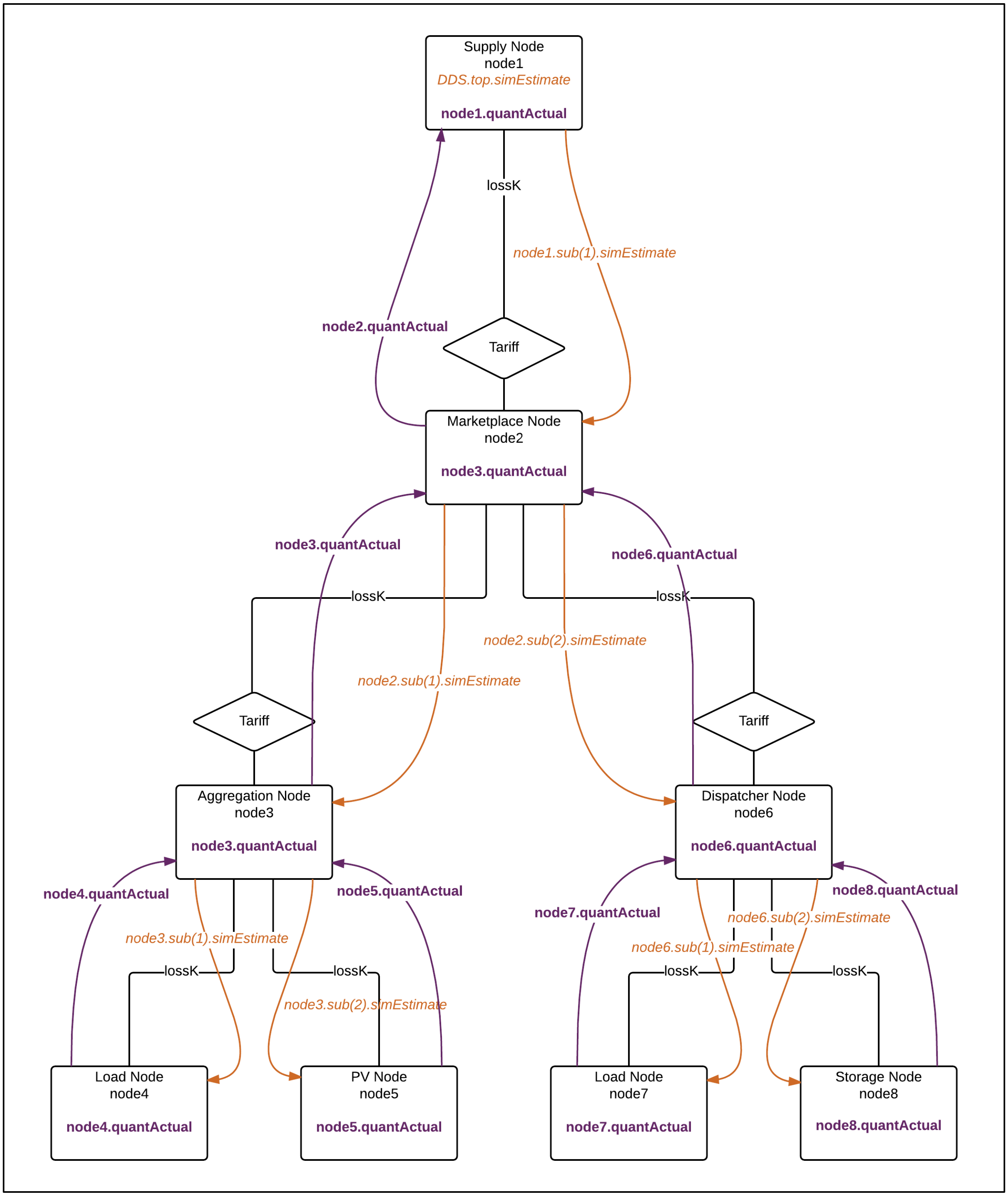


Figure 45: simEstimate, no Dispatch Nodes

Calling *DDS.top.simEstimate* completes the Real-Time Actions; simulation can proceed to Settlement. If the DTDM network does not contain Dispatcher Nodes, this was executed immediately after determining the bottom-level node consumption. However, if Dispatcher Nodes are included in the network, the Dispatcher Node process is required before calling *DDS.top.simEstimate*.

5.7.3 Run Dispatcher Nodes

If the DTDM network includes Dispatcher Nodes, then multiple iterations of *simEstimate* are required to determine the system energy flow.

Dispatcher Nodes represent fast-loop control within the DTDM network. For example, a household-level PV-Storage unit could be configured to prevent grid exports; this DTDM subsystem would include a Dispatcher Node that controls the energy storage based on real-time PV generation.

In practice, Dispatcher Nodes are constantly monitoring the real-time energy flow of their subsystems and adjusting the consumption (or generation) of another node. The node controlled by a Dispatcher Node is considered dispatchable. In the DTDM simulation, only storage nodes can be considered dispatchable; however, practical dispatch may also include demand-responsive loads and controllable generation units.

In simulation, dispatch levels must be defined. Because Dispatch Nodes respond to the simulated energy flow of their DTDM subsystem, the simulated response of a Dispatch Node also depends on the simulated response of all Dispatch Nodes in its DTDM subsystem. Note, it is assumed that the fast-loop control of dispatchable nodes is much faster than the market duration.

Dispatch levels are determined during initialization, when a node is assigned as a Dispatch Node. This is accomplished in the method *node.setDispatch*. The method *node.determineDispatchLevel* is called, which examines the existing DTDM network and updates *node.dispatchLevel*. This parameter reflects the number of Dispatch Node “beneath” the specified Dispatch Node. For example, NodeA is assigned as a Dispatch Node. It is the only Dispatch Node in the DTDM network: NodeA’s dispatch level is 1. Next, NodeB is assigned to be a Dispatch Node. NodeB is above NodeA in the DTDM hierarchy: NodeB’s dispatch level is 2.

The method *node.setDispatch* will also update other Dispatch Node’s dispatch levels, via *node.increaseDispatchLevel*, when they are impacted by a newly assigned Dispatcher Node. Continuing the example above, NodeC is assigned to be a Dispatch Node. NodeC lies below NodeA. NodeC’s dispatch level is 1, NodeA’s dispatch level is updated to 2, and NodeB’s dispatch level is updated to 3.

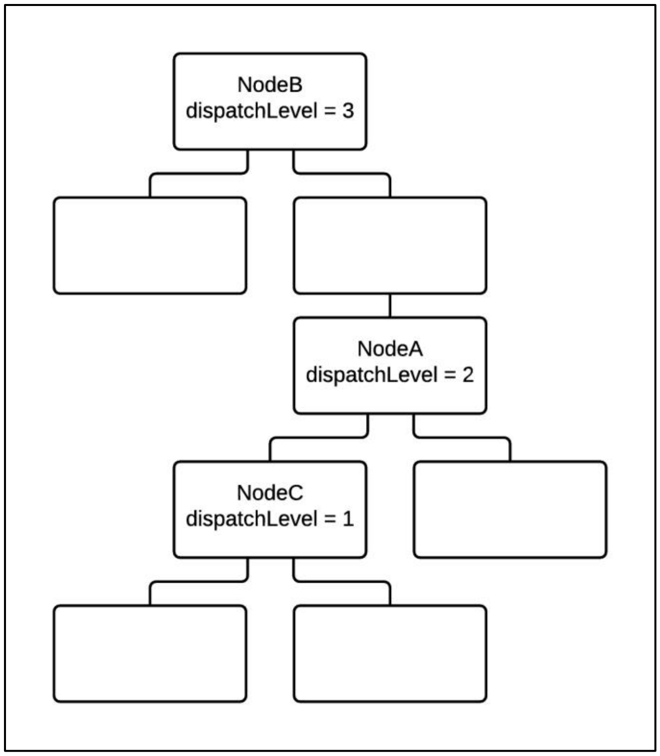


Figure 46: Example Dispatch Levels

Each node's dispatch level is set during initialization. When the simulation is executed the DDS object stores the highest Dispatch Node dispatch level as *DDS.dispatchLevels*. This determines how many iterative simulations are required for each timestep.

The iterative dispatch simulations are executed immediately after setting the bottom-level node consumption. Stepping through the dispatch levels, each Dispatcher Node performs *simEstimate* for its subsystem. It then executes the method *runDispatch*, which represents its fast-loop response. From the DDS object, within the current *DDS.run* timestep:

```
for lvl = 1:obj.dispatchLevels
% for each dispatch level
      for n = 1:size(obj.dispatchers,2)
      % check all dispatchers
            if obj.dispatchers(n).dispatchLevel == lvl;
```

```
            % if dispatcher level matches currently examined level,
            update quantActual
                  obj.dispatchers(n).simEstimate;
                  obj.dispatchers(n).runDispatch;
            end
      end
end
```

Recall, the designated dispatchable storage node set its *quantActual* to zero during *setQuantActual*. Thus, the result of *simEstimate* represents the sum of non-dispatchable energy flow.

```
nonDispatchQuant = obj.quantActual;
```

The Dispatcher Node method *runDispatch* represents the fast-loop response to measured subsystem energy flow.

Within *runDispatch*, the Dispatcher Node accomplishes two tasks. One, it determines a target quantity for the dispatchable storage node. Two, it sets the quantity for the dispatchable storage node.

In the DTDM simulation, a Dispatch Node can be set to either “contract” or “dynamic” mode. This is stored in the parameter *node.dispatchType*. For “contract” type dispatch, the target quantity is always set in an attempt to match the quantity indicated by the NMP on the demand curve submission.

```
targetQuant = obj.quantContrAvg;
```

“Contract” type dispatch is useful, but it may overestimate the value in matching the NMP to its demand curve submission. Consider a Dispatcher Node that underpredicted its non-dispatchable storage, faces a low Prediction Tariff penalty, and has a high opportunity cost of energy storage discharge. The optimal response for this node may be to purposely fail to meet its contracted quantity. It would face Prediction Tariff penalties, but those penalties would be less than the storage opportunity cost. This type of dispatch is categorized as type “dynamic”.

“Dynamic” type dispatch seeks this optimal target dispatch quantity. First, the Dispatcher Node replaces its DTDM network supernode with a Supply Node. This supply node uses the NMP to develop its supply curve. All tariffs between the Dispatcher Node and its actual supernode are left intact; they will be considered during optimal dispatch.

Note, a special case is additionally considered for “dynamic” Dispatcher Nodes. If the Dispatcher Node’s actual supernode is a Marketplace Node, the Dispatcher Node is considered an “in-line dispatcher”. Recognizing the role of this Dispatcher Node is to optimize energy flow to the larger energy network, this specific case also incorporates the tariffs between the Marketplace Node and its Supply supernode. Certainly, this may not be the only case in which this type of dispatch is useful. However, it is the only case implemented in the current DTDM simulation.

The Dispatcher Node then runs *market* to determine its updated clearing price and quantity.

The demand curve used for *market* is the aggregation of the dispatchable storage translated demand curve and an inelastic curve representing the non-dispatchable energy consumption.

```
obj.curve = obj.dispatchNode.curveSubmitTrans;
obj.curve(:,1) = obj.curve(:,1) + ...
      obj.Qbin*round(nonDispatchQuant/obj.Qbin);
```

This curve is adjusted, based on its borne tariffs. Note, this does not require demand curve translation, as there is no linkage loss between the Dispatcher Node and its NMP-derived Supply Node.

This adjusted demand curve and NMP-based supply curve are entered into the *market* function. The clearing point represents the optimal behavior. Thus, the clearing quantity determines the Dispatcher Node's target quantity.

```
[clearQ,clearP] = market(obj.super.curve,obj.curveSubmit);
targetQuant = clearQ;
```

Both "contract" and "dynamic" type Dispatcher Nodes provide a target energy quantity. This value is compared to the total non-dispatchable quantity found from *simEstimate.*

```
dispatchQuant = targetQuant - nonDispatchQuant;
```

This is the desired dispatchable quantity, from the Dispatcher Node's perspective. It is assumed that the Dispatcher Node directly controls the dispatchable storage node's energy quantity directly, without using indirect price signals. Thus, the Dispatcher simply translates this quantity, based on the linkage losses, and commands the dispatchable subnode.

```
k = obj.dispatchNode.lossK;
if k==0
      obj.dispatchNode.quantActual = dispatchQuant;
else
      obj.dispatchNode.quantActual =...
            (-1+sqrt(1+4*k*dispatchQuant))/(2*k);
```

```
end
```

As with all Storage Nodes, this value is compared to the storage node's State of Charge (SOC). The *quantActual* value for a Storage unit cannot result violating the SOC limits. This is covered in more detail in Section 6.

With the dispatchable storage consumption established, the simulation can proceed. If more dispatch levels are remaining, they execute the same *runDispatch* process. If no dispatch levels are remaining, the simulation calls *DDS.top.simEstimate* to determine the final system-wide energy flow. This concludes the simulated Real-Time Actions.

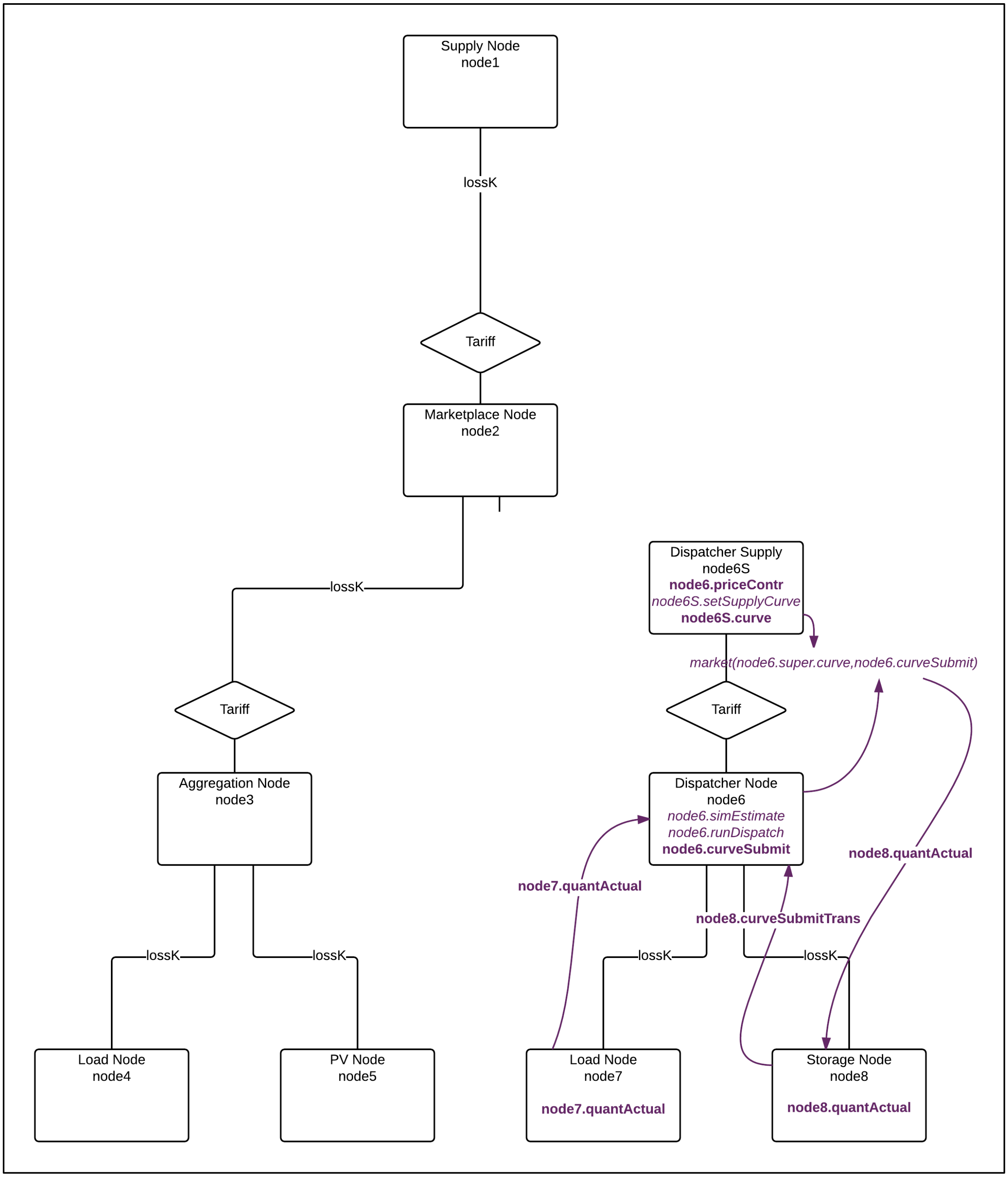


Figure 47: runDispatch Process

5.8 Simulation: Settlement

After Real-Time Actions are completed, the simulation proceeds to the Settlement phase. Like Real-Time Actions, this is accomplished for every time step. In simulation, settlement consists of updating the node and tariff meter arrays. This is accomplished with methods *node.meterUpdate* and *tariff.meterUpdate*.

```
for n = 1:size(obj.nodes,2)
      obj.nodes(n).meterUpdate;
end
for n = 1:size(obj.tariffs,2)
      obj.tariffs(n).meterUpdate;
end
```

All meter arrays include an entry for each timestep in the simulation. To limit recorded date, only nodes and tariffs with a true value for *obj.mFlag* are updated; the default value is true.

Node meters include the following direct updates:

```
timestep = obj.DDS.t;
obj.meterPriceContr(timestep) = obj.priceContr;
obj.meterQuantContrAvg(timestep) = obj.quantContrAvg;
obj.meterPriceClear(timestep) = obj.priceClear;
obj.meterQuantClearAvg(timestep) = obj.quantClearAvg;
obj.meterQuantActual(timestep) = obj.quantActual;
```

Additionally, payments and revenue are recorded. Subnode revenue is provided by selling energy to subnode's at their NMP. Supernode payment is based on purchasing energy at the NMP. Tariff payments are based on each tariff's *settle* method. The *settle* method is based on the tariff's specific instance. Examples are provided in Section 6. Final revenue is the subnode revenue less the supernode and tariff payments.

```
subRevenue = 0;
for n = 1:size(obj.sub,2)
      subRevenue = subRevenue + ...
            obj.sub(n).quantActual*obj.sub(n).priceContr;
end
obj.meterSubRevenue(timestep) = subRevenue;

if ~isempty(obj.super)
      superPayment = obj.quantActual*obj.priceContr;
else
      superPayment = 0;
end
obj.meterSuperPayment(timestep) = superPayment;

tariffPayment = 0;
for n = 1:size(obj.tariffs,2)
      obj.tariffs(n).settle;
      tariffPayment = tariffPayment + obj.tariffs(n).revenue;
end
obj.meterTariffPayment(timestep) = tariffPayment;
obj.meterRevenue(timestep) = subRevenue - superPayment - tariffPayment;
```

Storage nodes record one additional meter value: the timestep's state of charge. This uses the storage parameter, *obj.storageEnergyPct*.

```
if strcmpi(obj.type,'storage')
      obj.meterStoragePct(timestep) = obj.storageEnergyPct;
end
```

Tariff meter updates only record the subnode's actual energy quantity and revenue for the timestep. Note, recording time-varying parameters may also be useful and should be added to the simulation when appropriate.

```
timestep = obj.DDS.t;
quantActual = obj.sub.quantActual;
obj.meterQuantActual(timestep) = quantActual;
obj.meterRevenue(timestep) = obj.revenue;
```

The meter arrays will be used for analysis at the end of the simulation. With the Settlement step complete, the simulation can now proceed to the next timestep, returning the process described at the beginning of Section 5.6.

5.9 Analyze Results

Upon completion of the simulation, all records are stored as meter arrays at each node and tariff object. This can make it difficult for robust analysis, or later access. The DDS object method *DDS.saveMeters* is used to save all meter arrays to the workplace.

Often, it is useful to consider the overall results for owners within the system. Each owner may have multiple nodes and tariffs. The method *owner.edgeSet*, as described during initialization in Section 5.5, provides a useful. Recall, *owner.edgeSet* classified an owner's nodes and tariffs based on the final DTDM network configuration. Edge nodes and tariffs either indicate energy generation, energy flow to another party, or energy consumption. These nodes and tariffs will make it possible to easily determine each owner's net energy and cash flow; it ignore all energy and cash flow internal to an owner's subnetwork.

The method *owner.edgeAggMeter* analyzes the meter values for all edge nodes and tariffs. It determines the following values for each timestep: *edgeMeterQuantIn, edgeMeterQuantOut, edgeMeterLoad, edgeMeterRevenue*. Further, this is used to determine the effective price of energy for energy consumed or generated at timestep, *edgeMeterPriceEff* from the following:

```
obj.edgeMeterPriceEff = -
obj.edgeMeterRevenue./obj.edgeMeterQuantLoad;
```

This can provide an overall effective price of energy, averaged over the entire simulation period:

```
obj.edgeAggPriceEff = -
sum(obj.edgeMeterRevenue)/sum(obj.edgeMeterQuantLoad);
```

Certainly, there are many ways to interpret the results of the DTDM simulation. The meter arrays for each node can provide additional insights. Furthermore, additional parameters can be recorded by adding more meter arrays to the simulation.

Calling the method *node.analysisSummary* after a simulation provides the following summary for the node. All values are determined by examining the node's meter arrays.

```
--Node "ps26 Market" Analysis--
SUMMARY
 Net Energy:                -1.4426 kWh
 Avg Energy Flow:           -0.024043 kWh/min
ENERGY IMPORTS
 Net Energy:                0.5273 kWh
 Avg Energy Flow:           0.022926 kWh/min
 Total Time:                23 min
 Peak Energy Flow:          0.0538 kWh/min
 Payments (Super/Tariffs): $0.015528
 Avg Purchase Price:       $0.029447/kWh
ENERGY EXPORTS
```

```
 Net Energy:                 1.9699 kWh
 Avg Energy Flow:            0.053241 kWh/min
 Total Time:                 37 min
 Peak Energy Flow:           0.1094 kWh/min
 Revenue (Super/Tariffs):  $0.058525
 Avg Sale Price:           $0.02971/kWh
ENERGY IMPORT/EXPORT
 Effective Arbitrage Rate: $0.00026219/kWh
```

In addition to the summary provided above, one way to analyze system performance is to quantify the energy ramp rate for a given node. This is accomplished using the method *node.analyzeRamp*. This method uses the node's record energy consumption, stored in *node.meterQuantActual*, to determine the simplified moving average energy (*node.analysisQuantAvg)*, equivalent simplified moving average power ramp (*node.analysisQuantRamp*), the exponential decay running moving energy (*node.analysisQuantAvgED)*, and the equivalent exponential decay moving average power ramp (*node.analysisRampED*).

When *node.analyzeRamp* is called, it required the input variable *rampWindow*. This specifies the how far back the method looks, when determining the moving average. Selection of this variable will depend on the goals of the analysis.

For the simplified moving average, the quantity values recorded in *node.meterQuantActual* are padded, using the initial value of the array. As a result, the first entry of the array has disproportionate weight for the results with index 1:*rampWindow*. This is a compromise, better alternatives certainly exist, such as the exponential decay moving average.

```
Qactual = obj.meterQuantActual;
windowArray = (1/rampWindow)*ones(1,rampWindow);
```

```
Qpad = padarray(Qactual,[rampWindow-1,0],Qactual(1),'pre');
Qavg = filter(windowArray,1,Qpad);
Qavg = Qavg(rampWindow:end);
obj.analysisQuantAvg = Qavg;
obj.analysisRamp = zeros(size(Qavg));
obj.analysisRamp(2:end) = 60*(Qavg(2:end)-Qavg(1:end-1));
```

As shown above, the simplified moving average power ramp is defined by the change in the simplified moving average energy for each timestep, adjusted from energy per minute to average power per minute.

The exponential running average does not require padding. Additionally, the exponential moving average, by definition, places more emphasis on the “nearby” quantity values. For a detailed description of the exponential moving average, and a comparison to the simplified moving average, see Section 6.8 Ramping Tariff.

```
alphaED = 2/(rampWindow+1);
windowArrayED =repmat(1-alphaED,1,rampWindow).^(1:rampWindow);
windowArrayED = windowArrayED/sum(windowArrayED);
QavgED = filter(windowArrayED,1,Qpad);
QavgED = QavgED(rampWindow:end);
obj.analysisQuantAvgED = QavgED;
obj.analysisRampED = zeros(size(QavgED));
obj.analysisRampED(2:end) =60*(QavgED(2:end)-QavgED(1:end-1));
```

Like the simplified moving average, the exponential moving average ramp rate is defined by the change in the moving average energy for each timestep, adjusted from energy per minute to average power per minute.

For both the simplified and exponential moving average ramp values, it is useful to consider the cumulative sum of positive and negative ramping over the examined simulation. This is accomplished in the method, then saving to the Matlab workspace.

```
posRampSum = sum(obj.analysisRamp(obj.analysisRamp>0));
negRampSum = sum(obj.analysisRamp(obj.analysisRamp<0));
posRampSumED = sum(obj.analysisRampED(obj.analysisRampED>0));
negRampSumED = sum(obj.analysisRampED(obj.analysisRampED<0));

% Save variables to workspace
assignin('base','posRampSum',posRampSum);
assignin('base','negRampSum',negRampSum);
assignin('base','posRampSumED',posRampSumED);
assignin('base','negRampSumED',negRampSumED);
```

In general a “less rampy” node will have a lower cumulative sum for both positive and negative ramping than a “more rampy” node. Note, this approach does not make a distinction between the magnitude or duration of ramping, only the product of the magnitude and duration. Different analysis goals may seek to instead focus on either the magnitude or duration of ramping.

This method for analyzing node ramp rates is particularly useful when comparing results between simulation instances. This will be applied in the case studies in Section 7.

This concludes the description of the general DTDM simulation. The general description does not specify the details of behavior models or tariff structures included in the simulation. This is by design; the simulation must be flexible and allow implementation of various nod behavior models and tariff structures. However, proposal for each have been included in the simulation. Each will be described in Section 6.

6 Simulation: Behavior Models and Proposed Tariffs

6.1 Overview

Simulating a DTDM deployment requires: establishing the DTDM network; setting market parameters; representing actors' actions through behavior models; and defining tariff structures and parameter update protocols. This section describes simplified behavior models, for use in simulation, and proposed tariff constructions, for use in simulation and practical implementation. The behavior models are designed to provide reasonable and flexible representations of how market actors develop demand curves and determine their final energy consumption. The proposed tariffs are designed to capture critical externalities, with flexibility for system operators in practical implementation.

Behavior models must describe an actor's energy consumption preferences, prediction of consumption preferences as a demand curve, and actual energy consumption in response to prices and tariffs during Real-Time Actions. In some cases, consumption may be entirely negative, i.e. generation.

Behavior models described include: Load Nodes, PV Nodes, and Storage Nodes. Each includes a fundamental approach, with multiple possible variations.

Tariff designs are also proposed, to meet anticipated DTDM participant concerns. The DTDM process in Section 3.5.3 specifies a tariffs design moving from *structure* to *parameters* to *instance* to *settlement function* to *tariff curve*. However, proposed tariff designs may or may not follow this process. This is for two, non-mutually exclusive reasons: it has been done for convenience, where the tariff curve is generated directly; and it is a consequence of the iterative process of simulation and DTDM rule development.

Proposed tariff designs described include: Capacity; Ramping, Simplified Moving Average; Ramping, Exponential Moving Average; and Flat-Rate. Additionally, a proposed approach to Target Quantity tariffs is discussed, which can be applied to Volatility and Prediction Tariff types.

The behavior models and proposed tariffs described in this section are those used in the case study demonstrations in Section 7.

6.2 Load Node Behavior Model: Isoelastic with Anchor Point

The Load Node behavior model applies to bottom-level node that consists solely of electrical loads. A Load Node is described by the following parameters, each to be detailed: data source, prediction error, elasticity, anchor price, anchor quantity, and maximum quantity.

The Pecan Street dataset provides actual, minute-by-minute average power for household loads. This is considered a baseline, upon which the behavior model is established. The behavior model uses this data source for both demand curve generation and real-time actions.

The proposed behavior model for load nodes is an isoelastic price response, relative to a pre-established “anchor” price and quantity point.

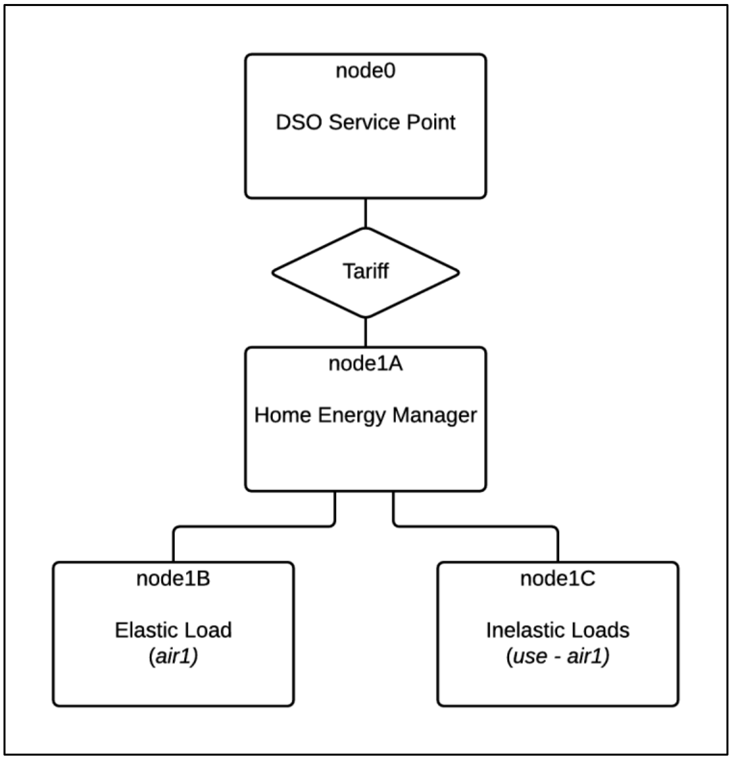


Figure 48: Two Load Nodes

This behavior model applies to both elastic and inelastic loads. A household may represented by multiple nodes, which are aggregated at a HEM node. Pecan Street data is used to establish the behavior of the contributing nodes, each of which has its own elasticity. Unless indicted, a simulated Pecan Street household has a HEM with two subnodes: one node for *air1* loads, i.e. elastic loads, and one node for all non-*air1* loads, i.e. inelastic loads.

6.2.1 Consumption Preferences and Demand Curve Generation

The Pecan Street baseline can be used to generate a demand curve. A load demand curve is described by the following parameters: price elasticity of demand; anchor energy quantity; anchor energy price for that quantity; prediction error; and maximum elastic demand. Note, most parameters are expected to change throughout the day, based on the end user's consumption patterns.

First, the anchor quantity and anchor price are established.

The anchor energy quantity, Q_{anchor}, and anchor energy price, P_{anchor}, represent a known or estimated point on the demand curve. A "learning" HEM could estimate this value based on the customer's behavior. Pecan Street data provide an anchor point for a given point in time: Q_{anchor} is the actual energy consumption recorded, while P_{anchor} is the retail rate observed by the customer for that period. However, the retail rate observed by the Pecan Street customer is unknown.

For simulation, and when the customer's observed retail rate is unknown, a baseline retail rate can be established. The Flat-Rate Equivalent price, P_{FRE}, is determined to be the constant volumetric retail rate that balances customer energy consumption and payments with wholesale energy transfer and revenue. In other words, P_{FRE} represents an "ideally tuned" retail rate. Determining this value requires customer load profiles, the physical system model, wholesale market data, and simulation over the examined time range.

Note, tuning P_{FRE} for a full DDS model will use the same value for all customers, using the aggregated energy consumption profile. This value would not change during the modeled time period, unless customers were exposed to TOU rates or demand charges.

Alternatively, rather than tuning P_{FRE}, it can simply be set to the average retail rate for the examined region.

For generating a load demand curve with Pecan Street data,

$$P_{anchor} = P_{FRE}$$

The anchor energy quantity, Q_{anchor}, is established using Pecan Street load data. The Pecan Street data provides the average power demand, in kW, during each one-minute period. For an arbitrary one-minute time period, *T*, and usage category *air1*, the energy in kWh is provided by

$$Q_T^{air1} = \frac{\langle Power \rangle_T^{air1}}{60}$$

For a market period of arbitrary length *M* minutes and starting at time *T*, Q_{anchor}^{air1} is provided by

$$Q_{anchor}^{air1} = \sum_{\tau=T}^{T+M} Q_\tau^{air1} = \frac{1}{60} \sum_{\tau=T}^{T+M} \langle Power \rangle_\tau^{air1}$$

For a household with only two subnode components, inelastic load is determined to be:

$$\langle Power \rangle_T^{inelast} = \langle Power \rangle_T^{use} - \langle Power \rangle_T^{air1}$$

where *air1* is the only elastic load extracted from the Pecan Street *use* category. In a general case:

$$\langle Power \rangle_T^{inelast} = \langle Power \rangle_T^{use} - \langle Power \rangle_T^{elastcat1} - \langle Power \rangle_T^{elastcat2} \ldots$$

This is then used to determine the inelastic load's anchor quantity, using the equation previously stated.

$$Q_{anchor}^{inelast} = \frac{1}{60} \sum_{\tau=T}^{T+M} \langle Power \rangle_\tau^{inelast}$$

To generate a fixed point on the demand curve, the anchor quantity is rounded to the nearest Qbin interval.

$$Q_{anchor} = round\left(\frac{Q_{anchor}}{Qbin}\right)Qbin$$

With anchor quantity and anchor price values determined, a load is next characterized by its price elasticity of demand. Price elasticity of demand is defined as

$$\varepsilon = \left|\frac{dQ/Q}{dP/P}\right|$$

Without using the absolute value, this definition yields a negative number for a decreasing (i.e. typical) demand curve. However, the unsigned magnitude is common usage for describing a price elasticity of demand, so the absolute value definition is used.

This behavior models assumes that the elastic loads are isoelastic. Isoelastic demand curves have the same elasticity at every point, i.e. the elasticity does not change with quantity. The isoelastic demand function is defined as

$$Q = kP^{-\varepsilon}$$

where *k* is a constant (unrelated to *k* that describes system losses). With a known price elasticity of demand, the anchor values can be used to determine *k*. Recognizing that the customer purchased Q_{anchor} units of energy at P_{anchor}:

$$Q_{anchor} = k(P_{anchor})^{-\varepsilon}$$

$$k = \frac{Q_{anchor}}{(P_{anchor})^{-\varepsilon}}$$

Substituting back into the isoelastic demand function provides an equation for an arbitrary price:

$$Q = \frac{Q_{anchor}}{(P_{anchor})^{-\varepsilon}} P^{-\varepsilon} = Q_{anchor} \left(\frac{P}{P_{anchor}}\right)^{-\varepsilon}$$

Alternatively, expressing the isoelastic function with *Q* as the independent variable:

$$Q = kP^{-\varepsilon}$$

$$P = \left(\frac{Q}{k}\right)^{-\frac{1}{\varepsilon}}$$

$$P = \left(\frac{Q(P_{anchor})^{-\varepsilon}}{Q_{anchor}}\right)^{-\frac{1}{\varepsilon}} = P_{anchor} \left(\frac{Q}{Q_{anchor}}\right)^{-\frac{1}{\varepsilon}}$$

Both expressions, with either *P* or *Q* as the independent variable, are valid and useful.

As described in Section 2.5, there is no standard estimate for customers' price elasticity of demand. Additionally, the elasticity is expected to be time- and use-dependent for a given customer. Various economic models use values ranging from 0.05 to 0.96. This behavior model requires initialization with an estimated price elasticity; this must be selected based on the specific scenario.

With P_{anchor}, Q_{anchor}, and ε known, a demand curve can be generated by determining price values for each Qbin interval:

$$P = P_{anchor}\left(\frac{Q}{Q_{anchor}}\right)^{-\frac{1}{\varepsilon}}$$

However, this method has some additional considerations.

In the above equation, inelastic loads ($\varepsilon = 0$) result in infinite prices. This is consistent with our interpretation of an inelastic load (i.e. the quantity will not change based on price) but is mathematically suspect. However, our demand curve definition includes a positive vertical line at the LH limit and a negative vertical line at the RH limit. Therefore, an inelastic load's demand curve can be expressed as a single point. This point serves as both the LH and RH limit.

$$[P_{anchor}, Q_{anchor}], \qquad \varepsilon = 0$$

For an elastic load, the demand curve can be expressed as *n* points in the following form.

$$\left[nQ_{bin}, P_{anchor}\left(\frac{nQ_{bin}}{Q_{anchor}}\right)^{-\frac{1}{\varepsilon}}\right], \qquad \varepsilon > 0,\ Q_{anchor} \neq\ 0$$

However, consideration must be made for instances where $Q_{anchor} = 0$. In this case, the above expression would undefined. Yet, when using historical data as an anchor quantity, as a practical expectation, energy may have been consumed at a lower price point. Note, this also demonstrates one of the limitations of the isoelastic curve model, to be later addressed in more depth.

To confront this possibility, "nearby" historical data should be examined. For example, "nearby" values could extend 12 hours before the examined market period and 12 hours after the examined market period. This serves to determine if the category of elastic consumption was potentially valuable, albeit at a lower price point. For example, Pecan Street data may indicate no air conditioning energy consumption for a given market period. If the examined period was in July, air conditioning may have been valuable at a lower price point than the retail energy rate. However, if the examined period was in December, air conditioning may not have been valuable, regardless of the price.

If the “nearby” data includes demand in the load category, then set $Q_{anchor} = Q_{bin}$. This provides an opportunity for energy demand when $P < P_{anchor}$. But, by design, the curve will not result in consumption when $P > P_{anchor}$. This provides a revised demand curve expression for *n* points:

$$\left[nQ_{bin}, P_{anchor}\left(\frac{nQ_{bin}}{Q_{bin}}\right)^{-\frac{1}{\varepsilon}}\right], \quad Q_{nearby} > 0,\ Q_{anchor} = 0$$

Note, one limitation to this approach occurs when $P = P_{anchor}$. The historical data indicates this should result in $Q = 0$. However, the above expression results in $Q_{demand} = Q_{bin}$. This is considered a negligible limitation, as it only occurs at a specific price and, by design, Qbin is very nearly zero.

If there is no “nearby” demand in the data, then the demand curve should be considered inelastic at zero load, providing the demand curve expression:

$$[0, P_{anchor}], \qquad Q_{nearby} = 0,\ Q_{anchor} = 0$$

This leads to the next demand curve constraint. The elastic load demand curve expression above will provides price values for quantity values of infinite size. This is not realistic; there is a practical limitation to a load’s ability to consume energy, regardless of price of energy. To control for this, a load’s maximum demand quantity is established. For example, this would correspond to an air conditioner's maximum possible load. Absent device data plates, Pecan Street data can estimate this value by finding the maximum recorded demand, $\langle Power\rangle_{max}^{air1}$ and translating it to an energy quantity, for a market duration *M*, using the following equation:

$$Q_{max}^{air1} = \sum_{\tau=T}^{T+M} \frac{\langle Power\rangle_{max}^{air1}}{60} = \frac{M}{60}\langle Power\rangle_{max}^{air1}$$

Like Q_{anchor}, this value is rounded to the nearest Qbin interval.

$$Q_{max} = round\left(\frac{Q_{max}}{Qbin}\right)Qbin$$

This value provides the RH limit to the isoelastic demand curve.

As a similar practical limitation, a price cap, P_{cap}, is considered. An isoelastic curve approaches infinite *P* as *Q* approaches zero. However, if there is an upper limit to the price, there is a lower limit to the quantity at which the load will respond to potential price signals. Subsequently, demand curve points can be eliminated when the calculated price exceeds the price cap. This limit is expressed as

$$P_{anchor}\left(\frac{nQ_{bin}}{Q_{anchor}}\right)^{-\frac{1}{\varepsilon}} \leq P_{cap}$$

By rearranging terms, this can also be expressed as a limit on nQ_{bin}:

$$nQ_{bin} \geq Q_{anchor}\left(\frac{P_{cap}}{P_{anchor}}\right)^{-\varepsilon}$$

With an isoelastic demand curve, this provides a LH limit. Note, this may be larger than Q_{bi}. As a practical consequence, this eliminates unneeded information transfer.

In summary, Q_{max} and P_{cap} provide additional limits to the elastic load demand curve expression, which is summarized to be the following, for *n* points:

$$\left[nQ_{bin}, P_{anchor}\left(\frac{nQ_{bin}}{Q_{anchor}}\right)^{-\frac{1}{\varepsilon}}\right], \qquad \varepsilon > 0,\ Q_{anchor} > 0,\ Q_{anchor}\left(\frac{P_{cap}}{P_{anchor}}\right)^{-\varepsilon} \leq nQ_{bin} \leq Q_{max}^{air1}$$

As an illustration of elasticities, the following data from Pecan Street Household #22 will be used. The illustrated market duration is five minutes. P_{anchor} is set to be \$0.10/kWh.

Datetime	*use* (kW)	*air1* (kW)
01-Jun-2014 22:10:00	1.4320	1.1200
01-Jun-2014 22:11:00	1.4260	1.1150
01-Jun-2014 22:12:00	1.4200	1.1090
01-Jun-2014 22:13:00	1.3970	1.1080
01-Jun-2014 22:14:00	1.3110	1.1150

Table 8: Pecan Street Household 22 Data

The plots below shows the impact of both considering the entire load inelastic and considering a portion of the load elastic.

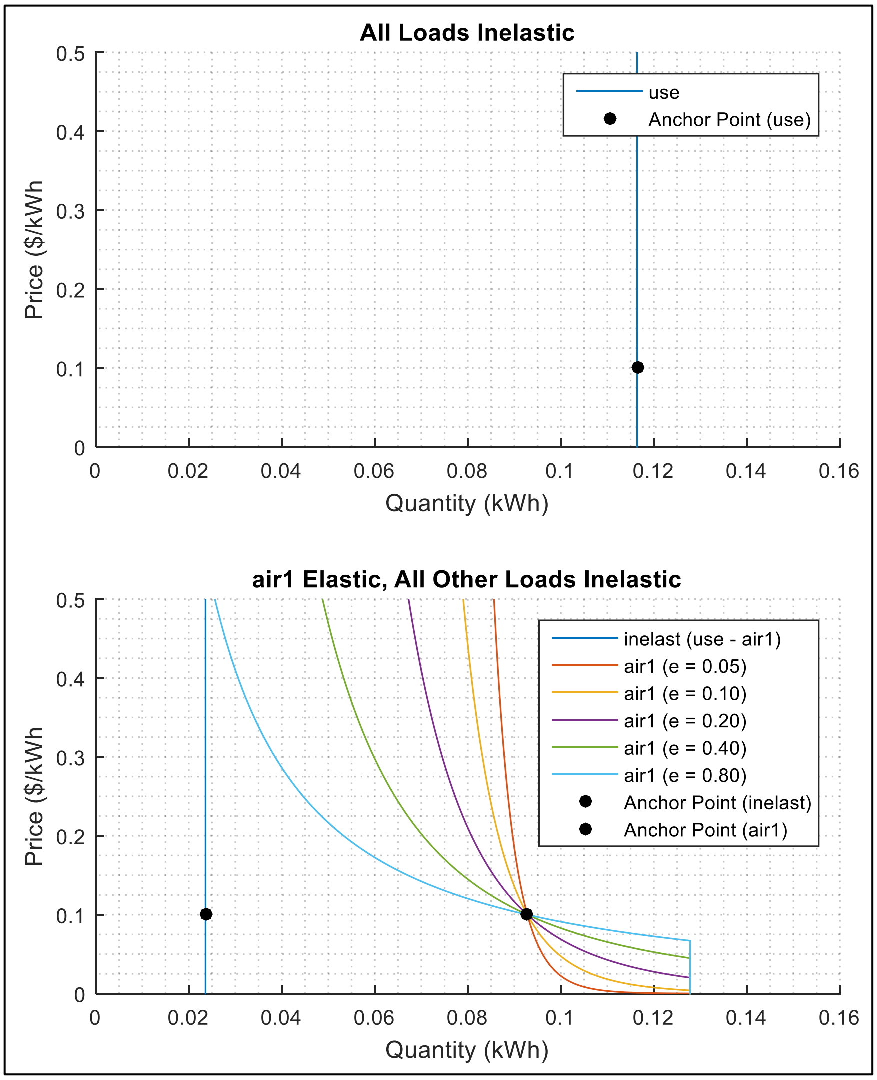


Figure 49: Isoelastic Demand Curve Examples

Some observations can be made from these illustrations. In the top plot, the entire load is considered inelastic, and the demand curve is a vertical line. In the bottom plot, *air1* is considered elastic, with various levels of elasticity illustrated. Notice, regardless of the elasticity, all *air1* demand curves pass through the anchor point. Additionally, as the elasticity approaches zero, the demand curve approaches a vertical line. However, all demand curves, regardless of their elasticity, are asymptotic to the horizontal axis. In other words, as the price approaches zero, all loads can be thought to say "why not?" as they seek to maximize their consumption. The maximum possible consumption is also clearly represented; this is the RH limit shared by all *air1* demand curves.

There is additional, relevant observation: in the bottom plot, for low elasticities, *air1* is approaches a horizontal line at large Q values. In certain cases, the slope of this line could violate the Pmin requirements imposed by the DTDM system operator. This occurs when the difference between two adjacent (Q,P) values is less than Pmin. If the Pmin requirement is violated, then demand curve is not monotonic with the required degree of precision. This would not be accepted by the supernode during Market Operation.

To prevent this violated, the isoelastic demand curve must be modified to maintain Pmin requirements. Specifically, when the unmodified slope is less than P_{min}/Q_{bin}, the slope is set to

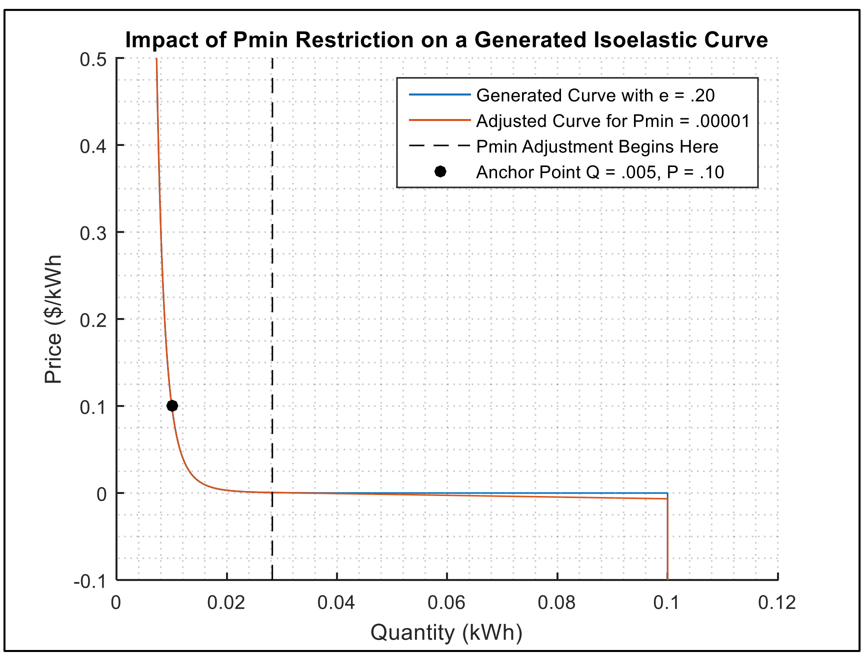


Figure 50: Example Impact of Pmin on an Isoelastic Demand Curve

exactly P_{min}/Q_{bin}. An example adjustment is illustrated in Figure 50.

Notice, for an isoelastic demand curve, the slope strictly decreases as the quantity increases. Thus, modification is accomplished to finding the first slope violation on the unmodified curve. The *(Q,P)* point before this violation establishes a reference price and quantity. All points of a greater quantity at set by using the following method:

$$P_{mod} = P_{ref} - \frac{P_{min}}{Q_{bin}}\left(Q - Q_{ref}\right), \qquad Q > Q_{ref}$$

Notice there is now a difference between the “true” demand curve and the modified curve. In fact, the curve now includes negative price values. The maximum error occurs at the Q_{max} value, the RH limit of the curve. This modification does not impact the ultimate consumption by the node, only their bid into the market. Thus the modification would only impact establishing clearing energy calculation if the market price falls between $P_{maxQ,unmodified}$ and $P_{maxQ,modified}$. It is expected this impact is small, when compared to other prediction errors. The DTDM system operator can reduce the impact of this error by setting Pmin to the smallest acceptable value. Additionally, the isoelastic demand curve itself is only a simplified behavior model; the “true” demand curve is not a wholly accurate representation of consumption preferences. Thus, this modification will only be used to meet DTDM system constraints and error will not be considered further.

Another illustration is provided. Again, data from Pecan Street Household 22 is used, as provided below. Like before, *air1* is considered an elastic load, with a price elasticity of 0.4. Additionally, *refrigerator1* is also considered an elastic load, with a price elasticity of 0.2. The market duration remains five minutes and P_{anchor} remains \$0.10/kWh. Additionally, a relatively large Pmin is used, to display the impact of Pmin slope modification.

Datetime	Market Period	*use*	*air1*	*refrigerator1*
01-Jun-2014 22:10:00	1	1.4320	1.1200	0.1170
01-Jun-2014 22:11:00		1.4260	1.1150	0.1170
01-Jun-2014 22:12:00		1.4200	1.1090	0.1170
01-Jun-2014 22:13:00		1.3970	1.1080	0.0940
01-Jun-2014 22:14:00		1.3110	1.1150	0
01-Jun-2014 22:15:00	2	1.3040	1.1090	0
01-Jun-2014 22:16:00		1.3010	1.1070	0
01-Jun-2014 22:17:00		1.2050	1.0120	0
01-Jun-2014 22:18:00		0.1930	0	0
01-Jun-2014 22:19:00		0.1530	0	0
01-Jun-2014 22:20:00	3	0.1240	0	0
01-Jun-2014 22:21:00		0.1280	0	0
01-Jun-2014 22:22:00		0.1280	0	0
01-Jun-2014 22:23:00		0.1200	0	0
01-Jun-2014 22:24:00		0.1180	0	0

Table 9: Pecan Street Household 22 Data

The plots in Figure 51 illustrate the impact of varying anchor quantities over time. Additionally, the right column of plots provide aggregated curves, illustrating the cumulative impact of a household with loads of differing elasticities.

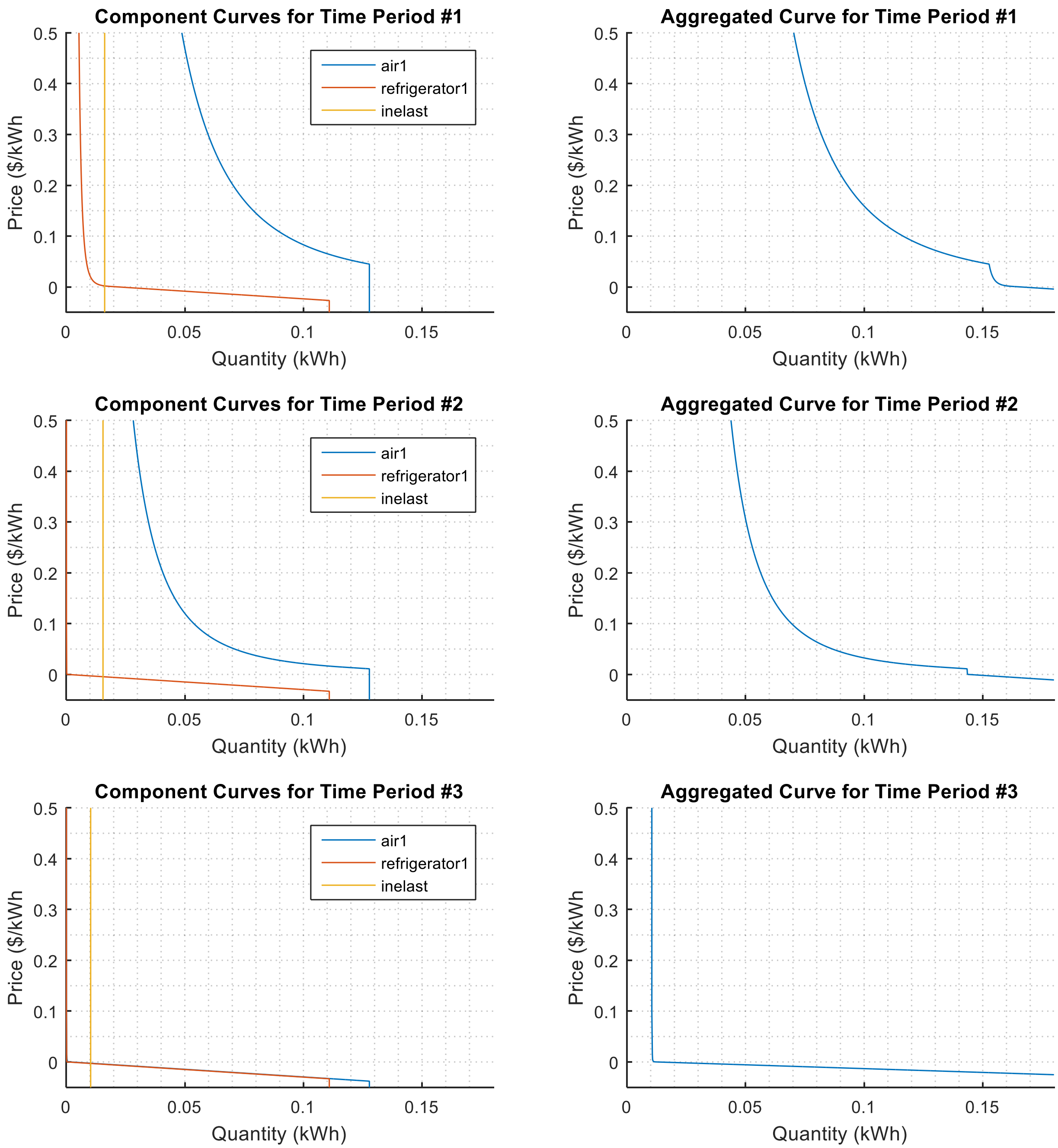


Figure 51: Example Load Curves with Aggregation

Finally, demand curve generation to allow for error in anticipating the future consumption preferences. For example, a HEM may use historic consumption patterns to predict future inelastic energy consumption. This prediction is cannot be made with certainty; there is always a possibility of error. This possibility is captured by a parameter: *predictErrorSD*. This parameter is the standard deviation of error, expressed as a percentage of the "true" demand. Thus, the percent error in prediction is determined using a random number generator:

$$Q_{anchor}^{estimate} = Q_{anchor}^{true}(1 + \max(-1, predictErrorSD * randn))$$

where *randn* produces a random number drawn from the standard normal distribution: the mean result of *randn* is 0, with σ = 1 and -σ = -1. Multiplying *randn* by the parameter *predictErrorSD* produces a percent error with a mean result of 0 and σ = *predictErrorSD* and -σ = -*predictErrorSD*; this is the percent by which the prediction will deviate from the "true" demand. Notice this value cannot fall below -1. This prevents the estimated demand from falling below Q = 0. There is no limit to incorrectly over-predicting future demand.

As a result, this parameter is used to add uncertainty to the "true" demand provided by the Pecan Street dataset. During real-time actions, this parameter will be ignored, and consumption will be driven by the "true" demand. Thus, the parameter reflects the inability to anticipate future consumption preferences. Note, when *predictErrorSD = 0*, the node will anticipate its future consumption preferences with perfect prediction.

These considerations are used by this isoelastic behavior model to develop demand curves using historical data. This is accomplished at the onset of the DTDM Market Operation.

6.2.2 Real-Time Actions

After submitting its demand curve during Market Operation, a load node will eventually receive a Node Marginal Price (NMP) for the upcoming market period. The NMP will be used to set the actual consumption of energy during the market period.

The NMP will impact energy consumption over the entire market period. The Pecan Street data provides minute-by-minute historical data, yet the market period can be set values greater than one minute. This behavior model considers the impact of the NMP on each discrete minute, regardless of the market duration. Thus, a new anchor quantity is established for the current time, *T*:

$$Q_{anchor}^{true,T} = \frac{\langle Power \rangle_T}{60}$$

The anchor quantity is not impacted by the prediction error parameter. This is considered the "true" consumption preference. Next, the NMP is used to determine the actual consumption, using the isoelastic quantity equation and the pre-established anchor price:

$$Q_{actual}^{T} = Q_{anchor}^{true,T} \left(\frac{P_{NMP}}{P_{anchor}} \right)^{-\varepsilon}$$

The point on the demand curve submission, as indicted by the NMP can be considered $Q_{contract}$. Notice, the minute-by-minute actual consumption quantity does not necessarily match this value. Even with perfect prediction, if the market period is greater than one-minute, this Q_{actual} will only be a fraction of $Q_{contract}$. Additionally, the actual quantity for each minute will respond independently to the NMP.

However, if the elasticity does not change and the node had no prediction error, the following observation can be made of the total energy consumption during the market period:

$$Q_{total} = \sum_{\tau=T}^{T+M} Q_{actual} = \sum_{\tau=T}^{T+M} Q_{anchor}^{true} \left(\frac{P_{NMP}}{P_{anchor}}\right)^{-\varepsilon} = \left(\sum_{\tau=T}^{T+M} Q_{anchor}^{true}\right)\left(\frac{P_{NMP}}{P_{anchor}}\right)^{-\varepsilon} = Q_{contract}$$

This demonstrates that the total actual consumption will match the contracted quantity for any market duration, when perfect preference prediction is achieved.

Additionally, the above equation for actual consumption holds for zero price elasticity in this case, it can be simplified to the following:

$$Q_{actual}^{T} = Q_{anchor}^{true,T}, \qquad \varepsilon = 0$$

This matches the expectation that inelastic demand will not respond to price signals.

Finally, consideration must be made for the maximum possible energy consumption. Like before, this value is determined using the historical dataset, but only considers a duration of one minute.

$$Q_{max} = \frac{1}{60}\langle Power\rangle_{max}$$

If the calculated energy consumption exceeds this value, the actual consumption will be set to Q_{max}. This also applies to non-positive price values. In summary, real-time consumption can be determined using the following equations:

$$Q_{actual}^{T} = \min\left(Q_{max}, Q_{anchor}^{true,T}\left(\frac{P_{NMP}}{P_{anchor}}\right)^{-\varepsilon}\right), \qquad P_{NMP} > 0$$

$$Q_{actual}^{T} = Q_{max}, \qquad P_{NMP} \leq 0$$

6.2.3 Limitations and Areas for Improvement

This behavior model is not without limitations. In particular, it treats each timestep as independent from all other timesteps. This is not realistic. Most loads will have temporal consumption dependencies; for example, current air conditioning energy consumption will certainly impact demand for future air conditioning energy consumption. Additionally, while this behavior model could support time-shifting elasticities, it does not include dependencies between consumption and elasticities. For example, the price elasticity for air conditioner energy consumption is certainly dependent on the previous consumption of air conditioner energy (i.e. price sensitivity decreases when approaching the boundaries of acceptable household temperature).

Another limitation to this behavior model is the isoelastic demand curve. One, this assumes all possible quantities are possible. However, this is not a valid assumption for most practical devices. Realistic demand curves should be “lumpy”, with a binary or stair-step shape. Two, the isoelastic anchor point curve does not work well for quantities with zero load in the data source. In these cases, the demand curve should not be drastically different from curves generated with other data source anchor quantities. Three,

Many of these limitations would be improved with a more robust behavior model, designed for the specific load type and application.

For example, an air conditioner behavior model would, ideally, reflect the thermal characteristics of its location. Time-varying user parameters would be target temperature and the customer's price-vs-performance sensitivity. The demand curve would be generated by determining the range of possible temperatures, based on the location's thermal characteristics, the air conditioner's range of operation, and the user parameters.

Similarly, other specific devices would have different behavior models, based on their goals and operating characteristics.

Finally, a robust load behavior model would also include projections for future prices. A device, optimized to minimize the cost to operate, may elect to not run, if the price is expected to fall in the near future.

6.3 PV Node Behavior Model

The PV Node behavior model applies to bottom-level node that consists solely of photovoltaic generation. A PV Node is described by the following parameters, each to be detailed: data source and prediction error.

The Pecan Street dataset provides actual, minute-by-minute average power for PV generation, in the category *gen*. This is considered a baseline, upon which the behavior model is established. The behavior model uses this data source for both demand curve generation and real-time actions.

6.3.1 Consumption Preferences and Demand Curve Generation

A PV Node's consumption is always negative; it never imports energy, only exports. The Pecan Street dataset provides the actual generation for each timestep. It is assumed that a PV array will always generate the maximum possible energy for a given period, unless the NMP is negative. In those cases, the PV array will curtail its generation.

To generate a demand curve, the PV Node must anticipate its future generation. This uses the Pecan Street historical record, stored in the parameter *node.pvDataSource*. Note, the Pecan Street dataset stores generation as average power (in kW) and as positive quantities. The parameter *node.pvDataSource* maintains this convention, but demand curves and actual consumption will reflect the DTDM energy and demand convention.

Additionally, the Pecan Street dataset includes instances of negative generation in its PV category, *gen*. In all cases, the imported energy is very small. It is unknown why this occurs, perhaps supporting equipment loads. Regardless, all negative values are removed from the data, to simplify the behavior model.

```
ps26gen2014(ps26gen2014<0) = 0;
DDS2.addNode('ps26 pv','ps26','ps26 Dispatch');
      DDS2.recent.type = 'PV';
      DDS2.recent.pvDataSource = ps26gen2014;
```

The parameter *pvDataSource* is referenced when generating a demand curve, which occurs in the method *node.setPVCurve*. First, the historical record and market duration is used determine the anticipated energy consumption. This uses the DTDM demand convention.

```
    estQ     =     -sum(obj.pvDataSource(refStart:refStart+(marketDuration-
1)))/60;
```

However, it is not assumed that the PV Node can anticipate future generation with perfect accuracy; an error factor is implemented.

The error factor is described by *node.predictErrorSD*. This is the same parameter used by Load Nodes, with the same implementation. The parameter is the standard deviation of error, expressed as a percentage of the "true" generation. Thus, the percent error in prediction is determined using a random number generator:

```
if obj.predictErrorSD ~= 0
    errorFactor = max(-1,obj.predictErrorSD*randn);
    estQ = estQ*(1+errorFactor);
end
```

where *randn* produces a random number drawn from the standard normal distribution: the mean result of *randn* is 0, with σ = 1 and -σ = -1. Multiplying *randn* by the parameter *predictErrorSD* produces a percent error with a mean result of 0 and σ = *predictErrorSD* and -σ = -*predictErrorSD*; this is the percent by which the prediction will deviate from the "true" demand. Notice this value cannot fall below -1. Because the true demand is a negative number, this prevents the estimated demand from rising above Q = 0. There is no limit to incorrectly over-predicting future generation.

As a last step, this estimated quantity is rounded to the nearest Qbin.

```
estQ = obj.Qbin*round(estQ/obj.Qbin);
```

This is provided the PV Node's predicted generation, but the node is required to generate a demand curve.

If $Q_{est} < 0$, it is recognized that the predicted energy generation provides the LH limit of the curve; no more energy can be generated than this value. Additionally, Q = 0 provides the RH limit of the curve; the PV Node is unable to use energy imports. Finally, the marginal cost of PV generation is $0/kWh, thus the "true" demand curve is a horizontal segment, at P = $0/kWh, between $Q = Q_{est}$ (a negative number) and $Q = 0$.

However, the DTDM rules preclude demand curves with horizontal segments; the Pmin restriction must be observed. To this end, the horizontal segment is adjusted to the smallest possible slope: -Pmin/Qbin. This slope adjustment can use either the LH or RH limit as a reference point. If the LH limit is used, the PV Node will require a NMP slightly less than P = 0 before curtailing its generation. Specifically, curtailment will only occur for:

$$P_{NMP} < \frac{Q_{est}}{Qbin} Pmin$$

Because this curtailment condition is not straightforward for Real-Time Action, the alternative choice is made. The RH limit will be used as the reference point for the Pmin adjustment of the demand curve. Thus the LH limit is slightly higher than P = 0. The curve is then generated by setting the Q and P values as follows:

```
obj.curve(:,1) = [estQ:obj.Qbin:0];
obj.curve(:,2) = [-(estQ/obj.Qbin)*obj.Pmin:-obj.Pmin:0];
```

Alternatively, if $Q_{est} = 0$, then the demand curve is a single point: [0,0]. It is not possible for $Q_{est} > 0$, based on *pvDataSource.*

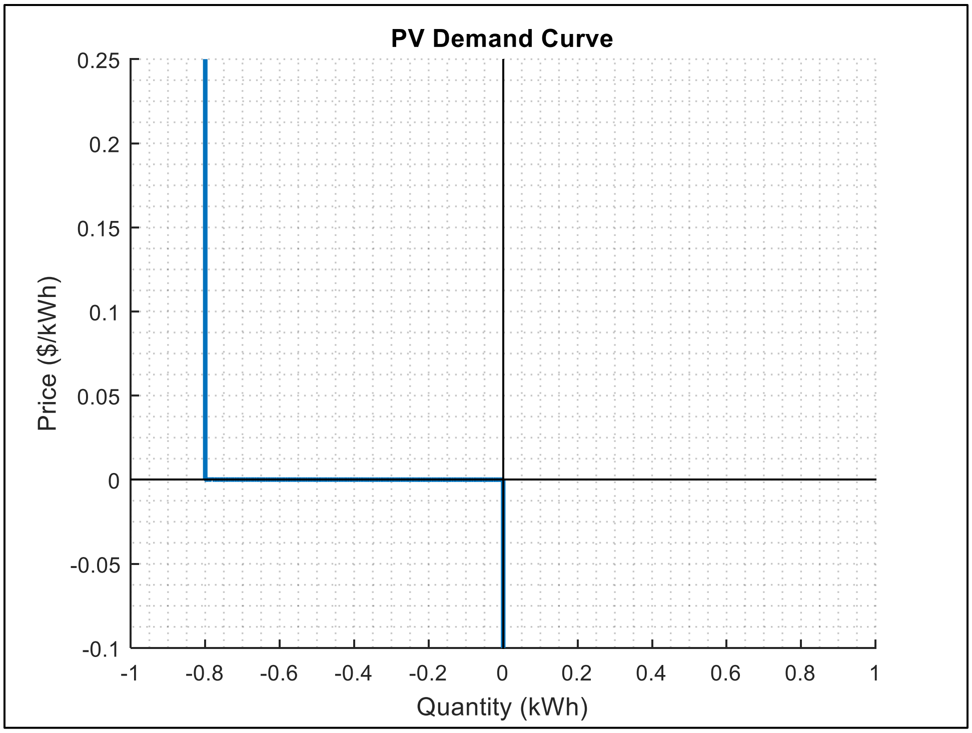


Figure 52: Example PV Node Demand Curve

The example PV Node demand curve above was generated using this method. In this example, Qest = -0.8, Qbin = 0.0001, and Pmin = 1e8.

6.3.2 Real-Time Actions

After submitting its demand curve during Market Operation, a PV Node will eventually receive a Node Marginal Price (NMP) for the upcoming market period. The NMP will be used to set the actual consumption (i.e. generation) of energy during the market period.

For this behavior model, the determination is straightforward: the PV Node will export energy for any positive price. For non-positive prices, the PV Node will curtail all generation. Note, the quantity generated does not include a prediction error, as used during demand curve generation. Thus, for any given timestep, *obj.DDS.t*, and a NMP described by *obj.priceContr*:

```
if obj.priceContr <= 0
      obj.quantActual = 0;
else
      obj.quantActual = -obj.pvDataSource(obj.DDS.tRef)/60;
end
```

Notice, this value may not match the quantity indicted on the PV Node's demand curve submission, due to the imposed Pmin restriction. In particular, if the DTDM clearing point requires a PV Node to generate at less than its full capacity, it will specify a price between the LH and RH limits. Based on the previously described LH limit determination, this would occur when the NMP meets the follow criteria:

$$0 < P_{NMP} < \frac{-Q_{est}}{Qbin} Pmin$$

In this case, the PV Node fails to meet the quantity specified by its demand curve submission. This may incur penalties, if a prediction-type tariff is imposed. However, it is assumed that avoiding this penalty is best controlled by a HEM optimization algorithm or dispatcher, not by actually adjusting the PV generation.

6.3.3 Limitations and Areas for Improvement

The PV Node behavior model has some known limitations, but perhaps less than that of the Load Node behavior model.

For one, the prediction error model may or may not reflect reality. Additional research is required to determine the ability to predicted distributed PV generation. Commercial products may exist to fill this niche. Additionally, the current model does not limit overestimating generation. This is not realistic; clearly a PV array's maximum output is known by its owner.

Second, curtailment may not be realistic, regardless of the price. This could be a limitation of control software or the owner's personal decision. This would result in a vertical demand curve, at the estimated export quantity. This is the same as an inelastic load demand curve.

Additional limitations to the PV Node behavior model may be identified with further testing and evaluation.

6.4 Storage Node Behavior Model

The Storage Node behavior model applies to bottom-level node that consists of energy storage units. The model was developed with electrochemical batteries in mind, but similar concepts apply to all energy storage devices. A Storage Node is described by the following parameters, each to be detailed: dispatch flag, control type, max discharge, max charge, energy capacity, current energy percent charge, demand curve control slope, price of max discharge, price of min discharge, price of min charge, price of max charge, exponential moving average period, and current exponential moving average value. Note, there are two methods for control, each with related behavior models. Not all parameters apply to each behavior model.

Parameters describing the storage unit's capabilities must be set during creation and initialization of the Storage Node. In simulation, these parameters are based on available commercial products, such as the Telsa Powerwall [19]. Unless denoted, the published specifications for the Tesla Powerwall will be used in all examples and case studies.

The storage unit's maximum energy capacity, *node.storageEnergyCap*, is expressed in kWh. This should reflect the possible range of energy storage. During the simulation, the current state of charge is reflected by the parameter *node.storageEnergyPct*. This is a percentage of the maximum energy capacity, and it update whenever the Storage Node imports or exports energy. Unless specified otherwise during initialization, this parameter begins the simulation at 0%.

It is assumed that the Storage Node is always able to accurately estimate its state of charge. Additionally, the current behavior model does not include any round-trip energy losses.

Additionally, the storage unit's maximum charge and discharge power are specified by *node.storageMaxCharge* and *node.storageMaxDischarge*, respectively. These parameters are expressed in kW. Note, some storage units may have normal and peak power parameters. The Node Owner may select whichever value matches their use case. Unless noted, the normal power parameter is always used in simulation, to meet the anticipated implementation.

```
DDS2.addNode('tesla powerwall node','owner','supernode');
    DDS2.recent.type = 'Storage';
    DDS2.recent.storageEnergyCap = 8;
    DDS2.recent.storageMaxDischarge = 3.3;
    DDS2.recent.storageMaxCharge = 3.3;
    DDS2.recent.storageEnergyPct = 0.5;
```

The Storage Node is initialized in a similar manner as all other bottom-level nodes. In addition to the parameters shown above, *node.storageControl* and *node.storageDispatchFlag* must also be set during initialization. These will be described in the following sections.

6.4.1 Consumption Preferences and Demand Curve Generation

The Storage Node behavior model is designed to take advantage of price arbitrage. Price arbitrage is taking advantages of price differences, in this case, at different points in time. In general, the Storage Node seeks to import energy (i.e. charge) when the price of energy is low and export energy (i.e. discharge) when the price of energy is high. In this way, a storage unit can achieve a return-on-investment.

Price arbitrage considerations drive the Storage Node's consumption preferences and demand curve generation.

The DTDM provides inherent flexibility in taking advantage in price arbitrage. The price considered by the Storage Node will naturally include opportunity costs faced by the Storage Node owner; this is a consequence of tariffs and the marginal cost convention. This is particularly the case for dispatchable Storage Nodes. Recall, Section 5.7.3 described Dispatcher Nodes that monitor real-time flow and dispatch energy storage to meet the optimal clearing quantity. A Storage Node is flagged as dispatchable with the parameter *node.storageDispatchFlag*. This does not impact the description of consumption preferences or demand curve generation in this section; the consequence of dispatchable storage will be described in the Storage Node behavior model's Real-Time Actions.

The Storage Node's preference for price arbitrage is reflected in its demand curve generation. This is accomplished in the method *node.setStorageCurve*.

First, the range of possible energy quantities are determined using the maximum charge and discharge power over the specified market duration.

```
Qstart = -obj.storageMaxDischarge*marketDuration/60;
Qstart = obj.Qbin*round(Qstart/obj.Qbin);
Qend = obj.storageMaxCharge*marketDuration/60;
Qend = obj.Qbin*round(Qend/obj.Qbin);
obj.curve(:,1) = [Qstart:obj.Qbin:Qend];
```

Notice, this provides the LH and RH limits of the curve, with the LH limit at a negative quantity (energy export) and the RH limit at a positive quantity (energy import).

Next, the demand curve sets its price points. This approach can be described as the linear threshold model. This approach needs four price values, each of which is a parameter: *priceMaxDischarge, priceMinDischarge, priceMinCharge,* and *priceMaxCharge*. The *priceMaxDischarge* set the LH limit; for any NMP at or above this price, the node will export energy at its maximum capacity. Similarly, the *priceMaxCharge* sets the RH limit; for any NMP at or below this price, the node will import energy at its maximum capacity. The *priceMinDischarge* and *priceMinCharge* indicate the prices at which the node will begin energy exports and imports, respectively. As a result, price points between *priceMinDischarge* and *priceMinCharge* all indicate zero quantity: the node neither exports or imports energy. This segment can be thought to represent the Storage Node's desired price arbitrage spread. These price points are illustrated in Figure 53.

In Figure 53, notice the slope of the curve in the energy export and import regions. Different slopes have different costs and benefits. To optimize profits, the Storage Node may seek a nearly horizontal slope; if the observed price meets the arbitrage spread, then the storage unit would never desire to export or import energy at partial capacity. Note, the demand curve submission must satisfy the Pmin restriction, so there these slopes cannot be perfectly horizontal. Alternatively, the Storage Node may seek to provide responsive energy balancing for its DTDM subsystem. In this case, a steeper slope may be desired; this would provide a larger range of price-quantity options presented to the Storage Node's supernode.

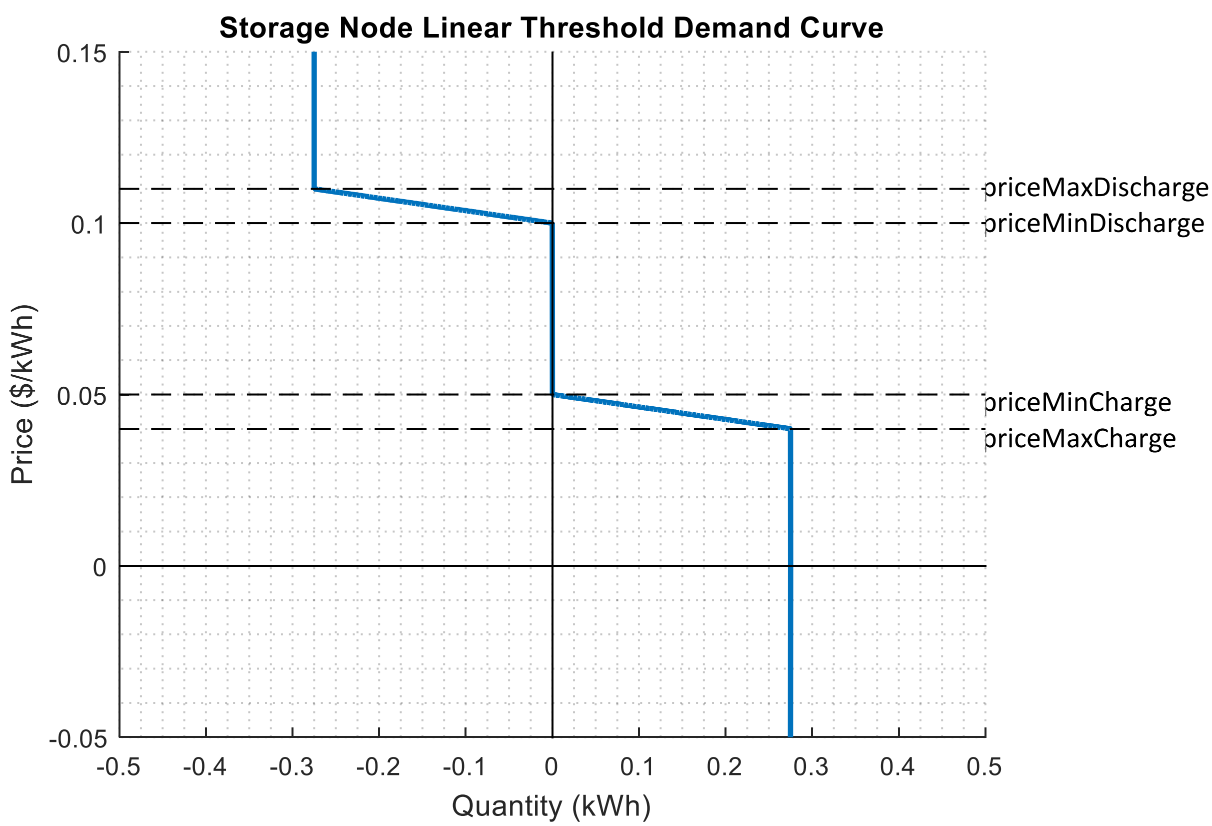


Figure 53: Storage Node Price Thresholds

There are two control methods for determining the thresholds, specified by *node.storageControl*. In 'manual', all price threshold values are specified explicitly. This must occur during initialization, but the parameters could also be directly updated during a simulation. In 'dynamic' control, price thresholds are determined dynamically, based on the exponential moving average of the NMP. Each will be described in the next step of demand curve generation: establishing the price points for each Qbin interval.

For ‘Manual’ control type nodes, the four price thresholds are set directly, establishing the demand curve slopes indirectly. As stated, the segment between *priceMinDischarge* and *priceMinCharge* is ideally vertical at Q = 0. However, this is not acceptable; only a single price can be at this Qbin interval. This is set as *priceZero*, the midpoint between *priceMinDischarge* and *priceMinCharge*. These two price points will actually be recoreded at –Qbin and +Qbin, respectively.

```
    priceZero                                                    =
(obj.storagePriceMinDischarge+obj.storagePriceMinCharge)/2;
```

Next, the discharge and charge slopes are set, using the known points:

```
            [Qstart, obj.storagePriceMaxDischarge]
            [-Qbin,  obj.storagePriceMinDischarge]
            [+Qbin,  obj.storagePriceMaxCharge   ]
            [Qend,   obj.storagePriceMinCharge   ]

    dischargeSlope = (obj.storagePriceMinDischarge-...
        obj.storagePriceMaxDischarge)/(-obj.Qbin-Qstart);
    chargeSlope = (obj.storagePriceMaxCharge-...
        obj.storagePriceMinCharge)/(Qend-obj.Qbin);
```

Because the four price threshold values were set during initialization, these slopes must be check for Pmin restriction compliance. If the Pmin restriction is met, the demand curve will be based on these values.

For Storage Nodes with control type ‘Dynamic’, the demand curve slopes are set directly, and the price thresholds are set dynamically during simulation. Specifically, the price point at Q = 0, *priceZero*, is determined by tracking the exponential moving average of the Storage Node’s observed NMP, stored in the parameter *node.storageDynEMAcurrent*. This parameter is updated after receiving the NMP in Real-Time Actions. This behavior model's "thought process" can be described as: if the price is higher than it has been, I'll export energy, and if the price is lower than it has been, I'll import energy.

Note, the 'Dynamic' control does not include an arbitrage spread. In other words, *priceMinDischarge* and *priceMinCharge* are essentially the same as *priceZero*. The only difference is ±Pmin, which is imposed to meet the DTDM Pmin restriction. Refinements to this behavior model would include a user-defined arbitrage spread, among other improvements.

The parameter *storageControlSlope* must be set during initialization for 'Dynamic' control type Storage Nodes. This determines the last component required for setting the demand curve price points. Note the previously described trade-off between profit maximization and control range when setting this parameter.

```
priceZero = obj.storageDynEMAcurrent;
obj.storagePriceMinDischarge = priceZero + obj.Pmin;
obj.storagePriceMinCharge = priceZero - obj.Pmin;
dischargeSlope = obj.storageControlSlope*(-obj.Pmin/obj.Qbin);
chargeSlope = obj.storageControlSlope*(-obj.Pmin/obj.Qbin);
```

The two processes described above establish the following variables, for both 'Manual' and 'Dynamic' control type Storage Nodes: *priceZero, storagePriceMinDischarge, dischargeSlope, storagePriceMinCharge, chargeSlope*. These variables are used to develop the demand curve price points, as follows:

```
obj.curve(:,2) = priceZero;
obj.curve(obj.curve(:,1)<0,2) = obj.storagePriceMinDischarge + ...
      (obj.curve(obj.curve(:,1)<0,1)+obj.Qbin)*dischargeSlope;
```

```
obj.curve(obj.curve(:,1)>0,2) = obj.storagePriceMinCharge + ...
      (obj.curve(obj.curve(:,1)>0,1)-obj.Qbin)*chargeSlope;
```

With the Storage Node's demand curve established, the final check is ensure the range of quantities specified can be supported by the storage units current state of charge. Any quantities that would result in a state of charge outside 0-100% are trimmed from the demand curve. This used the current storage state of charge, *node.storageEnergyPct*, which is updated at the end of Real-Time Actions, and the storage unit's capacity, *node.storageEnergyCap*.

```
availableDischarge = obj.storageEnergyCap*obj.storageEnergyPct;
availableCharge = obj.storageEnergyCap*(1-obj.storageEnergyPct);
obj.curve(obj.curve(:,1)<-availableDischarge,:) = [];
obj.curve(obj.curve(:,1)>availableCharge,:) = [];
```

This concludes the Storage Node's demand curve generation. Note, generation of a demand curve occurs for all Storage Nodes, both dispatchable and non-dispatchable.

6.4.2 Real-Time Actions

After submitting its demand curve during Market Operation, a Storage Node will eventually receive a Node Marginal Price (NMP) for the upcoming market period. The NMP will be used to set the actual consumption of energy (whether importing or exporting) during the market period.

First, recall the role of dispatchable nodes in Section 5.7.3. A dispatchable node generates a demand curve, but its energy consumption is directly by its Dispatcher Node. A dispatchable storage node is indicated by the parameter *node.storageDispatchFlag*. If this parameter is true, the Storage Node will not set its own energy consumption during Real-Time Action; consumption will be determined by the Dispatcher Node's fast-loop process.

```
        case {'storage'}
              if obj.storageDispatchFlag
                    obj.quantActual = 0;
              else
                    obj.setStorageQuantActual;
              end
```

If not a dispatchable storage node, the Real-Time energy consumption is set in the method *node.setStorageQuantActual*. In this method, the Storage Node compares the NMP to its demand curve parameters, using linear interpolation as necessary. This is performed by a series of `elseif` statements, moving left to right on the demand curve. This process sets the parameter *node.quantActual* for the timestep.

```
  if obj.priceContr >= obj.storagePriceMaxDischarge
  % maximum discharge
        obj.quantActual = -obj.storageMaxDischarge/60;
  elseif obj.priceContr >= obj.storagePriceMinDischarge
  % limited discharge
        Qstart = -obj.storageMaxDischarge/60;
        obj.quantActual =  Qstart + (obj.priceContr-...
              obj.storagePriceMaxDischarge)*(-obj.Qbin-Qstart)/...
             (obj.storagePriceMinDischarge-
        obj.storagePriceMaxDischarge);
  elseif obj.priceContr > obj.storagePriceMinCharge
  % no charge or discharge
        obj.quantActual = 0;
  elseif obj.priceContr > obj.storagePriceMaxCharge
  % limited charge
        Qend = obj.storageMaxCharge/60;
        obj.quantActual = obj.Qbin + (obj.priceContr-...
              obj.storagePriceMinCharge)*(Qend-obj.Qbin)/...
              (obj.storagePriceMaxCharge-obj.storagePriceMinCharge);
  else
```

```
% maximum charge
      obj.quantActual = obj.storageMaxCharge/60;
end
```

Next, *node.quantActual* is checked against the current state of charge and discharge and charge power limits, to verify the storage unit parameters are not exceeded.

```
obj.quantActual = max(-obj.storageMaxDischarge/60,obj.quantActual);
obj.quantActual = min(obj.storageMaxCharge/60,obj.quantActual);
availableDischarge = obj.storageEnergyCap*obj.storageEnergyPct;
availableCharge = obj.storageEnergyCap*(1-obj.storageEnergyPct);
obj.quantActual = max(-availableDischarge,obj.quantActual);
obj.quantActual = min(availableCharge,obj.quantActual);
```

This finalizes *node.quantActual*. Next, the storage unit's state of charge is updated.

```
      obj.storageEnergyPct = obj.storageEnergyPct + ...
            obj.quantActual/obj.storageEnergyCap;
```

Finally, if the Storage Node has control type 'Dynamic', the NMP exponential moving average (EMA) must be updated. This is controlled by the EMA period, *node.storageDynEMAperiod*, which must be set during initialization. The standard alpha form of EMA is assumed. For a detailed description of EMA calculation, see Section 6.8 Ramping Tariff.

```
if strcmpi(obj.storageControl,'dynamic')
      if obj.DDS.t == 1
      % for first iteration, "pad" with first provided price point
            obj.storageDynEMAcurrent = obj.priceContr;
      end
      alpha = 2/(obj.storageDynEMAperiod+1);   % Standard alpha form
      deltaEMA = alpha*(obj.priceContr-obj.storageDynEMAcurrent);
```

```
        obj.storageDynEMAcurrent = obj.storageDynEMAcurrent +
    deltaEMA;
    end
```

This concludes the Real-Time Actions for the Storage Node behavior model.

6.4.3 Limitations and Areas for Improvement

The Storage Node behavior models has multiple known limitations and areas for improvement. These fall into two categories: storage unit representation and profit maximization algorithms.

The current behavior model does not include any limits or targets for cycling rates. For example, the 7 kWh Tesla Powerwall is designed for daily cycling [19]. The behavior model should incorporate this as a target cycling rate. If the price thresholds result in cycling more than one time a day, the storage unit will reach its end-of-life earlier than expected; the arbitrage spread should be increased. If the price thresholds result in cycling less than one time a day, the storage unit is not taking advantage of enough small price arbitrage opportunities; the arbitrage spread should be decreased. By incorporating a cycling target, these adjustments could be accomplished autonomously and dynamically.

Additionally, this behavior model does not include round-trip efficiency losses; all energy imported is available as energy to be exported. This is not realistic. The Tesla Powerwall specifications indicate 92% round-trip DC efficiency [19]. Efficiency losses could be implemented in multiple ways. One possible implementation is to reduce the state of charge parameter, *node.storageEnergyPct*, by a fixed value every timestep. This would simulate losses over time. Alternatively, or in addition, losses could be imposed during the import and export processes. For a given efficiency, α, this could be represented by:

$$Q_{\Delta Storage} = \alpha Q_{actualDemand}$$

This behavior model also assumes the state of charge is known with accuracy, at any point in time. This may or may not be accurate. Further analysis of this assumption, and incorporation into the behavior model, is warranted.

Finally, this behavior model does not include peaking power capabilities, only steady-state power capabilities. Many storage units have the option to import or export at increased power levels for short durations. Determining when and why a storage unit should utilize this feature is an opportunity for further analysis.

The second category of limitation is the profit maximization algorithm. This is an area for robust improvements, so only general considerations are listed.

As previously stated, the 'Dynamic' control mode does not include an arbitrage spread. This would be a useful addition to the current behavior model.

Neither control mode includes a prediction of future prices. An effective profit-maximization control algorithm would certainly include this component. In particular, the prediction of future prices would represent an anticipated opportunity cost, especially when coupled with storage cycling targets and limitations. Development of this algorithm provides ample opportunity, but this requires establishing a known set of DTDM rules and analyzing the characteristics of a specific DTDM network implementation.

This concludes the last node behavior model. Load, PV, and Storage Nodes are all implemented in the DTDM simulation. Next, the proposed tariffs designs are described.

6.5 Tariff Design Overview

As described in Section 3.5.3, a tariff is defined by its *structure*, which is established prior to DTDM operation, and *parameters*, with are set dynamically during Market Operation. A *tariff instance* is a tariff structure with specified parameters. The tariff instance is used to generate a *tariff settlement function* and *tariff curve*. The following sections highlight the process for implementing tariffs in the Matlab simulation, as well specific tariffs designs.

Note, the DTDM process specifies a tariffs design moving from *structure* to *parameters* to *instance* to *settlement function* to *tariff curve*. However, implementation in the simulation may or may not follow this process. This is for two, non-mutually exclusive reasons: it has been done for convenience, where the tariff curve is generated directly; and it is a consequence of the iterative process of simulation and DTDM rule development.

As a result, implementation of tariff design into the DTDM simulation is an area for further development. In particular, improvements can be made in defining and limiting the communication between nodes owners and tariff owners.

Regardless, the general relationship between the revenue collected by the *tariff settlement function* and *tariff curve* is as follows:

$$Revenue(Q) = \int_{o}^{Q} P_{tariff}(Q)\, dQ + Revenue_{Q=0}$$

$$P_{tariff}(Q) = \frac{d}{dQ} Revenue(Q)$$

As a result, a *tariff curve* can fully describe a *tariff settlement function*, when the revenue for Q = 0 (i.e. the offset) is known. As previously stated, tariff curves are considered supply curves and must be non-descending, although they are not required to be strictly ascending.

During tariff creation, the tariff's location (i.e. subnode), owner, and type are specified. The current simulation only provides for tariff types: capacity, ramping simplified moving average, ramping exponential moving average, and flat-rate. In addition, this section will provide for the general construction of a target quantity tariff, which would be used for prediction and volatility tariffs.

During simulation Market Operation, the tariff object is called to directly generate its tariff curve in the method *tariff.generateCurve*. This is based on the tariff type and parameters. During simulation Real-Time Actions, tariff objects are not called; their curves and parameters have already been presented to actors in the DTDM network. During simulation Settlement, the tariff objects establish revenue collection with the method *tariff.settle*. This method is used by the subnode, to update its *node.meterTariffPayment* array, and by the tariff, to update its *tariff.meterRevenue* array.

For each tariff design, the following will be described: goals, tariff curve generation, settlement, and limitations.

6.6 Capacity Tariff

A capacity tariff seeks the capture the externality imposed physical system limitations. Specifically, it attempts to disincentivize energy quantities that exceed component ratings, such as transformers or feeders. In addition to providing a price signal disincentive to subnodes, revenues collected by a capacity tariffs could be used to fund component upgrades and replacements. Capacity tariffs provide an opportunity to replace demand charges in a traditional rate structure.

A capacity tariff is placed on the DTDM network linkage with the physical system limitation.

6.6.1 Goals and Tariff Curve Creation

A capacity tariff, and its tariff curve, is characterized the component power rating, the acceptable direction of energy for, the energy quantity at which the tariff takes effect, and the shape of the tariff curve. It is assumed that the capacity tariff LH and RH limits are at the component energy limits. Like any supply curve, it is assumed that P = -Pcap for all quantities less than the LH limit and P = Pcap for all quantities more than the RH limit.

It is assumed capacity tariffs do not collect any revenue for Q = 0, so the translation between the revenue function and the tariff curve is simplified.

During tariff creation, the component power rating must be specified, in kW. This is stored in the parameter *tariff.capPwrLimit*. Additionally, tariff must specify the acceptable direction of flow. This is stored in the parameter *tariff.capFlow*. Acceptable values are 'bi' for bidirectional flow (the default), 'pos' for positive flow (energy imports only) and 'neg' for negative flow (energy exports only). Bidirectional flow assumes the same power rating applies in both directions.

A capacity tariff generates its tariff curve with the method *tariff.setCapCurve*, which is called in *tariff.generateCurve*.

First, the component power rating is converted to an energy limitation, based on the market duration. This is stored in the parameter *tariff.capQlimit*.

```
obj.capQlimit = obj.capPwrLimit*(marketDuration/60);
```

Note, this values is rounded to the nearest Qbin interval.

Next, the acceptable flow direction determines the quantity values of the tariff curve.

```
limitNeg = -obj.capQlimit;  % default for bi flow
limitPos = obj.capQlimit;   % default for bi flow
if strcmpi(obj.capFlow,'pos')
    limitNeg = 0;           % adjust for pos flow
elseif strcmpi(obj.capFlow,'neg')
    limitPos = 0;           % adjust for neg flow
end
obj.curve = [];     % clear existing curve
obj.curve(:,1) = limitNeg:obj.Qbin:limitPos;
```

The tariff curve prices values are determined by the capacity tariff curve shape, which is dictated by the parameter *tariff.capRampType*. This value must be initialized to one of the following options: 'flat', 'linear', or 'quadratic'.

'Flat' ramp type capacity curves do not provide any incentives, other than the LH and RH limits. This type of capacity curve does not influence the DTDM behavior, other than preventing quantities beyond the component capacities. Thus, the tariff curve is P = 0 for all quantities.

```
case {'flat'}
      obj.curve(:,2) = 0;
```

'Linear' and 'quadratic' ramp type capacity curves seeks to disincentivize quantities near, but not at the LH and RH limits. This is accomplished by "ramping" the tariff curve as it approaches the LH and RH limits. A 'linear' ramp type does this with a linear function; a 'quadratic' ramp type does this with a quadratic function. Of these two options, the 'quadratic' ramp type provides stronger price signals as the capacity limit is approach; this is useful for limiting the burden on the DTDM subsystem.

Both these ramp types must specify the quantity at which the ramp begins. This is stored as the parameter *tariff.capQrampPct*, which is a percentage of the maximum energy capacity. During curve generation, this applies as follows, for both ramp types:

```
QrampStart = obj.capQlimit*obj.capQrampPct;
```

The other relevant parameter is *tariff.capPpeak*, which specifies the price value at the LH and RH limit of the tariff curve. The larger this value, the steeper the tariff curve for quantity values beyond *QrampStart*.

These parameters should be selected by the system stability and financial goes of the tariff owner. For example, if the tariff subnodes fail to predict their preferences accurately, a low *tariffQrampPct* may be desired. Alternatively, if the tariff owner seeks to collect revenues for infrastructure upgrades, a large *tariff.capPpeak* may be desired.

Next, these values are used to develop the price points on the tariff curve.

For a 'linear' ramp type, the following equation provides the tariff curve:

$$P_{tariff} = sign(Q)\big(abs(Q) - Q_{ramp}\big)\left(\frac{P_{peak}}{Q_{limit} - Q_{ramp}}\right), \qquad abs(Q) \geq Q_{ramp}$$

Which is implemented in Matlab as follows:

```
pricesPos = (obj.curve>QrampStart).*(obj.curve-QrampStart)*...
      obj.capPpeak/(limitPos-QrampStart); % positive quantities
pricesNeg = (obj.curve<-QrampStart).*(obj.curve+QrampStart)*...
      -obj.capPpeak/(limitNeg+QrampStart); % negative quantities
obj.curve(:,2) = pricesPos + pricesNeg;
```

For a 'quadratic' ramp type, the following equation provides the tariff curve:

$$P_{tariff} = sign(Q)\left(abs(Q) - Q_{ramp}\right)^2 \frac{P_{peak}}{\left(Q_{limit} - Q_{ramp}\right)^2}, \qquad abs(Q) \geq Q_{ramp}$$

Which is implemented in Matlab as follows:

```
pricesPos = (obj.curve>QrampStart).*(obj.curve-QrampStart).^2*...
      obj.capPpeak/(limitPos-QrampStart)^2; % positive quantities
pricesNeg = (obj.curve<-QrampStart).*(obj.curve+QrampStart).^2*...
      -obj.capPpeak/(limitNeg+QrampStart)^2; % negative quantities
obj.curve(:,2) = pricesPos + pricesNeg;
```

Note, unidirectional flow tariffs do not seek to disincentivize quantities near, but not at, zero energy flow.

Illustrations of the tariff curves are provided in Figure 54. For all plots, *tariff.capQlimit* = 0.5 kWh. For 'linear' and 'quadratic' ramp type, *tariff.capQrampPct* = 0.8 and *tariff.capPeak* = $0.05/kWh.

This concludes the generation of the capacity tariff curves. Next, settlement is described.

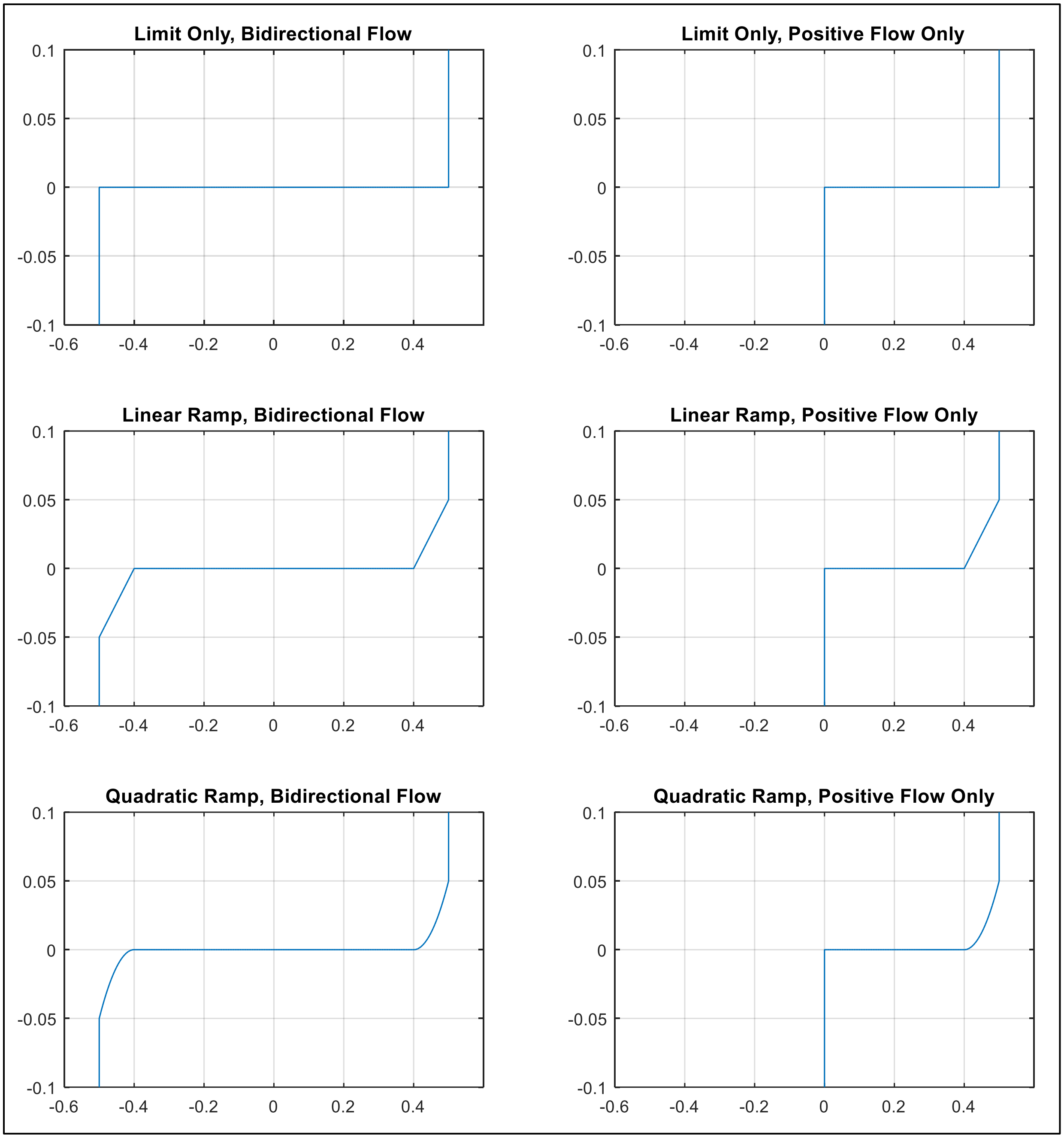


Figure 54 Capacity Tariff Curve Examples

6.6.2 Settlement

Settlement is performed, for all tariffs, in the method *tariff.settle*. This method checks the tariff type, parameters, and the subnode *quantActual*, then updates the parameter *tariff.revenue*. This parameter is the revenue collected by the tariff from its subnode, for the current timestep; this value is used to update tariff and subnode meter arrays.

```
quant = obj.sub.quantActual;
```

First, the method checks to see if the component limits were exceeded in the timestep. If so, the tariff curve is assessed tariffs based on Pcap.

```
if (strcmpi(obj.capFlow,'neg')&&quant>0)
      obj.revenue = quant*obj.DDS.Pcap;
elseif (strcmpi(obj.capFlow,'pos')&&quant<0)
      obj.revenue = quant*-obj.DDS.Pcap;
elseif (quant>limitQ)||
% if exceeds maximum flow, use Pcap for the difference
      obj.revenue = (quant-limitQ)*obj.DDS.Pcap;
elseif (quant<-limitQ)||
% if exceeds minimum flow, use -Pcap
obj.revenue = (quant+limitQ)*-obj.DDS.Pcap;
```

If these conditions are not met, the ramp type is considered. For 'flat' type tariffs, there is no revenue between the LH and RH limits.

```
elseif strcmpi(obj.capRampType,'flat')
      obj.revenue = 0;
```

For 'linear' and 'quadratic' ramp type tariffs, the quantity is compared to *tariff.capQrampPct*. For quantities exceeding *tariff.capQrampPct,* settlement determined by taking the integral of the tariff curve, recognizing there is no "offset" payment.

$$Revenue(Q) = \int_{o}^{Q} P_{tariff}(Q)\,dQ + Revenue_{Q=0}$$

$$P_{tariff,linear}(Q) = sign(Q)\left(\frac{P_{peak}}{limitQ - rampStart}\right)(|Q| - rampStart), \qquad |Q| \geq rampStart$$

$$Revenue_{linear}(Q) = \left(\frac{P_{peak}}{2(limitQ - rampStart)}\right)(|Q| - rampStart)^2, \qquad |Q| \geq rampStart$$

$$P_{tariff,quadr}(Q) = sign(Q)\left(\frac{P_{peak}}{(limitQ - rampStart)^2}\right)(|Q| - rampStart)^2,$$

$$|Q| \geq rampStart$$

$$Revenue_{quadr}(Q) = \left(\frac{P_{peak}}{3(limitQ - rampStart)^2}\right)(|Q| - rampStart)^3, \qquad |Q| \geq rampStart$$

Which, are implemented in Matlab as follows:

```
else
      rampStart = limitQ*obj.capQrampPct;
      if (abs(quant)<rampStart)
            obj.revenue = 0;
      else
            switch lower(obj.capRampType)
            case {'linear'}
                  obj.revenue = (obj.capPeak/(2*(limitQ-...
                        rampStart)))*(abs(quant)-rampStart)^2;
            case {'quadratic'}
                  obj.revenue = (obj.capPeak/(3*(limitQ-...
                        rampStart)^2))*(abs(quant)-rampStart)^3;
            end
      end
end
```

This concludes the settlement for capacity tariffs, as current implemented in the DTDM simulation.

6.6.3 Limitations and Areas for Improvement

This tariff proposal is not without limitations or areas for possible improvement.

One, the settlement process indicates the most significant limitation of this proposed tariff design: settlement occurs for each timestep but the tariff curve applied to the market duration, which may have been more than one timestep. Consider a five-minute market duration. A load may exceed a component rating for one minute of the market duration, which keeping its average load within the component rating. This possibility is not accurately controlled by the DTDM and proposed tariff design.

In addition, with settlement every timestep, the provided tariff curve does not necessarily represent the subnode's interpretation of the tariff instance. The subnode's interpretation of the tariff's impact should include the minute-by-minute impact of the tariff instance on its consumption. This adds a layer of complexity not included in the current DTDM simulation.

As a starting point to resolving these limitations, the DTDM simulation should be follow the tariff process outlined in Section 3.5.3. Specifically, the tariff owner should specify the tariff structure and parameters only. Then a subnode owner behavior model should interpret the tariff instance into a tariff curve, for use in demand curve adjustment. This places hedging and risk mitigation solely upon the subnode owner. This observation applies to all proposed tariff designs; more comments on simulation improvement are listed in Section 8.3.

6.7 General Target Quantity Tariff

A target quantity tariff seeks to incentivize the subnode consumption at a target quantity. This general tariff design is described because it has many applications. For example, a volatility tariff could be implemented to incentivize stable minute-by-minute energy consumption; this would use the previous energy consumption as a target quantity. Alternatively, a prediction tariff could be implemented to incentivize consuming the quantity specified by the NMP of the subnode's demand curve submission; this would use that quantity as the target quantity.

The general target quantity tariff is not applied in the current DTDM simulation.

Unlike the capacity tariff, it cannot be assumed that a target quantity tariff does not collect revenue for Q = 0. In fact, this "offset" parameter provides flexibility to the tariff owner.

A simple target quantity tariff can be described by four variables: targetQ, priceAbove, priceBelow, and revenueOffset. The tariff design is simple:

$$Revenue(Q) = P_{above}\left(Q - Q_{target}\right) + Revenue_{offet}, \qquad Q > Q_{target}$$

$$Revenue(Q) = Revenue_{offset}, \qquad Q = Q_{target}$$

$$Revenue(Q) = P_{below}\left(Q_{target} - Q\right) + Revenue_{offset}, \qquad Q < Q_{target}$$

Returning to the general revenue to tariff curve conversion described in Section 6.5, it is useful to use targetQ, rather than Q = 0, as the reference point.

$$Revenue(Q) = \int_{targetQ}^{Q} P_{tariff}(Q)\, dQ + Revenue_{Q=targetQ}$$

From this equation, the tariff curve can be developed:

$$P_{tariff}(Q) = \frac{d}{dQ} Revenue(Q)$$

$$P_{tariff}(Q) = P_{above}, \qquad Q > Q_{target}$$

$$P_{tariff}(Q) = 0, \qquad Q = Q_{target}$$

$$P_{tariff}(Q) = P_{below}, \qquad Q < Q_{target}$$

Notice, there are no quantity limits inherent to the general target quantity tariff. It is expected that any component capacity limits would be implemented in a separate capacity tariff.

Figure 55 illustrates three related general target quantity tariff instances.

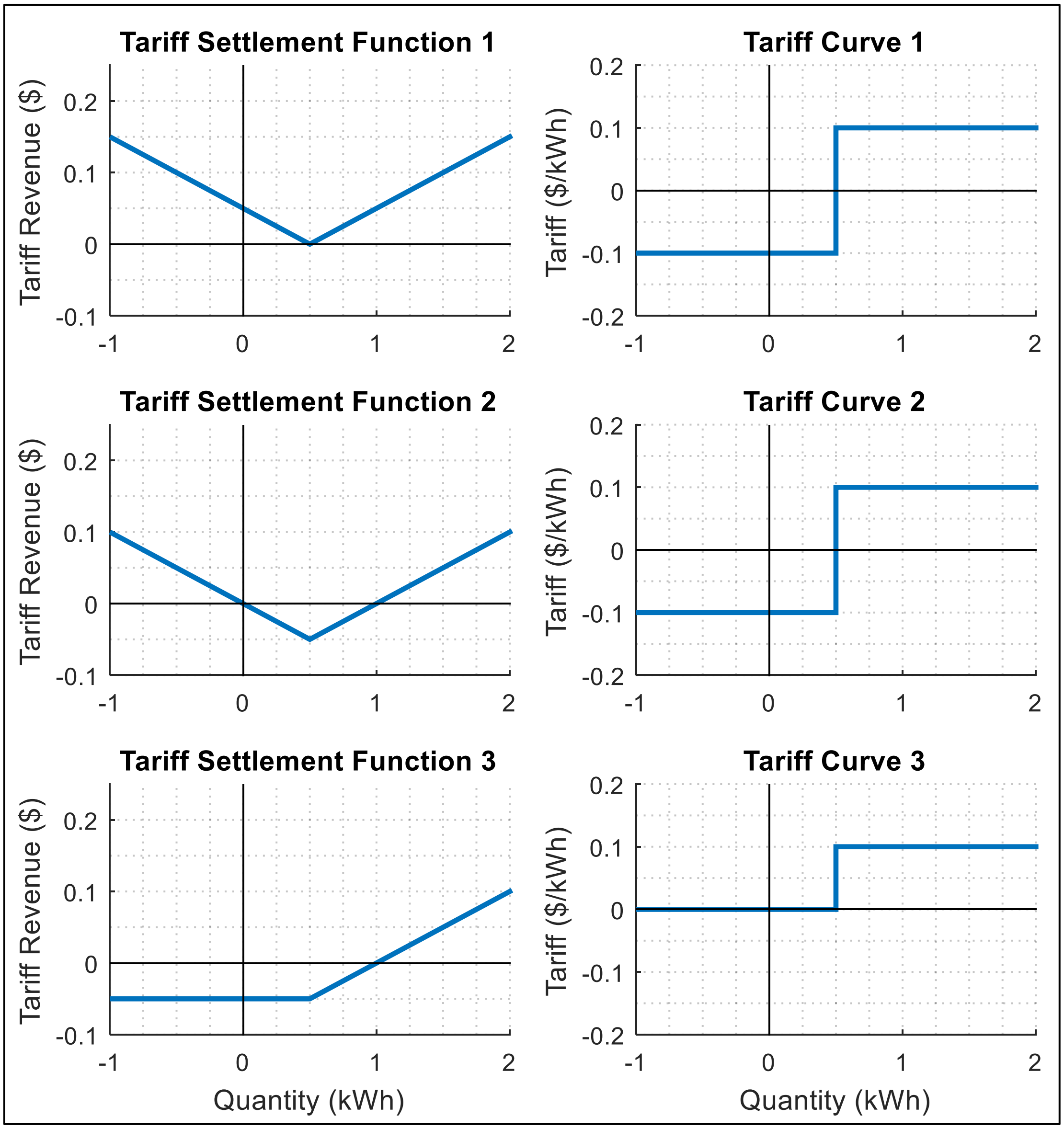


Figure 55: Example Target Quantity Tariffs

All tariff instances use targetQ = 0.5 kWh.

Instance 1 uses priceAbove = \$0.10/kWh, priceBelow = \$0.10/kWh, and revenueOffset = \$0.

Instance 2 uses priceAbove = \$0.10/kWh, priceBelow = \$0.10/kWh, and revenueOffset = \$-0.50.

Instance 3 uses priceAbove = \$0.10/kWh, priceBelow = \$0/kWh, and revenueOffset = \$-0.50.

Notice Tariff Instances 1 and 2 provide the same tariff curves (i.e. incentives) to their subnodes, despite different revenues. This provides an opportunity to secure customer support and mitigate tariff over-collection. In fact, notice Tariff Instance 2 actually makes payments to the subnode, if the actual quantity is within ±0.5 kWh of the target quantity.

Additionally, Figure 55 illustrates an inherent limitation to the target quantity tariff parameters. The tariff curve is required to be non-descending. This requires a convex tariff settlement function. Using the parameter definitions presented, this requirement provides the following restriction:

$$P_{above} \geq -P_{fall}$$

Alternative tariff designs may choose to use more complex settlement functions and tariff curves. This is acceptable, insomuch as the tariff curve is non-descending and the structure and parameters are communicated to the subnode. For example, a target quantity tariff may elect to use a quadratic revenue function; this would require additional parameters and would results in a linearly sloped tariff function.

The observations made for a general target quantity tariffs are useful when concepts to other, related tariffs. These observations apply for the proposed Ramping Tariffs, for both Simplified and Exponential Moving Average methods.

6.8 Ramping Tariff: Simplified Moving Average

A ramping tariff seeks the capture the externality of increasing (or decreasing) energy consumption over time. The proposed tariff design uses a modification of the general target quantity tariff, with the Simplified Moving Average (SMA) of energy consumption as the basis of the target quantity.

Alternatively, Section 6.8 will proposed a tariff design using the Exponential Moving Average of energy consumption as the basis of the target quantity.

6.8.1 Goals and Tariff Curve Creation

A ramping tariff is characterized by *tariff.rampWindow*, which is the duration from which the SMA is derived, and the incentives placed on positive and negative ramping: *rampPriceRise* and *rampPriceFall*, respectively. Ramping is defined in terms of average power per minute, not energy per minute. Positive ramping is when the SMA of average power consumption increases, based on the average power of the most recent timestep. Negative ramping is when the SMA of average power consumption falls.

The parameter *rampWindow* determines how many historical quantities are included in the SMA. As a starting point, this value could reflect the denominator in a ramping reserve ancillary market or system requirement.

For example, the Advanced Research Project Agency – Energy (ARPA-E) Funding Opportunity Announcement (FOA) for Network Optimized Distributed Energy Systems specifies a category specifically for “synthetic ramping reserves” [20]. The performance criteria for synthetic ramping reserves include a response time of less than 10 minutes and ramp time of less than 30 minutes. A dynamic tariff configured to meet this requirement would need a common ramping metric denominator; this would likely result in *rampWindow* between 10 and 30 minutes. However, more analysis is warranted in selecting this parameter value.

Although the ramping tariff uses average power instead of energy, the incentive parameter limitations described for general target quantity tariffs still apply. The following inequality must be satisfied, to provide a valid tariff curve:

```
obj.rampPriceRise >= -obj.rampPriceFall
```

Like the capacity tariff, the DTDM simulation develops this tariff curve directly based on the tariff parameters. This is determined in the method *tariff.setRampCurve*, which is called in *tariff.generateCurve*.

The general equation for a Simplified Moving Average comes from financial applications. It is the unweighted mean of *n* points, extending back in time. If M is the current value, the SMA for *n* points is:

$$SMA = \frac{x_{M-(n-1)} + x_{M-(n-2)} + \cdots + x_{M-1} + x_M}{n}$$

Substituting tariff parameters, SMA reference power is established by the following

$$previousAvgKW = 60 * \frac{sum\big(Q(t - rampWindow{:}\, t - 1)\big)}{rampWindow}$$

And the impact of a new, arbitrary quantity value, *newQ*, on the SMA is

$$newAvgKW = 60 * \frac{sum\big(Q(t - rampWindows + 1{:}\, t - 1)\big) + newQ}{rampWindow}$$

Thus, the change in average power SMA can be simplified to the following

$$\Delta AvgKW = newAvgKW - previousAvgKW = 60 * \frac{newQ - Q(t - rampWindow)}{rampWindow}$$

Notice, the impact on the average power SMA is determined solely by the new quantity, *rampWindow*, and the quantity set to "drop out" of the SMA. Thus, the ramping condition can be described as follows, as a function of the new energy quantity.

$$newQ > Q(t - tScope), \qquad \Delta AvgKW > 0, \qquad rise$$

$$newQ < Q(t - tScope), \qquad \Delta AvgKW < 0, \qquad fall$$

Note, the simulation determined previous quantities by referencing *tariff.sub.meterQuantActual*. For simulation timesteps *t*, the SMA cannot be determined until *t* > *tariff.rampWindow*. For these initial timesteps, the SMA is determined by "padding" the array with the initial recorded quantity. This approach is straightforward, but clearly has limitations. Specifically, too much weight is placed on the initial timestep quantity. To work around these limitations, the simulation should be allowed to run for many timesteps before the period of interest.

These equations are used to develop the tariff curve directly.

First, the quantity values must be set. The method *tariff.setRampCurve* recognizes that the tariff curve is infinitely horizontal. To limit the size of the array, the size of the subnode's demand curve is assessed. As long as the tariff curve exceeds the demand curve, it has the same effect during curve adjustment. Notice, this requires the tariff curve generation to always occur after the subnode curve generation. This is the order of events in the DTDM simulation.

```
startQ = obj.sub.curve(1,1)-obj.Qbin;
```

```
endQ = obj.sub.curve(end,1)+obj.Qbin;
obj.curve = [];
obj.curve(:,1) = [startQ:obj.Qbin:endQ];
```

Next, the reference quantity, *refQ*, is established by determining the recorded quantity that will "drop off" from the SMA calculation. However, if the market duration is greater than one timestep, multiple quantities will "drop off" the SMA calculation. The sum of these values is used, because it will be compared to the sum of considered values added to the SMA calculation. Considering padding for early simulation timesteps, this is implemented as:

```
if obj.DDS.t <= obj.rampWindow
      padQ1count = obj.rampWindow - obj.DDS.t + 1;
      refQ = max(marketDuration,padQ1count)*obj.sub.meterQuantActual(1)...
            +               sum(obj.sub.meterQuantActual(1:(marketDuration-
      padQ1count)));
else
      refQ = sum(obj.sub.meterQuantActual(obj.DDS.t-...
            obj.rampWindow:obj.DDS.t-obj.rampWindow+marketDuration-1));
end
```

Finally, *refQ* is used to determine the price points along the tariff curve. The parameters *rampPriceFall* and *rampPriceRise* are in terms of average power (kW) per minute, so they are translated to a marginal cost per kWh for the tariff curve.

```
lowerP = -obj.rampPriceFall*60*marketDuration/obj.rampWindow;
upperP = obj.rampPriceRise*60*marketDuration/obj.rampWindow;
obj.curve(:,2) = lowerP*(obj.curve(:,1)<refQ) +
upperP*(obj.curve(:,1)>refQ);
```

The resulting tariff curve has the same construction as the general target quantity tariff. The only difference is the source of the target quantity and the units of the pricing parameters.

6.8.2 Settlement

Settlement for the SMA ramping tariff is straight-forward and is based on the general target quantity tariff concept.

First, the change in average power is determined, using the most recent energy quantity and the "dropped off" energy quantity. As during tariff curve generation, the initial record quantity is used for padding, as necessary. Settlement occurs for every timestep, so *marketDuration* does not factor into this equation. For a timestep's given consumption, *quant*:

```
if obj.DDS.t == 1
      deltaKW = 0;
elseif obj.DDS.t <= obj.rampWindow
      deltaKW = (quant - obj.meterQuantActual(1))*(60/obj.rampWindow);
else
      deltaKW = (quant - obj.meterQuantActual(obj.DDS.t-...
            obj.rampWindow))*(60/obj.rampWindow);
end
```

Next, revenue is determined at *rampPriceRise* or *rampPriceFall*, based on the sign of *deltaKW*.

```
if deltaKW > 0
      obj.revenue = deltaKW*obj.rampPriceRise;
elseif deltaKW < 0
      obj.revenue = abs(deltaKW)*obj.rampPriceFall;
else
```

```
        obj.revenue = 0;
    end
```

Some observations can be made from this settlement method. One, as long as the relative restrictions on *rampPriceRise* and *rampPriceFall* are met, the tariff revenue can be negative. This could be a payment to the subnode for successfully providing negative ramping.

However, this design does not include the revenue offset described in the general target quantity tariff design. This does not impact the tariff curve and, thus, does not impact the DTDM energy balance. Yet, the tariff owner has reduced flexibility for determining the correct revenue balance. The revenue offset parameter is not inherently precluded from this tariff design; it should be added as desired by the tariff owner.

6.8.3 Limitations and Areas for Improvement

In addition to the lack of a revenue offset parameter, there are some fundamental limitations to the SMA ramping tariff.

One, the padding for reference values in the initial simulation timesteps is crude and inaccurate. An improved tariff design would incorporate a better approximation for previous energy quantities.

Two, like the capacity tariff design, there is an inherent conflict between one-minute settlement and longer market durations. The provided tariff curve does not necessarily represent the subnode's interpretation of the tariff instance. This should be resolved retricting the DTDM simulation to the tariff process outlined in Section 3.5.3. Specifically, the tariff owner should specify the tariff structure and parameters only. Then a subnode owner behavior model should interpret the tariff instance into a tariff curve, for use in demand curve adjustment.

Three, this tariff design inherently promotes volatility between timesteps. This is the design's most significant disadvantage. Notice, by using SMA, the change in average power only considers one reference quantity: the quantity "dropping off". In the next timestep, a new quantity is considered as a reference quantity. As a result, volatile energy flow (i.e. jumps in energy consumption between adjacent timesteps) is incentivized to propagate across timesteps.

This effect is illustrated in Figure 56 below. The energy consumption of multiple loads is illustrated over time. The aggregated load is represented by "ps26 Market", which is subject to a SMA Ramp Tariff.

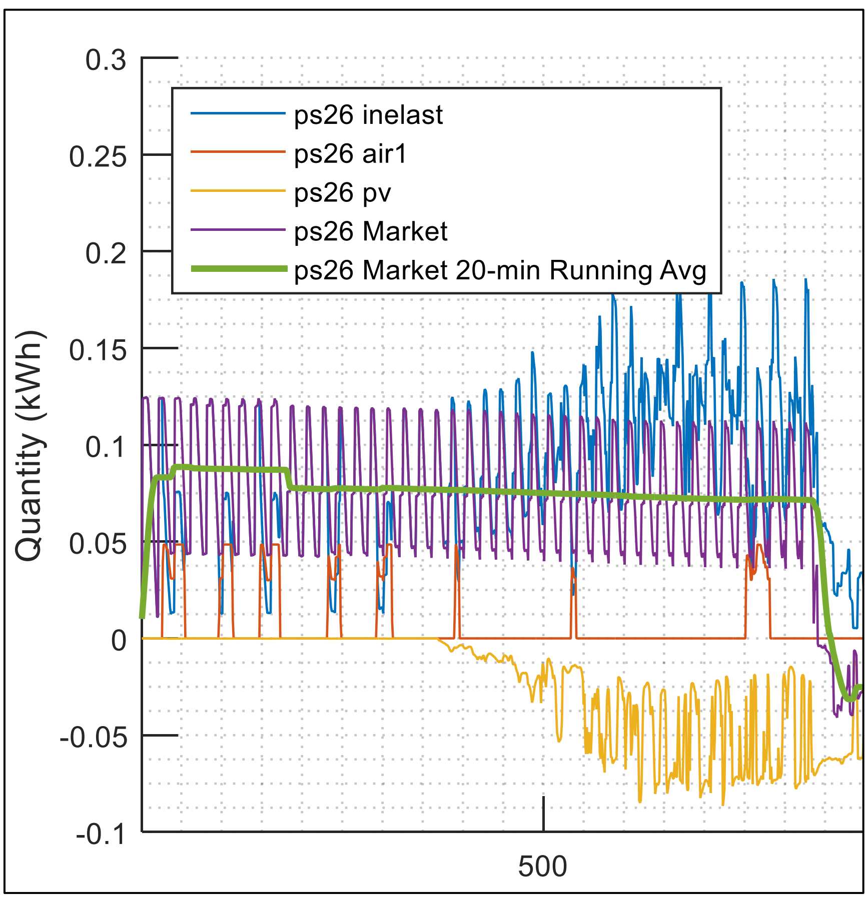


Figure 56: SMA Ramp Cycling

As can be seen, the running average (i.e. SMA) of “ps26 Market” is smooth. However, the actual minute-by-minute energy quantities is extremely volatile, with a predictable cycle. This is the market incentivizing consumption to match the quantity about to “drop off” at every timestep. This cycle is not broken until there is a significant change in the basic consumption preferences in the system.

There are multiple ways to implement the SMA ramping tariff, while mitigating this effect. One approach would be to introduce a volatility tariff, using the previous timestep’s quantity as the target quantity. This would provide damping over time. However, this adds another layer of complexity to the system.

Alternatively, the ramping tariff could be redesigned to discount previous energy quantities as they recede into the distance. This is the approach taken by the proposed Exponential Moving Average Ramping Tariff.

6.9 Ramping Tariff: Exponential Moving Average

Like the ramping tariff previously described, this tariff design seeks the capture the externality of increasing (or decreasing) energy consumption over time. The proposed tariff design uses a modification of the general target quantity tariff, with the Exponential Moving Average (SMA) of energy consumption as the basis of the target quantity.

6.9.1 Goals and EMA Technical Basis

Like the SMA ramping tariff, the EMA ramping tariff includes a parameter for the moving average period, *tariff.rampEMAnPeriod*, and two price incentives: *tariff.rampEMApriceRise* and *tariff.rampEMApriceFall*. Additionally, the tariff stores the running EMA value in the parameter *tariff.rampEMAcurrent*.

As before, the following parameter relationship must be satisfied:

```
obj.rampEMApriceRise >= -obj.rampEMApriceFall
```

The definition of the EMA differs from the SMA. The EMA is a weighted moving average. Generally, a weighted moving average gives different weights to different data points within the series. Specifically, EMA decreases the weight of data points exponentially as they move further into the past. An EMA is described with the coefficient α, which describes the degree of weighting decrease. Larger α values discount older values more quickly. α must be between 0 and 1.

For a data series *Q*, the exponential moving average for time *t* is described as [21]:

$$EMA_t = \alpha(Q_t + (1-\alpha)^1 Q_{t-1} + (1-\alpha)^2 Q_{t-2} + (1-\alpha)^3 Q_{t-3} + \cdots)$$

If the previous EMA value is known, this can be simplified to

$$EMA_t = \alpha Q_t + EMA_{t-1}(1-\alpha)$$

This provides the change in the EMA due to Q_t, for a specified α a parameter and when EMA_{t-1} is known.

$$\Delta EMA = EMA_t - EMA_{t-1}$$

$$\Delta EMA = \alpha Q_t + EMA_{t-1}(1-\alpha) - EMA_{t-1}$$

$$\Delta EMA = \alpha Q_t - \alpha EMA_{t-1}$$

$$\Delta EMA = \alpha(Q_t - EMA_{t-1})$$

If EMA is determined using the minute-by-minute energy quantities, then this change in EMA is also in units of energy. With one-minute market duration and settlement, this can be easily expressed as a change in average power per minute.

$$\Delta\langle kW\rangle = \Delta EMA * 60 = \alpha(Q_t - EMA_{t-1}) * 60$$

This relationship is used in tariff settlement, where price incentives are expressed in terms of \$/ΔavgKW. Additionally, this relationship illustrates the value in storing the previous EMA value, *tariff.rampEMAcurrent*.

Before describing the tariff curve generation, α must be defined in terms of the provided tariff parameters.

When calculating EMA, the sum of the weights of the last *n* terms is described by $1 - (1 - \alpha)^{n+1}$, which approaches 1 as *n* approaches infinity. Alternatively, the sum of the weights prior to the last *n* terms is described by $(1 - \alpha)^{n+1}$.

This relationship provides insight on the number of data points needed for a specified degree of accuracy when determining EMA. For example, for a calculation of EMA to include 99.9% of its weight, the following relationship must be satisfied:

$$1 - (1 - \alpha)^{n+1} \geq .999$$

where *n* is the number of points considered. This can be rewritten as:

$$n + 1 \geq \frac{\log(.001)}{\log(1 - \alpha)}$$

Or generally as:

$$n \geq \frac{\log(1 - accuracy)}{\log(1 - \alpha)} - 1$$

The above holds true for any value of α between 0 and 1. However, it is useful to set α in reference to the "*N*-period", using the following equation:

$$\alpha = \frac{2}{N + 1}$$

By using this relationship, the weight of the last *N* terms converges on $1 - e^{\wedge} - 2 \approx 0.8647$ as *N* increases. However, the smaller the *N*-period, the greater the weight on the last *N* terms. As a consequence, the size of the *N*-period describes the EMA's "responsiveness" to changes in the data set. Small *N*-period values are more sensitive to entry-by-entry changes in data. This convention leads to the *N*-period as the primary parameter in describing the EMA function.

This parameter is stored as *tariff.rampEMAnPeriod*. Like *rampWindow* for the SMA ramp tariff, this values should be set with consideration of the denominator in the system ramp definition. However, the EMA adds a layer of complexity to this parameter; selecting this value for practical implementation requires further analysis.

In Matlab, the EMA for a vector *data* can be calculated for a specified *alpha* using a 1-D filter function in the following form:

$$EMA = filter(alpha, [1, alpha - 1], data)$$

Like the SMA calculation, consideration must be made for the initial simulation timesteps. The SMA ramp tariff used the initial quantity value for "padding". The EMA ramp tariff will use the same approach. Again, alternatives exist, but this approach is computationally straightforward. This method assumes the initial value provides leading padding of infinite length.

By assuming the initial value padding extends to infinitely, the EMA definition for t = 1 can be rewritten as

$$EMA_1 = \alpha(Q_1 + (1-\alpha)^1 Q_1 + (1-\alpha)^2 Q_1 + (1-\alpha)^3 Q_1 + \cdots)$$

which simplifies to

$$EMA_1 = Q_1$$

Note, this matches initial SMA value calculated with this convention. This is a useful result for comparing the SMA and EMA tariffs in simulation.

Using this initial value of EMA, subsequent values of EMA can be calculated using the previous definition of ΔEMA:

$$EMA_2 = EMA_1 + \Delta EMA = EMA_1 + \alpha(Q_2 - EMA_1)$$

This method is used for iterative calculation, such as determining EMA during Settlement for each timestep.

However, it may be desired to generate the EMA of a data set without iterative calculation. In this case, it is impractical to provide the infinite padding that this method implies. Instead, it is useful to pad the data set with a single value that represents this infinite series. This value is can be described as Q_0 and is found by returning to the EMA definition and recognizing that any point "earlier" than Q_0 must be zero.

$$EMA_1 = \alpha(Q_1 + (1-\alpha)^1 Q_0 + (1-\alpha)^2(0) + (1-\alpha)^3(0) + \cdots)$$

which simplifies to

$$EMA_1 = \alpha(Q_1 + (1-\alpha)^1 Q_0)$$

Using the previous relationship $EMA_1 = Q_1$ and solving for Q_0 yields

$$\alpha(1-\alpha)Q_0 = EMA_1 - \alpha Q_1 = (1-\alpha)Q_1$$

$$Q_0 = \frac{(1-\alpha)}{\alpha(1-\alpha)} Q_1 = \frac{Q_1}{\alpha}$$

This is a useful result when using the Matlab *filter* function to quickly generate the EMA of a data series using initial value padding. The result is equivalent to the iterative calculation of EMA.

6.9.2 Tariff Curve Generation

The aforementioned definitions for EMA and α are used for settlement and tariff curve generation. In particular, the equation for the change in average power is:

$$\Delta\langle kW\rangle = \Delta EMA * 60 = \alpha(Q_t - EMA_{t-1}) * 60$$

Which provides the revenue equation:

$$Revenue(Q) = \Delta\langle kW\rangle * priceRise = \alpha(Q_t - EMA_{t-1}) * 60 * priceRise, \qquad Q_t > EMA_{t-1}$$

$$Revenue(Q) = 0, \qquad Q_t = EMA_{t-1}$$

$$Revenue(Q) = \Delta\langle kW\rangle * priceFall = \alpha(EMA_{t-1} - Q_t) * 60 * priceFall, \qquad Q_t < EMA_{t-1}$$

Note, like the SMA ramp tariff, this tariff design does not include a tariff revenue offset. This could certainly be added to the tariff design, without impacting the principles described.

Note, α is always assumed to have the following form:

$$\alpha = \frac{2}{N+1}$$

Using the revenue equations, the tariff curve can be determined as follows:

$$P_{tariff}(Q) = \frac{d}{dQ} Revenue(Q)$$

$$P_{tariff}(Q) = \alpha * 60 * priceRise, \qquad Q_t > EMA_{t-1}$$

$$P_{tariff}(Q) = 0, \qquad Q_t = EMA_{t-1}$$

$$P_{tariff}(Q) = -\alpha * 60 * priceFall, \qquad Q_t < EMA_{t-1}$$

To implement this in Matlab, first the curve quantities are determined. Like the SMA ramp, this is based on the subnode's demand curve, to prevent the need to produce an infinitely horizontal tariff curve.

```
startQ = obj.sub.curve(1,1)-obj.Qbin;
endQ = obj.sub.curve(end,1)+obj.Qbin;
obj.curve = [];
obj.curve(:,1) = [startQ:obj.Qbin:endQ];
```

Next, the current EMA is used as the target quantity, with the equations above setting the tariff curve prices.

```
alpha = 2/(obj.rampEMAnPeriod+1);
lowerP = -obj.rampEMApriceFall*60*alpha;
upperP = obj.rampEMApriceRise*60*alpha;
refQ = obj.rampEMAcurrent;
obj.curve(:,2) = lowerP*(obj.curve(:,1)<refQ) + ...
      upperP*(obj.curve(:,1)>refQ);
```

Note, the above only applies for *DDS.t* > 1. For the initial timestep, there is no reference EMA values, so there is no tariff curve or settlement. After the first timestep, a reference EMA value will have been established.

This completes tariff curve generation, for the proposed EMA ramping tariff design.

6.9.3 Settlement

Settlement is accomplished using the revenue equations defined in Section 6.9.2. Additionally, the settlement process also includes updating the parameter *tariff.rampEMAcurrent*. For a given measured energy consumption, *quant*:

```
alpha = 2/(obj.rampEMAnPeriod+1);
if obj.DDS.t == 1
      obj.revenue = 0;
      obj.rampEMAcurrent = quant;
else
      deltaEMA = alpha*(quant-obj.rampEMAcurrent);
      if deltaEMA > 0
            obj.revenue = deltaEMA*60*obj.rampEMApriceRise;
      elseif deltaEMA < 0
            obj.revenue = deltaEMA*60*obj.rampEMApriceFall;
      else
            obj.revenue = 0;
      end
      obj.rampEMAcurrent = obj.rampEMAcurrent + deltaEMA;
end
```

Note, the tariff curve and settlement process for this tariff design assumes the settlement period and market duration are both one minute. This avoids the disruptions described as limitations for the capacity and SMA ramp tariff designs. However, it limits the DTDM simulation and implementation.

Additionally, the EMA settlement does not include a revenue offset. This parameter should be added for tariff owner flexibility. Adding this parameter would not impact the DTDM response to the tariff, only the collected revenues.

6.9.4 Limitations and Areas for Improvement

The EMA ramp tariff serves to reduce the cycling that occurs when implementing the SMA ramp tariff. A comparison of results will be provided as a case study in Section 7. However, the EMA ramp tariff shares some of the same limitations as the SMA ramp tariff. For one, the EMA ramp tariff would benefit from a revenue offset parameter.

Additionally, using the initial record quantity for padding the EMA provides a crude approximation. With the EMA, the impact of this approximation extends to infinitely, although the impact drops off exponentially. Like the SMA tariff, a simulation should run in advance of the area of interest, to limit the impact of this approximation.

6.10 Flat-Rate Tariff

The final proposed tariff is the most straight-forward. A flat-rate tariff simply applies a constant price incentive to energy passing through the linkage.

A flat-rate tariff could represent incentives (e.g. a subsidy) or disincentivies (e.g. a variable carbon tax). As an example, a flat-rate tariff could be based on a Renewable Portfolio Standard (RPS). In this case, the utility could recognize the financial benefit of distributed renewable energy production, as it contributed to the utility's overall RPS. Based on a defined structure, the utility could provide a subsidy for all exports from DER. This would be represented by a Flat-Rate Tariff imposed above the DER node in the DTDM network.

This tariff design is described as a "flat-rate" because the tariff curve is flat. However, the tariff need not be constant over time, like in the RPS example.

One example of a time-varying flat-rate tariff is a carbon tax on grid imports. In this case, a flat-rate tariff is imposed between the DTDM Marketplace Node and the Wholesale Grid. If the fuel mix of current grid generation is available, the carbon emissions per kWh can be estimated. With a pre-established carbon tax per pound of carbon, a flat-rate tariff can reflect the externalities of wholesale energy carbon emissions, on a time-varying basis.

The simulated implementation of this tariff includes only two parameters. The current rate is *tariff.flatRate*. To support time-varying rates, this is provided by referencing *tariff.flatDataSource*.

The revenue equation and resulting tariff curve function are as follows:

$$Revenue(Q) = flatRate * Q$$

$$P_{tariff}(Q) = flatRate$$

Notice, the tariff curve is constant; hence the name "flat-rate tariff".

The tariff curve is set in the method *tariff.setFlatCurve.* Like the ramping tariffs, the tariff curve quantities are determined by the subnode's demand curve:

```
startQ = obj.sub.curve(1,1)-obj.Qbin;
endQ = obj.sub.curve(end,1)+obj.Qbin;
obj.curve = [];
obj.curve(:,1) = [startQ:obj.Qbin:endQ];
```

The price points on the tariff curve are established by referencing *tariff.flatDataSource* for the current timestep.

```
obj.flatRate = obj.flatDataSource(marketStart);
obj.curve(:,2) = obj.flatRate;
```

During settlement, the revenue for a given *quant* is provided by:

```
obj.revenue = quant*obj.flatRate;
```

This tariff design is very straightforward. However, it has some limitations and areas for improvement.

One, the proposed tariff design does not support market durations longer than one minute. Doing so would provide a challenge for time varying flat-rate tariffs. Specifically, the revenue function for a time-varying rate with market duration *n* is as follows:

$$Revenue(Q_1, Q_2, \ldots, Q_n) = rate_1 Q_1 + rate_2 Q_2 + \cdots + rate_n Q_n$$

In the current DTDM simulation, the tariff generates the tariff curve directly. If the flat-rate does not change over time, this is a straightforward derivative of the revenue function. However, for a time-varying rate with time varying consumption preferences, there is nuance in interpreting the revenue function. This nuance requires value judgements and probabilistic risk assessment by the subnode. Like other tariff designs, this could be resolved by requiring the tariff curve interpretation to be accomplished by the subnode's behavior model. This is an area for additional research.

Two, the proposed flat-rate tariff applies to all quantities, extending to positive and negative infinity. However, this does not match all prosed use cased for the flat-rate tariff. For example, a grid-import carbon tax would only apply to imports, not exports to the grid. With this in mind, consideration must be made to ensure the tariff revenue function is convex and the tariff curve is non-descending.

In many use cases, this will occur naturally. However, use cases can be contrived that seek to violate these conditions (e.g. a subsidy for exports but no penalty for imports). When this occurs, the overall system should be examined for solutions. The subnode may naturally have consumption limits, eliminating the non-convexity from the set of possible quantities. Alternatively, artificial tariff curve limits could be imposed. It is not expected that this will be a common concern, but it must be considered in any implementation.

This concludes the Behavior Models and Proposed Tariff Designs included in the DTDM simulation. Next, some case studies are presented to demonstrate the interactions of these behavior models and tariff designs, within the DTDM.

7 Simulation: Case Studies

The following case studies are presented to demonstrate the interactions between behavior models and tariff designs within the DTDM. The fundamental purpose of the DTDM is to facilitate energy transactions and provide differentiated prices based on energy, losses, and externalities. To demonstrate the DTDM's ability to do so, the example network will include multiple sources and impose externalities at different locations.

7.1 Overview

All case study scenarios will use the DTDM network illustrated below. Some scenarios will also include additional nodes or tariffs; those scenarios will have updated network diagrams.

The case study scenarios simulate a residential feeder with 20 households. Each household has a Home Energy Manager (HEM) through which it participates in the DTDM. Four households have rooftop PV behind the meter; there is one such household on each distribution transformer. The DSO operates a market at substation, at which point the distribution system connects to the larger electricity grid.

The simulation runs for 72 hours, representing 12:00 on June 1, 2014 12:00 to 12:00 on June 3, 2014, with one-minute simulation timesteps. The market duration is one minute. Pricing for imports from the electricity grid is provided by NY ISO real-time wholesale energy pricing for this time period [18].

.

Household loads and PV generation are determined using Pecan Street data as a baseline [17]. Each household consists of an air conditioner, from the Pecan Street *air1* category, and "other" load, from subtracting *air1* from the Pecan Street *use* category. PV generation will be determined using Pecan Street *gen* category data as a baseline. As described in previous sections, this data will be cleaned, so all load anchor quantities are non-negative and all PV reference quantities are non-positive.

Unless denoted, the price elasticity for air1 is 0.6 and the price elasticity for other loads is 0.2. This applies to all households. These values are increased beyond typical values solely to illustrate the effects of the market. For an analysis with more accurate price elasticities, more accurate behavior models would be warranted, especially with regards to time-varying and consumption-dependent elasticities.

For the anchor price in the load node behavior models, the Flat-Rate Equivalent price is determined by calculating the median NY ISO real-time price, for the period examined. This ensures that the DTDM prices will result in load nodes consuming both more and less than their baseline energy quantities.

In the DTDM, each household is a unique owner, the wholesale grid node is owned by the ISO, and all other nodes are owned by the DSO.

For simplification, only three loss linkage parameters are used in the DTDM network. The power factor is assumed to be 0.9 for the entire system, at all times.

One, virtual nodes and intra-household linkages are assumes to be zero. The latter assumes the all household energy measurements occur at the same point: the AMI or HEM.

For the others, recall the description of the linkage loss parameter in Section 5.3.3:

$$k = \left(\frac{60 * R}{1000 * V_{nom}^2 * pf^2}\right)$$

Two, service conductors are assumed to connect households to their distribution transformer. The nominal voltage is assumed to be 120 V. The conductor is assumed to be #3 AWG copper conductor, to support 100 A. The National Electric Code, Table 8 indicates #2 AWG copper conductors have a per length resistance of 0.201 Ω per 1000 feet [22]. It is assumed these service conductor is 50 feet long.

$$k_{service} = \left(\frac{60 * \left(\frac{.201}{1000} * 50\right)}{1000 * 120^2 * 0.9^2}\right) = 5.1698\ x\ 10^{-8}$$

Three, feeder conductors are assumed to connect distribution transformers to other distribution transformers and to the distribution bus. It is assumed that each feeder linkage is 500 feet in length. The nominal voltage is assumed to be 12.47 kV phase-to-phase or 7.2 kV phase-to-ground. The conductor is assumed to be bare, steel-reinforced, aluminum conductor. A Southwire product specification sheet indicates #6 AWG has an allowable ampacity of 105 A and a resistance of 0.806 Ω / 1000 feet. This will be used at the feeder conductor assumption.

$$k_{feeder} = \left(\frac{60 * \left(\frac{.806}{1000} * 500\right)}{1000 * 7200^2 * 0.9^2}\right) = 5.7585\ x\ 10^{-10}$$

Each case study scenario will examine an aspect of the DTDM rules, simulation, behavior models, and/or proposed tariffs. Each scenario will include multiple simulations for comparison. The first simulation is used as a baseline, while the subsequent simulations are used to compare the results of differing parameters.

The scenarios are described as follows:

Scenario 1: Load Elasticity

Simulation 1.1: Baseline; zero elasticity and zero prediction error for all loads and PV generation

Simulation 1.2: Reasonable elasticity and reasonable prediction error

Simulation 1.3: Exaggerated elasticity and reasonable prediction error

Scenario 2: Capacity Tariff

Simulation 2.1: Baseline; no capacity tariff

Simulation 2.2: Capacity tariff on DSO Xfmr 4 with limits only (i.e. no tariff curve ramp)

Simulation 2.3: Capacity tariff on DSO Xfmr 4 with tariff curve ramp at 80%

Scenario 3: Ramp SMA Tariffs

Simulation 3.1: Baseline; no tariffs

Simulation 3.2: Weak Ramp SMA tariff on DSO Market

Simulation 3.3: Strong Ramp SMA tariff on DSO Market

Scenario 4: Ramp EMA Tariff Location

Simulation 4.1: Baseline; no tariffs

Simulation 4.2: Ramp EMA tariff on DSO Market

Simulation 4.3: Ramp EMA tariff on all Household HEMs

Scenario 5: Dispatcher Node

Simulation 5.1: Baseline; prediction error, ramp EMA tariff on DSO Market, no Dispatcher

Simulation 5.2: Dispatcher Node with dispatchable storage; ‘contract’ type control

Simulation 5.3: Dispatcher Node with dispatchable storage; ‘dynamic’ type control

The Matlab code initializing the case studies is provided in Appendix A.

Finally, a note on evaluating cast study results. It is difficult to compare case study results for a specific node. Consider adding a PV node to a DTDM network. Naturally, the value of solar generation is the based on the NMPs at which the PV node exports energy. However, if the PV node is added to a DTDM subsystem with a common owner, that owner may value solar generation more than this naïve calculation. Specifically, the PV node will impact the NMP for other nodes in the subsystem; this may adjust the energy consumption and NMPs for the owner’s other nodes. Thus, the total value derived from the PV node cannot be simply described by its energy exports.

However, this makes comparisons between scenarios difficult. For the above example, even levelized cost of energy an incomplete representation of the impact. A better description would be the customer’s economic surplus for electric energy. This could be estimated by examining the customer’s demand curves and the resulting consumption and pricing. This method for estimating value is not accomplished in these case studies.

Instead, these case study provide an opportunity to observe the general trends and results of node behavior models and tariff designs. Again, these case studies are not intended to be a demonstration of the expected results of DTDM implementation. They are intended as a demonstration of interaction between actors and tariffs, within the constraints of the market rules. As such, some scenarios will include realistically unreasonable values for parameters that are not the focus of the scenario; this is done to emphasis the result of interactions of interest.

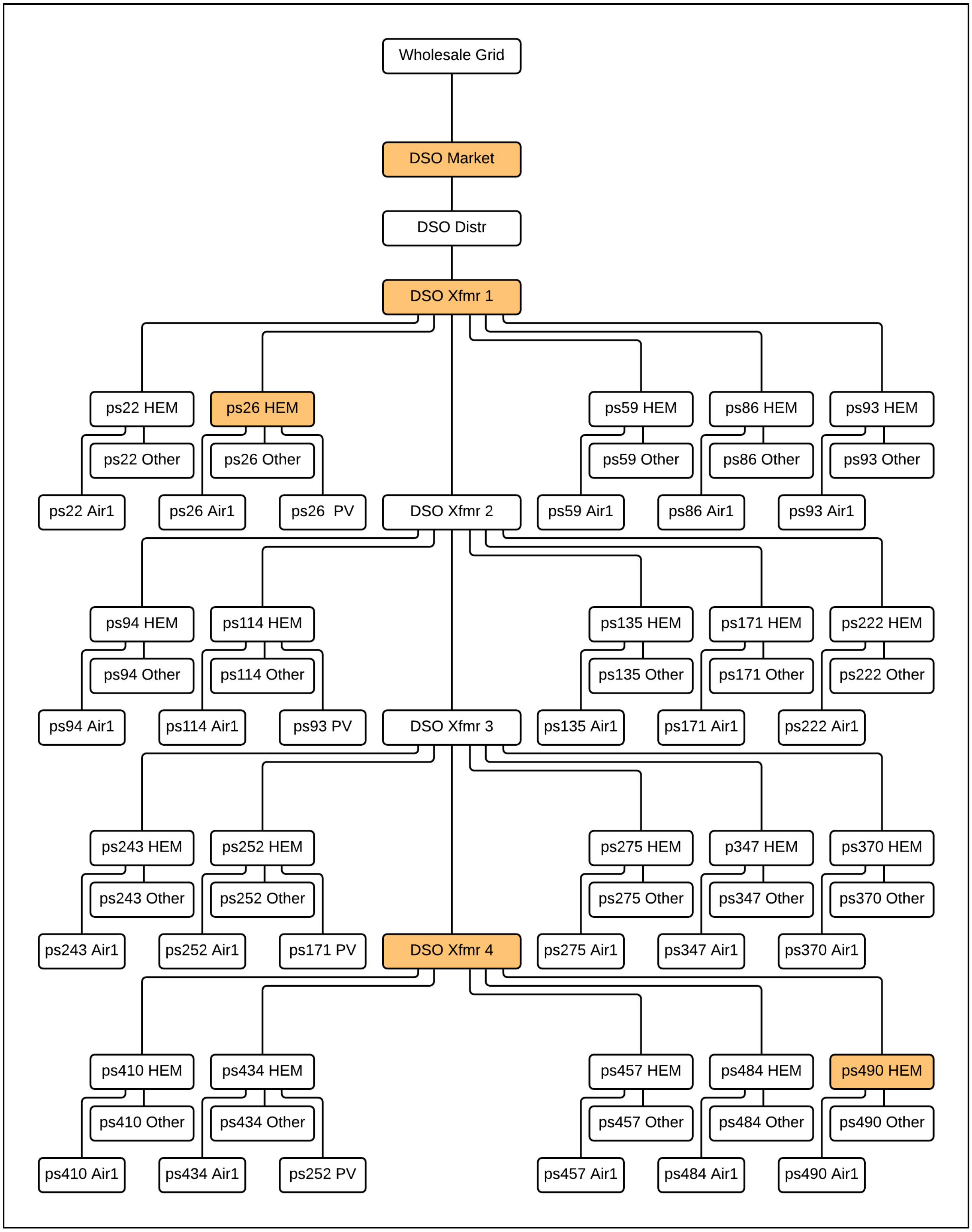


Figure 57: DTDM Network for Case Studies (Baseline; Scenarios 1.1, 1.2, 1.3, 2.1, 3.1)

7.2 Scenario 1: Load Elasticity

7.2.1 Configuration

This scenario is designed to demonstrate the impact of incorporating price elasticity to load nodes.

In Scenario 1.1, the base case is established.

```
elastAir1 = 0;          % Elasticity of air1 loads
predictAir1 = 0;        % Prediction error for air1 loads
elastOther = 0;         % Elasticity of all other loads
predictOther = 0;       % Prediction error for all other loads
predictPV = 0;          % Prediction error for PV generation
```

In Scenario 1.2, reasonable elasticity and prediction error are implemented.

```
elastAir1 = 0.4;        % Elasticity of air1 loads
predictAir1 = 0.05;     % Prediction error for air1 loads
elastOther = 0;         % Elasticity of all other loads
predictOther = 0.2;     % Prediction error for all other loads
predictPV = 0.05;       % Prediction error for PV generation
```

In Scenario 1.3, exaggerated elasticity and prediction error are implemented.

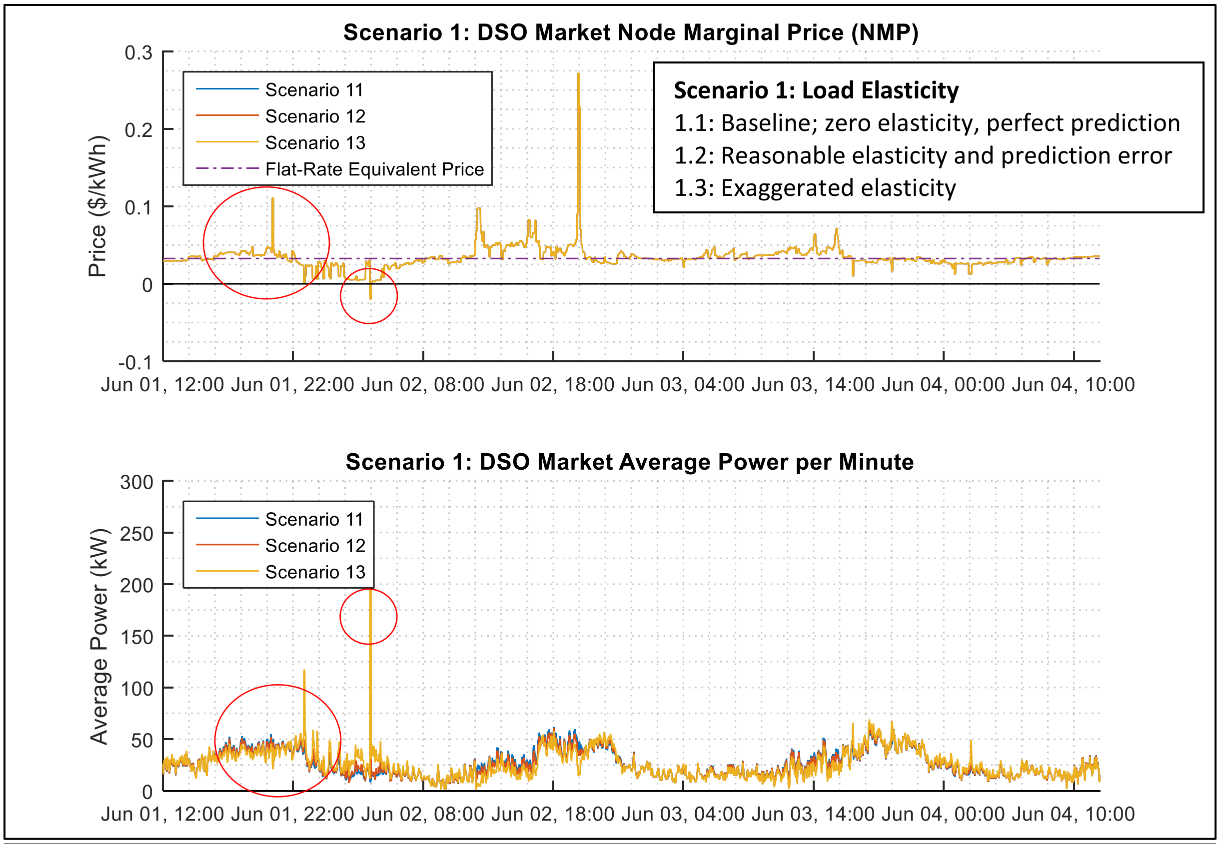
```
elastAir1 = 0.8;        % Elasticity of air1 loads
predictAir1 = 0.05;     % Prediction error for air1 loads
elastOther = 0.4;       % Elasticity of all other loads
predictOther = 0.2;     % Prediction error for all other loads
predictPV = 0.05;       % Prediction error for PV generation
```

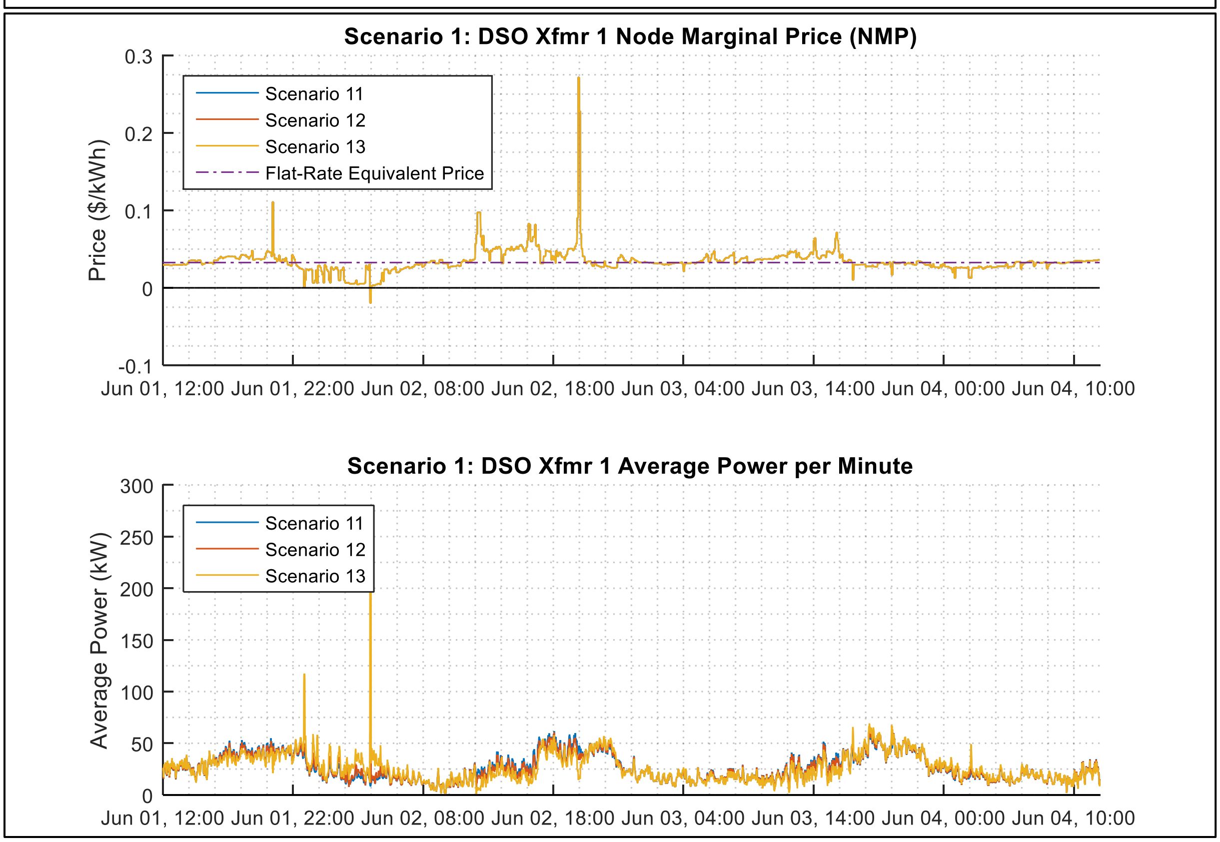
7.2.2 Results

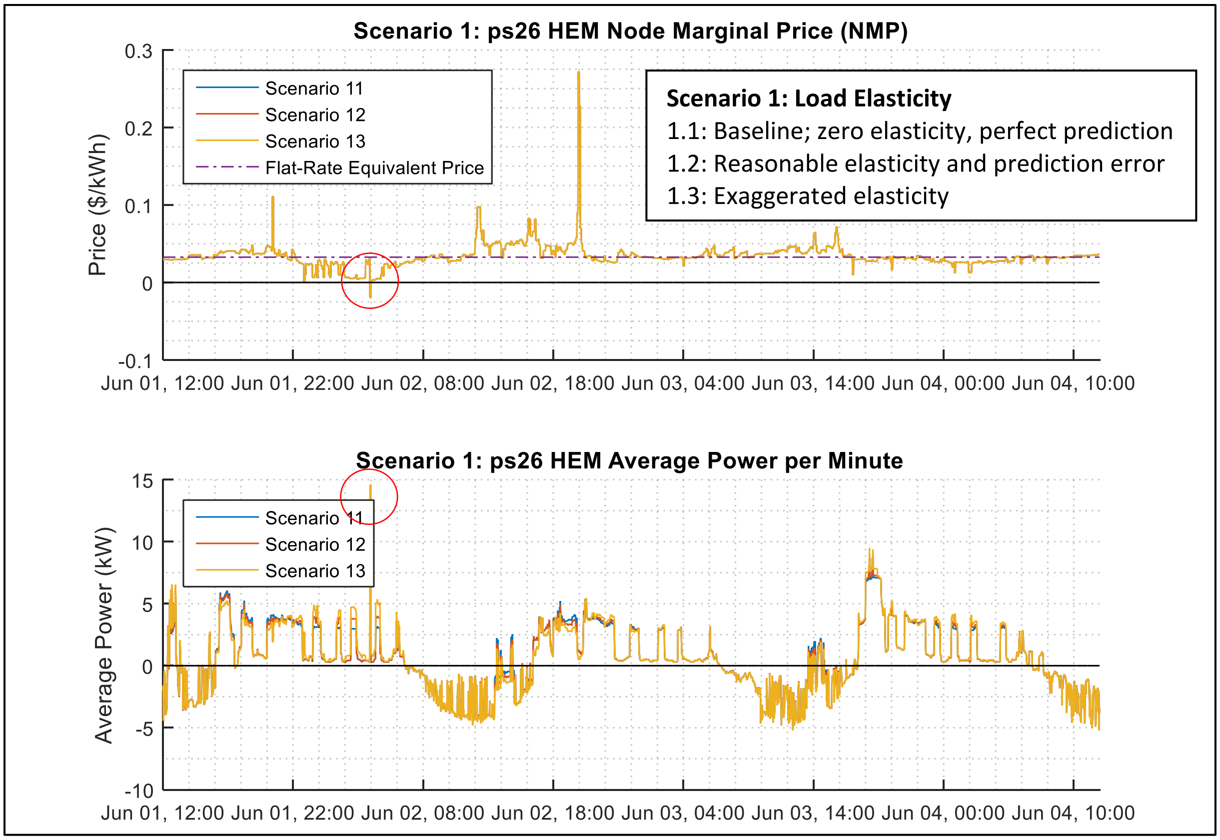
The plots in Figure 58 below illustrate the results of the simulation. All scenarios use the same plot configuration for illustration. Each examined node is given its own set of plots.

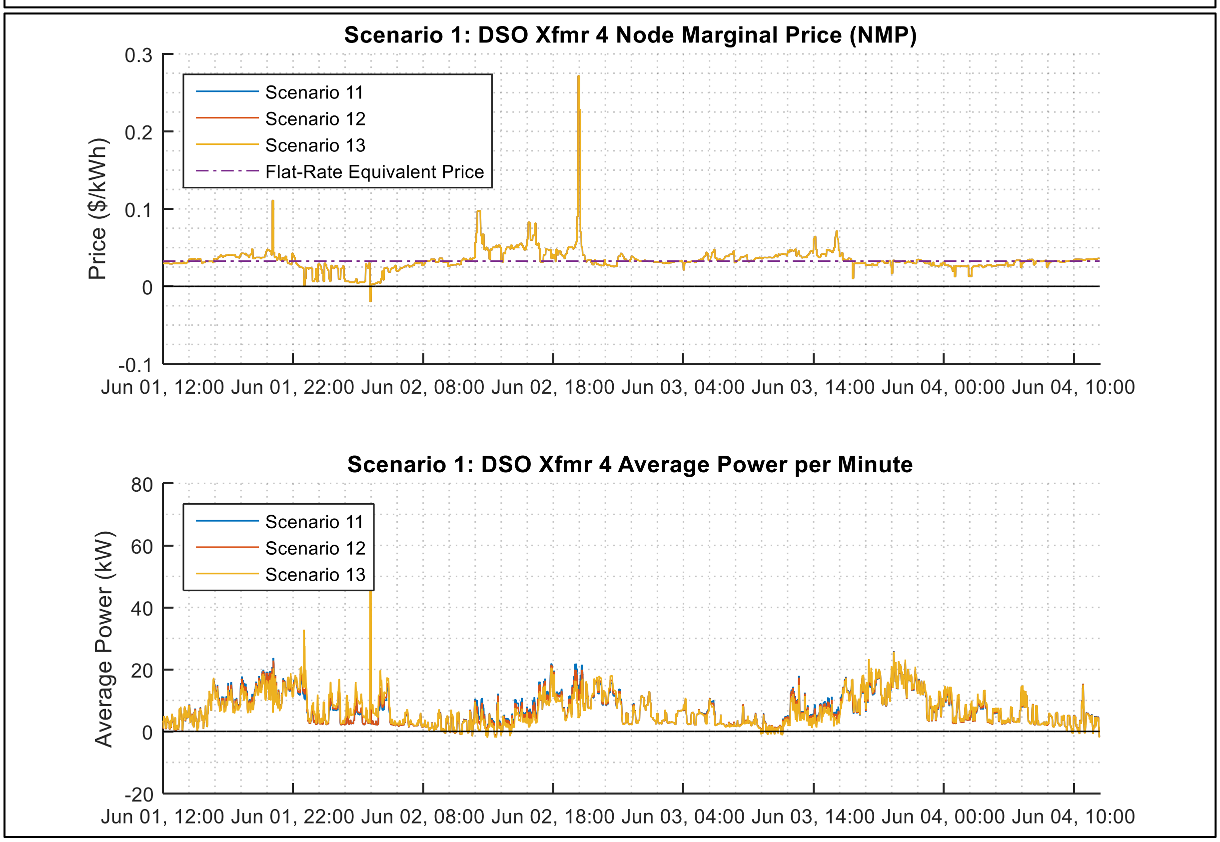
The top subplot displays the NMP for the node, *priceContr*, for each of the scenarios. The Flat-rate Equivalent price used for the simulation, *priceFRE*, in provided as a reference. Recall, the NMP is not the same as the node's clearing price, which will be modified based on the imposed tariffs.

The bottom subplot displays the average power per minute, based on *quantActual*, for each of the scenarios.


Scenario 1: DSO Market Node Marginal Price (NMP)
Price ($/kWh)
Scenario 11
Scenario 12
Scenario 13
Flat-Rate Equivalent Price
Scenario 1: Load Elasticity
1.1: Baseline; zero elasticity, perfect prediction
1.2: Reasonable elasticity and prediction error
1.3: Exaggerated elasticity
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 1: DSO Market Average Power per Minute
Average Power (kW)
Scenario 11
Scenario 12
Scenario 13
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00


Scenario 1: DSO Xfmr 1 Node Marginal Price (NMP)
Price ($/kWh)
Scenario 11
Scenario 12
Scenario 13
Flat-Rate Equivalent Price
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 1: DSO Xfmr 1 Average Power per Minute
Average Power (kW)
Scenario 11
Scenario 12
Scenario 13
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00


Scenario 1: ps26 HEM Node Marginal Price (NMP)
Price ($/kWh)
Scenario 11
Scenario 12
Scenario 13
Flat-Rate Equivalent Price
Scenario 1: Load Elasticity
1.1: Baseline; zero elasticity, perfect prediction
1.2: Reasonable elasticity and prediction error
1.3: Exaggerated elasticity
0.3
0.2
0.1
0
-0.1
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 1: ps26 HEM Average Power per Minute
Average Power (kW)
Scenario 11
Scenario 12
Scenario 13
15
10
5
0
-5
-10
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00


Scenario 1: DSO Xfmr 4 Node Marginal Price (NMP)
Price ($/kWh)
Scenario 11
Scenario 12
Scenario 13
Flat-Rate Equivalent Price
0.3
0.2
0.1
0
-0.1
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 1: DSO Xfmr 4 Average Power per Minute
Average Power (kW)
Scenario 11
Scenario 12
Scenario 13
80
60
40
20
0
-20
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00

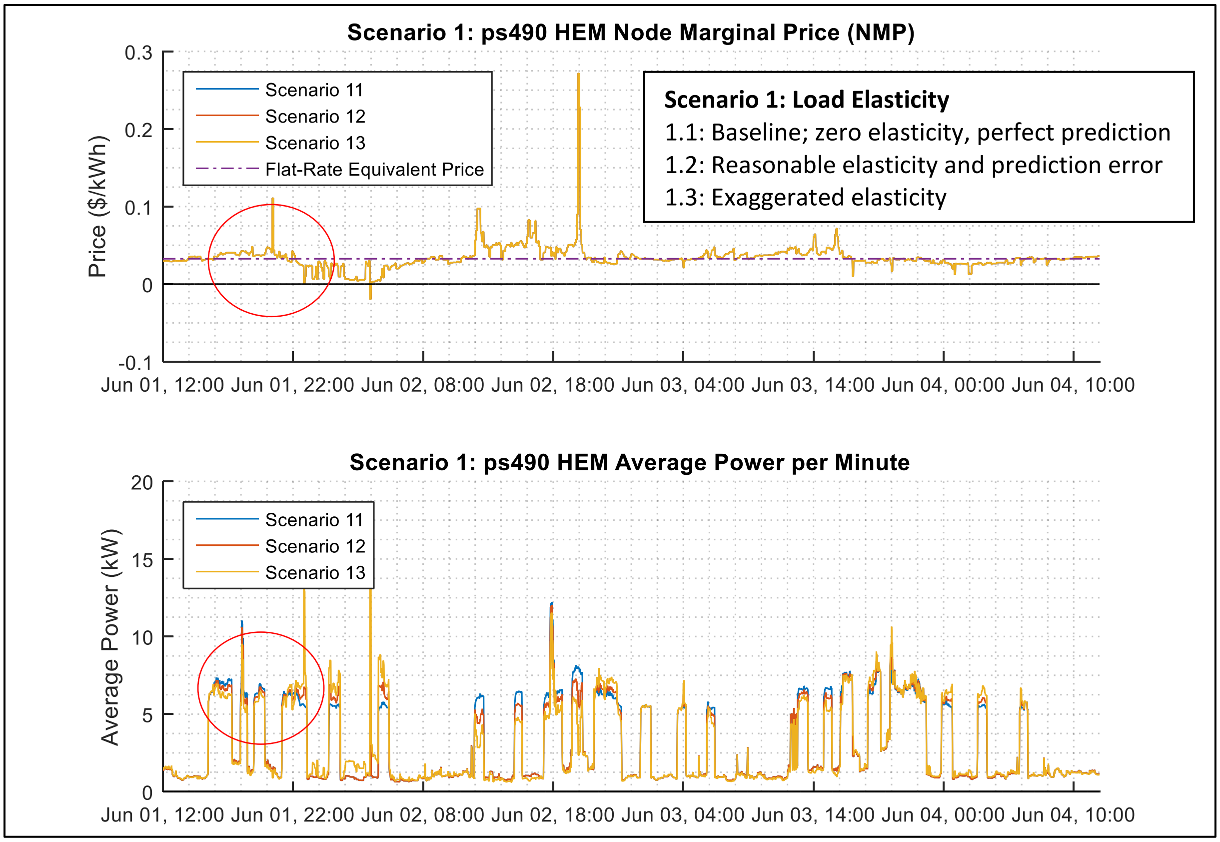


Figure 58: Scenario 1 Results

7.2.3 Observations

This case study provides evidence that elastic load nodes modify their behavior based on price signals. When the NMP is greater than the FRE price, the consumption decreases; when the NMP is less than the FRE price, the consumption increases. This observation, as well as the following observations, is highlighted with a red circle in Figure 58 above.

As the elasticity is increased, the impact on the consumed energy increases. Observe that the quantity difference increases with the magnitude of the baseline quantity; this is a consequence of the isoelastic function.

Additionally, despite this effect occurring only at the bottom-level nodes, it is clear there is a cumulative impact at the top-level DSO Market Node. Note, this node also represents the energy flow observed by the larger electrical grid.

There is no observed impact by implementing prediction error. This is because the PV energy sources are inelastic (other than curtailment) and the wholesale grid is considered an infinite bus. The impact of prediction error is seen when tariffs are implemented in the system. This will be demonstrated in Scenario 5.

Finally, observe the massive price spike at 03:55 on 2-Jun-2014. This is a consequence of the negative wholesale energy price. Based on the isoelastic behavior model, all load nodes, upon seeing this as the NMP, consumed their maximum possible energy quantity. The cumulative consumption is over five times the largest recorded base energy quantity. This may not be a realistic response; not every node may be able to respond in this way.

Yet this spike in demand does indicate a potential problem: when every node in the system uses the same optimization algorithm, the cumulative response to an extreme price may be unacceptable. If this were to occur in a practical system, it would most likely cause overload conditions.

Scenario 1 does not include any tariffs, so it does not provide a complete demonstration of the DTDM in action. Specifically, this overload condition may have been avoided with capacity tariffs.

7.3 Scenario 2: Capacity Tariff

7.3.1 Configuration

This scenario is designed to demonstrate the impact of adding a capacity tariff to the system. Additionally, by implementing a capacity tariff at only one location in the DTDM network, the scenario illustrates impact of tariffs on different subsections of the network.

For this scenario, the following prediction and elasticity values are used. These values also apply for Scenarios 3-4. The elasticity is increased to make clear the interactions of tariff and node price signals. To achieve a similar effect in an actual implementation, tariff parameter may need to be modified. Additionally, perfect prediction is used to simplify the network. Without perfect prediction, either energy storage or hedging is necessary to achieve the desired response to tariffs.

```
elastAir1 = 0.6;         % Elasticity of air1 loads
predictAir1 = 0;         % Prediction error for air1 loads
elastOther = 0.2;        % Elasticity of all other loads
predictOther = 0;        % Prediction error for all other loads
predictPV = 0;           % Prediction error for PV generation
```

In Scenario 2.1, the base case is established. This uses the network shown in Scenario 1, with no tariffs. In Scenario 2.2, a capacity tariff is added above DSO Xfmr 4. This is a non-ramping capacity tariff with the power capacity limit set to 20 kW. In Scenario 2.3, the capacity tariff is modified. The tariff is changed to a 'linear' type ramp curve. With *capPpeak* = $0.05/kWh and *capQrampPct* = 0.8; the tariff begins assessing penalties at 16 kW.

7.3.2 Results

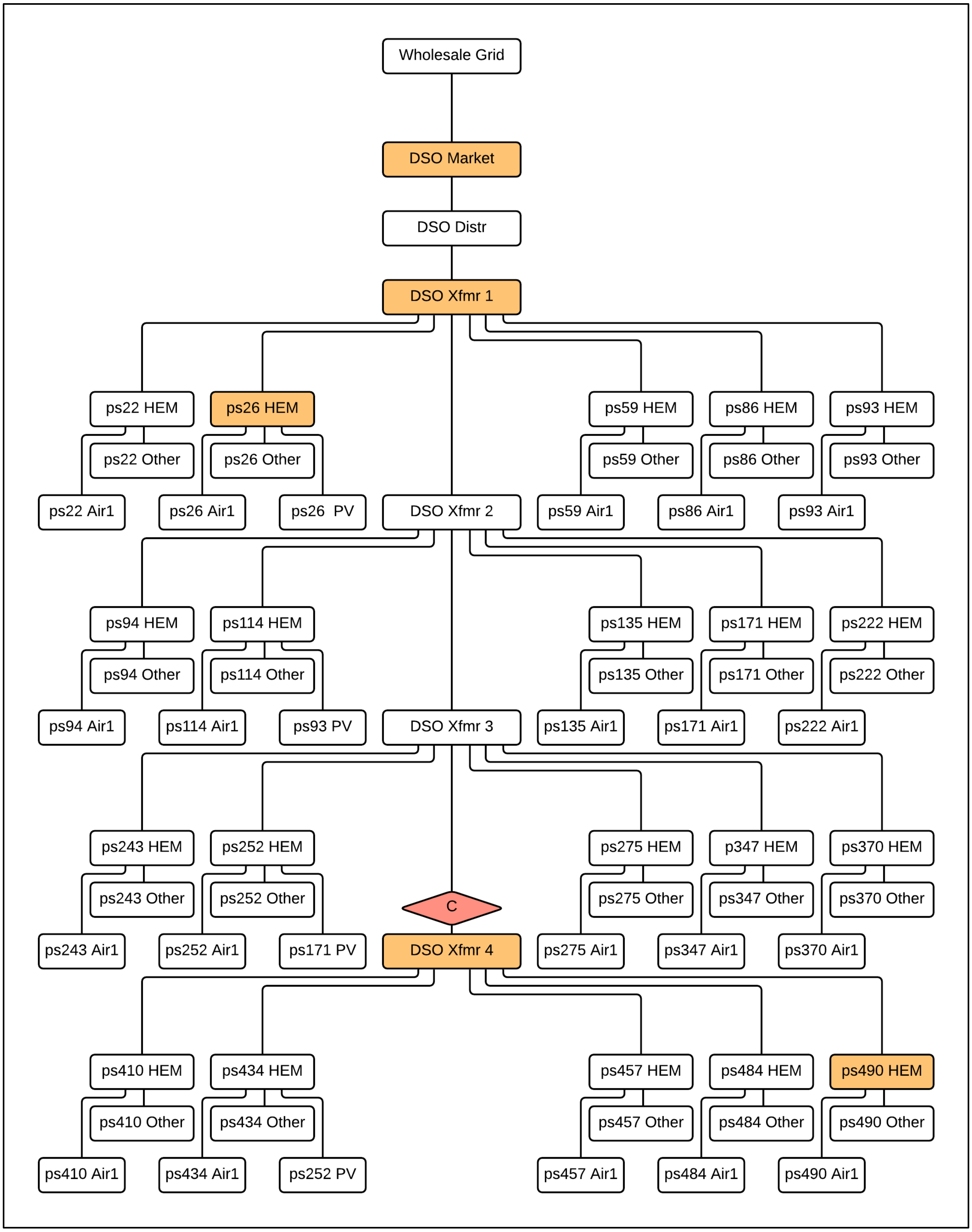
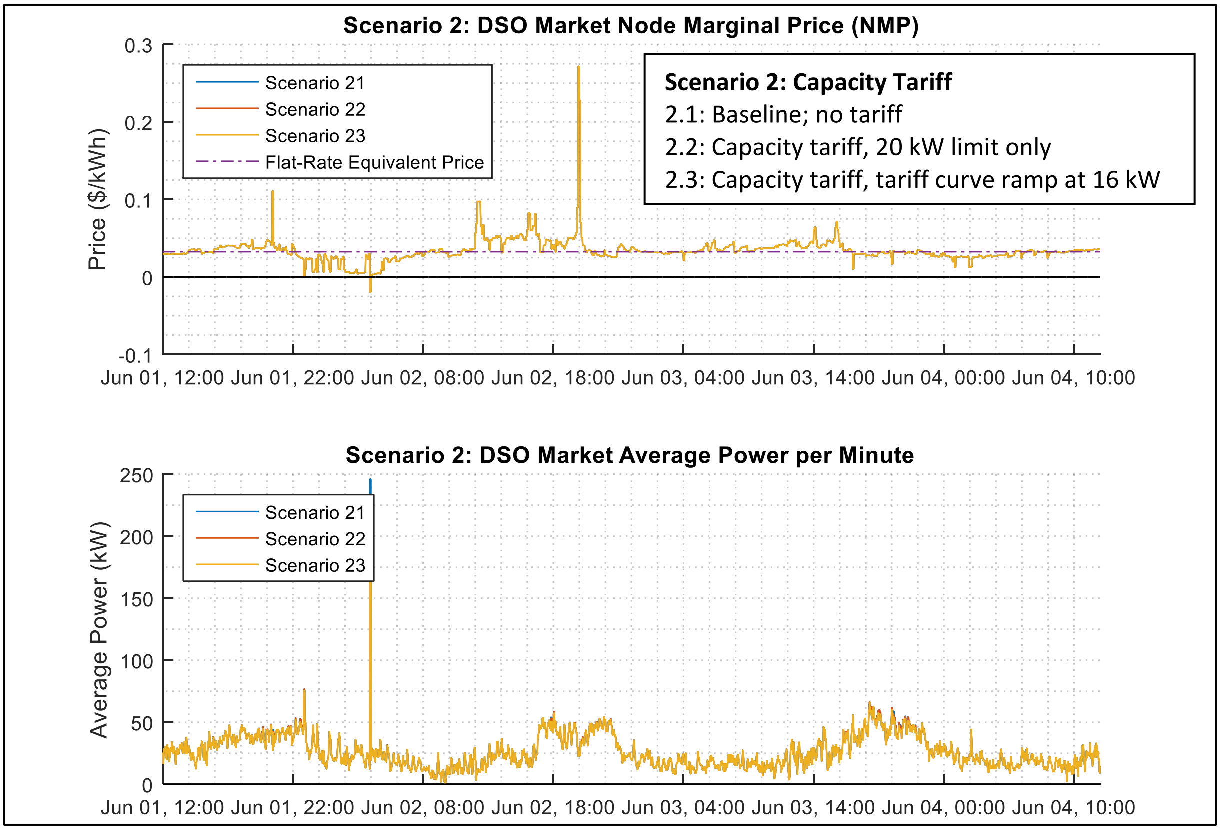
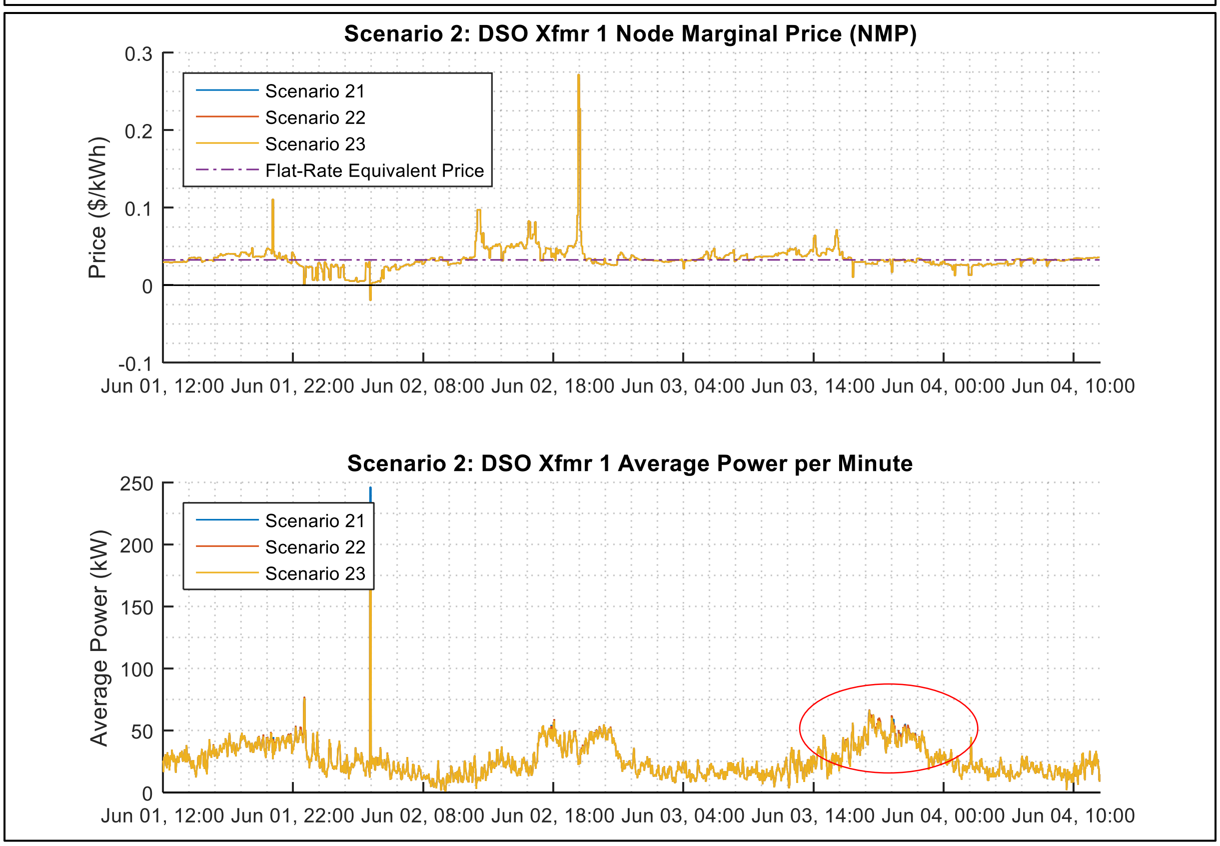
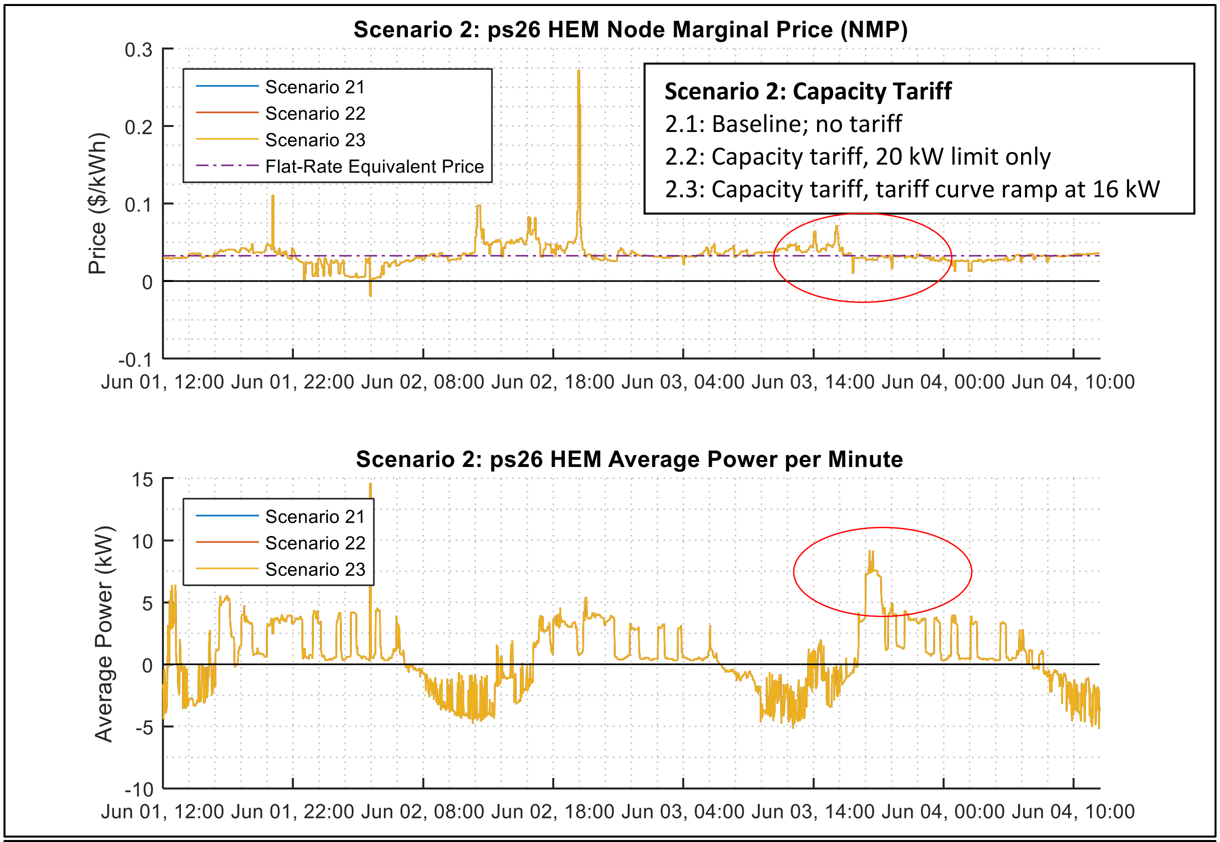
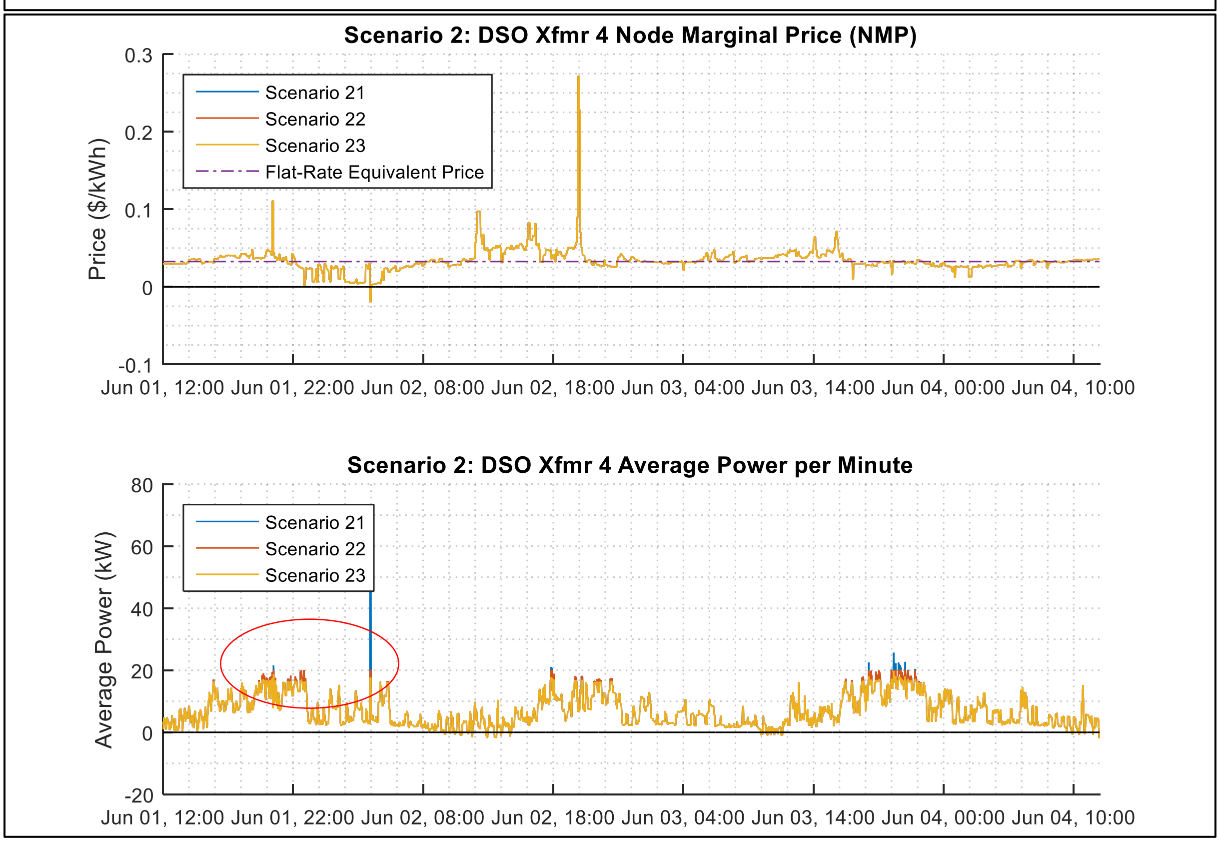


Figure 59: DTDM Network with Capacity Tariff (Scenario 2.2, 2.3)


Scenario 2: DSO Market Node Marginal Price (NMP)
Scenario 21
Scenario 22
Scenario 23
Flat-Rate Equivalent Price
Scenario 2: Capacity Tariff
2.1: Baseline; no tariff
2.2: Capacity tariff, 20 kW limit only
2.3: Capacity tariff, tariff curve ramp at 16 kW
Price ($/kWh)
0.3
0.2
0.1
0
-0.1
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 2: DSO Market Average Power per Minute
Scenario 21
Scenario 22
Scenario 23
Average Power (kW)
250
200
150
100
50
0
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00


Scenario 2: DSO Xfmr 1 Node Marginal Price (NMP)
Scenario 21
Scenario 22
Scenario 23
Flat-Rate Equivalent Price
Price ($/kWh)
0.3
0.2
0.1
0
-0.1
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 2: DSO Xfmr 1 Average Power per Minute
Scenario 21
Scenario 22
Scenario 23
Average Power (kW)
250
200
150
100
50
0
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00


Scenario 2: ps26 HEM Node Marginal Price (NMP)
Price ($/kWh)
0.3
0.2
0.1
0
-0.1
Scenario 21
Scenario 22
Scenario 23
Flat-Rate Equivalent Price
Scenario 2: Capacity Tariff
2.1: Baseline; no tariff
2.2: Capacity tariff, 20 kW limit only
2.3: Capacity tariff, tariff curve ramp at 16 kW
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 2: ps26 HEM Average Power per Minute
Average Power (kW)
15
10
5
0
-5
-10
Scenario 21
Scenario 22
Scenario 23
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00


Scenario 2: DSO Xfmr 4 Node Marginal Price (NMP)
Price ($/kWh)
0.3
0.2
0.1
0
-0.1
Scenario 21
Scenario 22
Scenario 23
Flat-Rate Equivalent Price
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 2: DSO Xfmr 4 Average Power per Minute
Average Power (kW)
80
60
40
20
0
-20
Scenario 21
Scenario 22
Scenario 23
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00

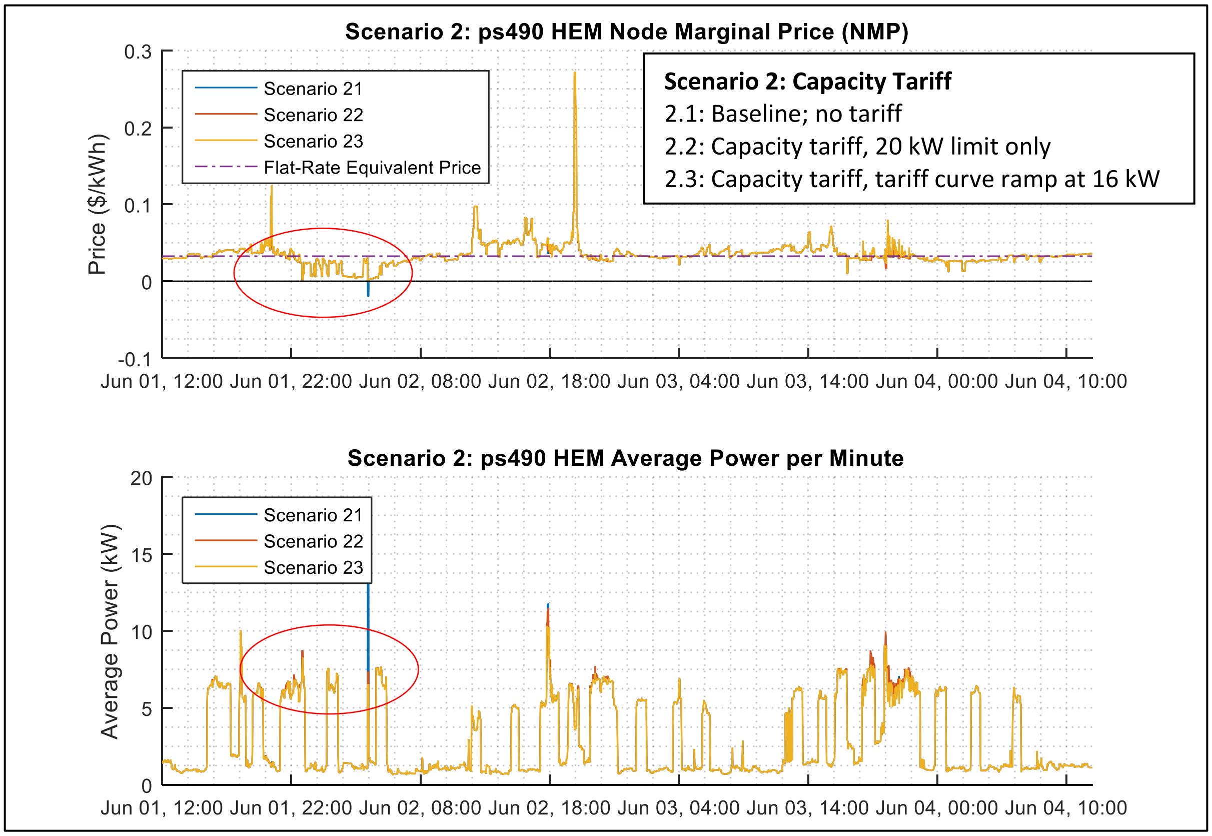


Figure 60: Scenario 2 Results

7.3.3 Observations

There are numerous observations to be made from these results. These are highlighted with red circles in Figure 60 above.

One, the intended impact is observed by examining the average power at DSO Xfmr 4. With the tariff in place, the demand never exceeds the component limits. This is a successful implementation. Notice the price is used to produce this response. The NMP for DSO Xfmr 4 matches that of the wholesale market; this is the same price at DSO Market. However, the DSO Xfmr 4 node is determining a clearing price for all of its subnodes; this is seen as the NMP for ps490 HEM. Observe that, when the capacity would have otherwise exceeded 20 kW, the NMP for ps490

HEM rises above that in the base case. This is the price signal used to incentivize reduced energy consumption. This is a useful result.

Two, the plot of price points for ps490 HEM provides interesting results. In scenarios other than the base case, the price no longer falls below zero. This is to prevent the massive consumption response induced by negative prices, as seen in Scenario 1. However, the price also changes at other points in time. These are points in time at which the natural system response exceeded or approached the limits of the capacity tariff. As a result, all nodes, including ps490 HEM received a higher NMP; despite the fact those timestep were not necessarily periods of high consumption for ps490 HEM. The differentiated price signals reflect the DTDM subsystem conditions, not simply an individual's energy consumption.

Three, the plot of quantities for DSO Xfmr 4 illustrate the impact of the ramp-type capacity tariff curve in Scenario 2.3. In Scenario 2.2, the capacity tariff simply prevents consumption over 20 kW. The quantities in this scenario never exceed that amount. However, in Scenario 2.3, the tariff imposes penalties for all consumption above 16 kW, up to the limit at 20 kW. This penalty is seen in the plot; quantities rarely reach above 16 kW. However, average power flow does, in fact, exceed 16 kW, although in limited magnitude and at limited occasions. These forays above 16 kW can be seen as instances in which customers' preference justified facing the tariff. This is a useful result. The DSO may select tariff parameters to meet the system goals, based on the system and their risk analysis. For example, the tariff curve shape can serve as a "buffer", to limit (but not prevent) power flow from approaching the true component limits.

Four, notice that the effective quantity limit observed by ps490 HEM is different at different points in time, despite the fundamental capacity tariff being unchanged. Specifically, in Scenario 2.2, the capacity tariff results in a consumption of 7.33 kW at 3:57 on 2-June but a consumption of 11.4 kW at 17:53 on 2-June. In fact, the latter is very close to the node's baseline consumption. This further demonstrates differentiated price signals as a tool for capturing externalities. At both timesteps, the DTDM subsystem faced a capacity limit. But the system operator did not impose an across-the-board quantity restriction in either occasion. Instead, a price signal is determined from the demand curves submitted; the NMP received by HEM nodes reflect the aggregated demand curve. As a result, different nodes are able to consume energy based on their own individual, time-varying preferences. This is a useful result.

Five, observe the impact at DSO Market and DSO Xfmr 1: there are reductions in the aggregated quantities consumed at certain points in time, although those nodes did not need to change their NMP. This is the tariff providing a differentiated price signal only to those nodes that impact the energy flow across the physical device being considered.

Six, observe the impact of the tariff on ps26 HEM. Notably, there is no impact; the prices and behavior of this node is completely unchanged. This is the desired result. ps26 HEM has no impact of the capacity of the linkage with the imposed tariff. This is also evidence of the DTDM being capable of providing differentiated price signals.

Finally, there is a fundamental consequence of the NMP for DSO Xfmr 4 being different from the NMP it communicates to its subnodes, ps490 HEM included. The price faced by subnodes is higher than the price faced by DSO Xfmr 4; this is "uplift". In Scenario 2.2, the capacity tariff collected no revenue. The uplift is a result of the NMP used to avoid the capacity tariff limits. In Scenario 2.3, the capacity tariff collected revenue for all quantities above 16 kW. However, in this case the DSO is the tariff owner; this is still uplift for DSO. The tariff serves its purpose: it avoided exceeding the component capacity limits. In doing so, it collected revenue.

This tariff avoidance and tariff collection uplift provides a new revenue stream for DSO business models. The uplift could be used by the DSO to fund their O&M. Or it could be distributed to customers as a dividend at the end of the month. Or it could be used to subsidize local renewable generation. Or the DSO may have set the tariff parameters with a revenue offset parameter, so the uplift was never collected in the first place. The DTDM does not dictate how these revenues are managed; that is a business model decision. Instead, the DTDM provides new opportunities for business models in the DDS environment.

7.4 Scenario 3: Ramp SMA Tariffs

7.4.1 Configuration

This scenario is designed to demonstrate the impact of adding a ramping tariff that used the simplified moving average (SMA). For this scenario, the prediction and elasticity values from Scenario 2 are used.

In Scenario 3.1, the base case is established. This uses the network shown in Scenario 1, with no tariffs.

In Scenario 3.2, a ramp SMA tariff is added to the DSO Market node. This uses a 20-minute moving average, with $0.005/ΔkW as the penalty for both positive and negative ramping. As a result, this tariff incentivizes steady energy flow.

In Scenario 3.3, the ramp SMA tariff parameters are increased. Now \$0.020/ΔkW is used as the penalty for both positive and negative ramping.

7.4.2 Results

The results are provided below. In addition to the bottom plots, the 20-minute SMA is shown for each node and scenario. Additionally, to make the illustrations more clear, each plot is also provided for at focused time period: from 14:00 to 19:00 on 2-June.

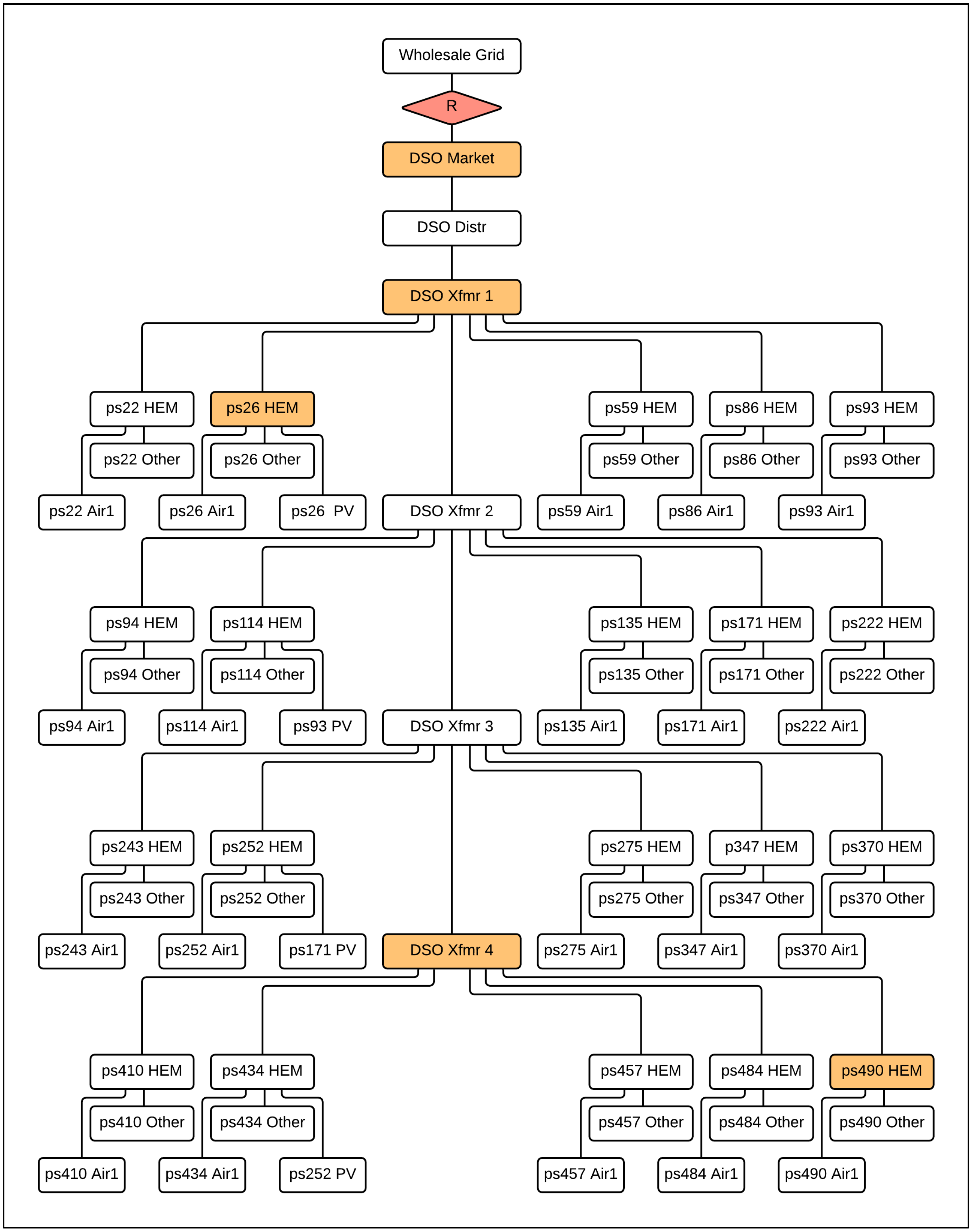
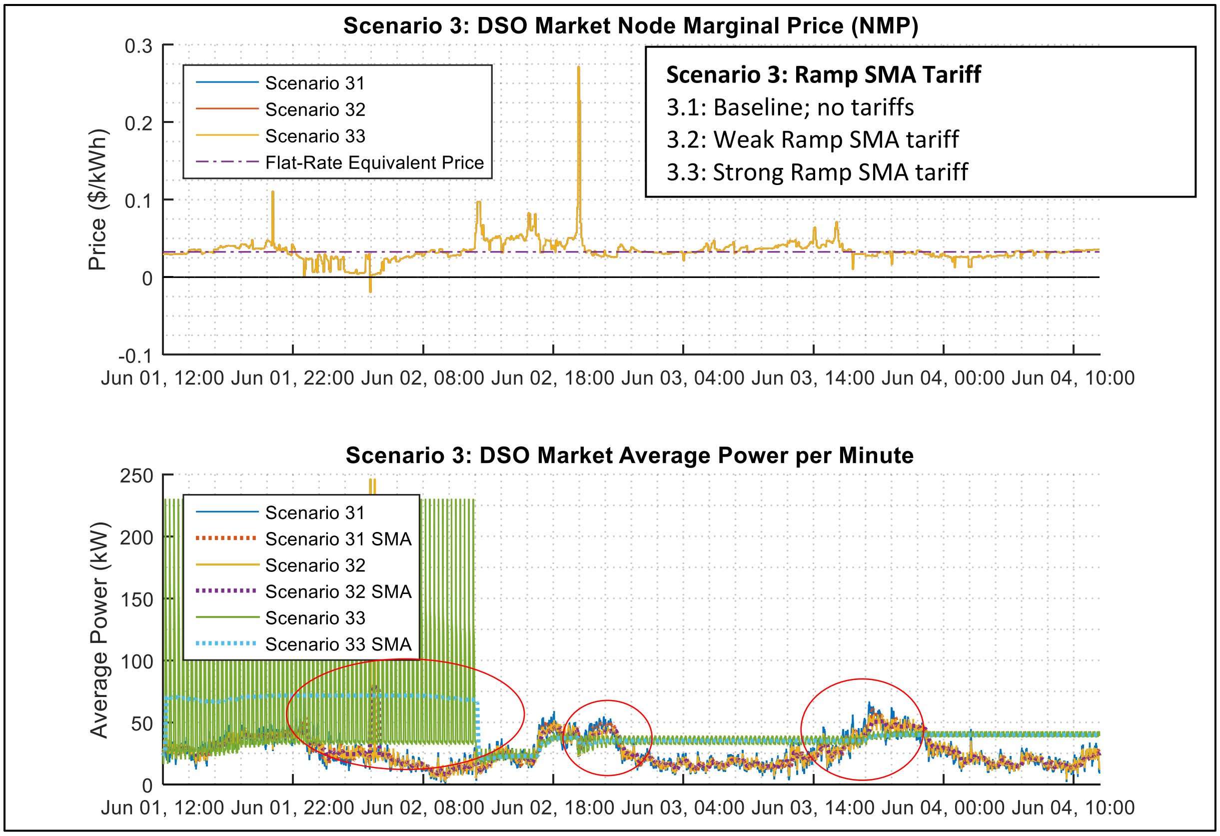
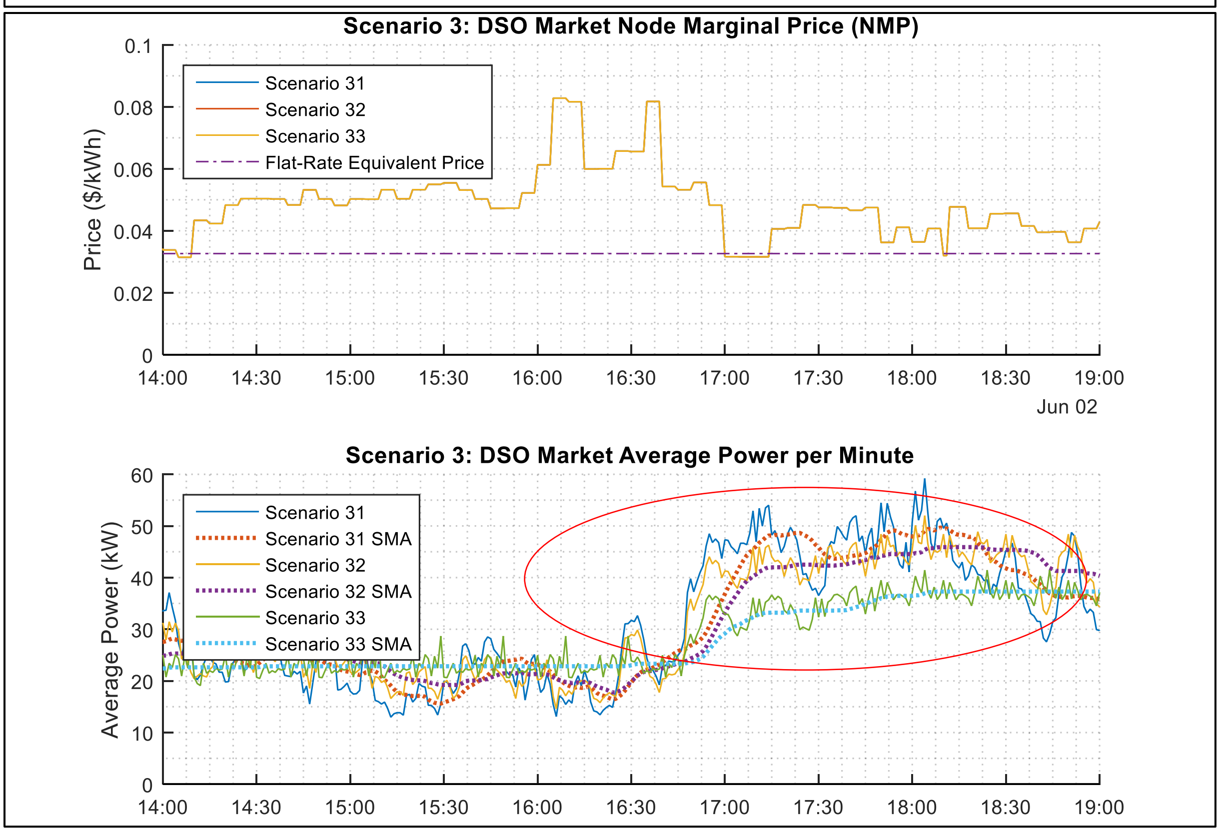


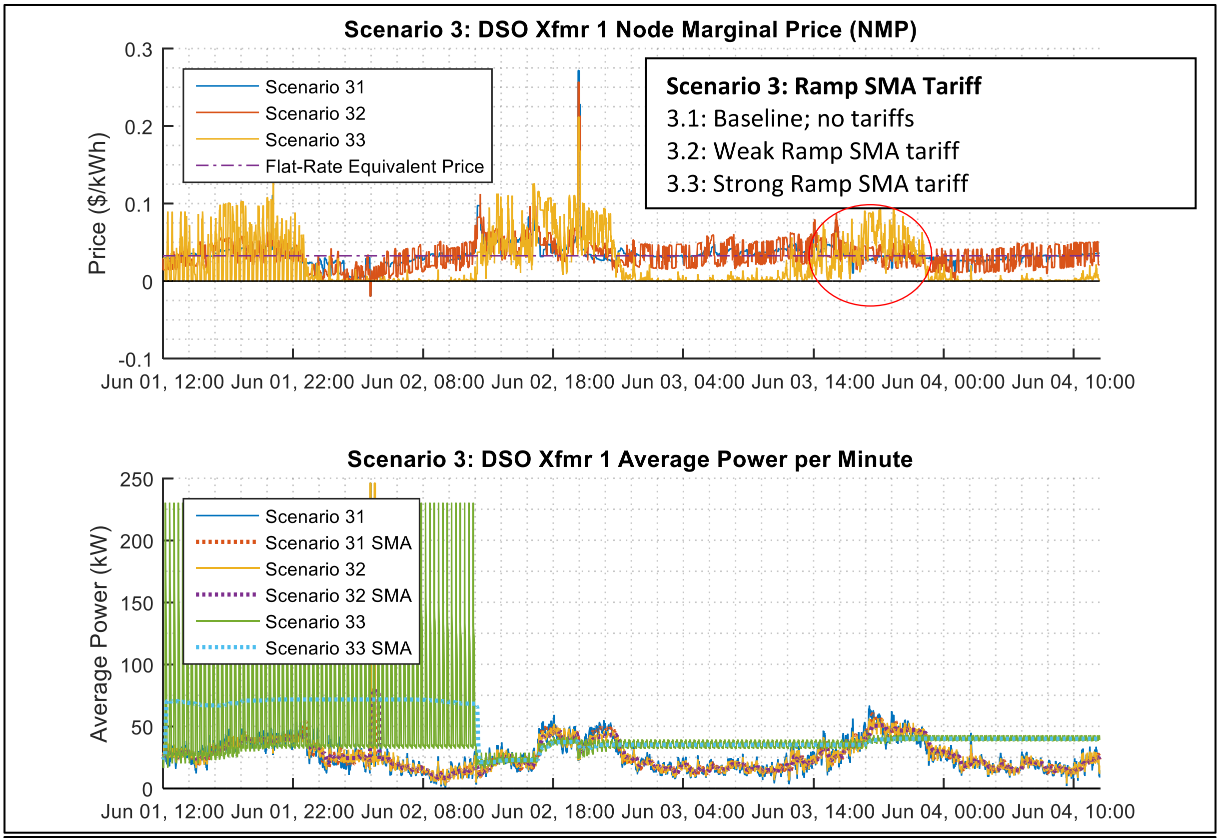
Figure 61: DTDM Network with Ramp Tariff (Scenarios 3.2, 3.3, 4.2)

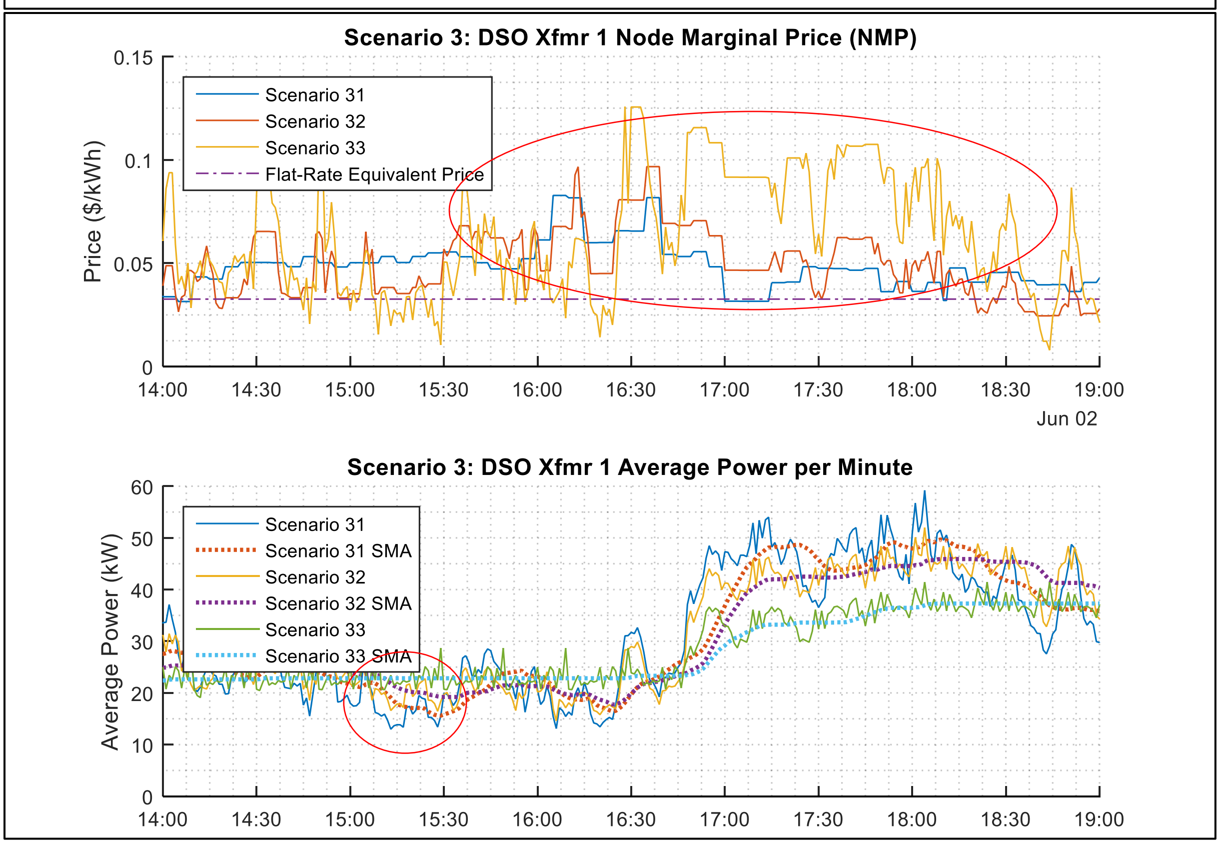


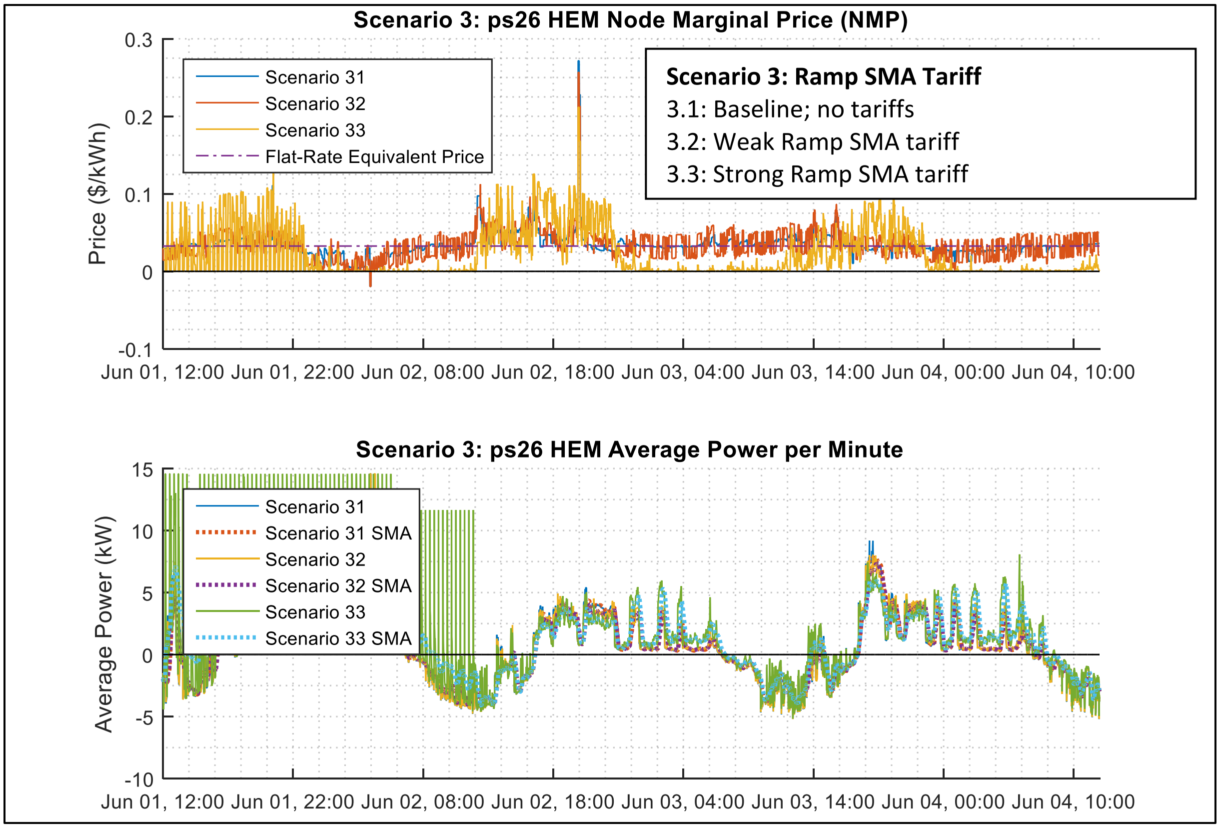
Scenario 3: DSO Market Node Marginal Price (NMP)
Scenario 31
Scenario 32
Scenario 33
Flat-Rate Equivalent Price
Scenario 3: Ramp SMA Tariff
3.1: Baseline; no tariffs
3.2: Weak Ramp SMA tariff
3.3: Strong Ramp SMA tariff
Price ($/kWh)
0.3
0.2
0.1
0
-0.1
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 3: DSO Market Average Power per Minute
Scenario 31
Scenario 31 SMA
Scenario 32
Scenario 32 SMA
Scenario 33
Scenario 33 SMA
Average Power (kW)
250
200
150
100
50
0
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00

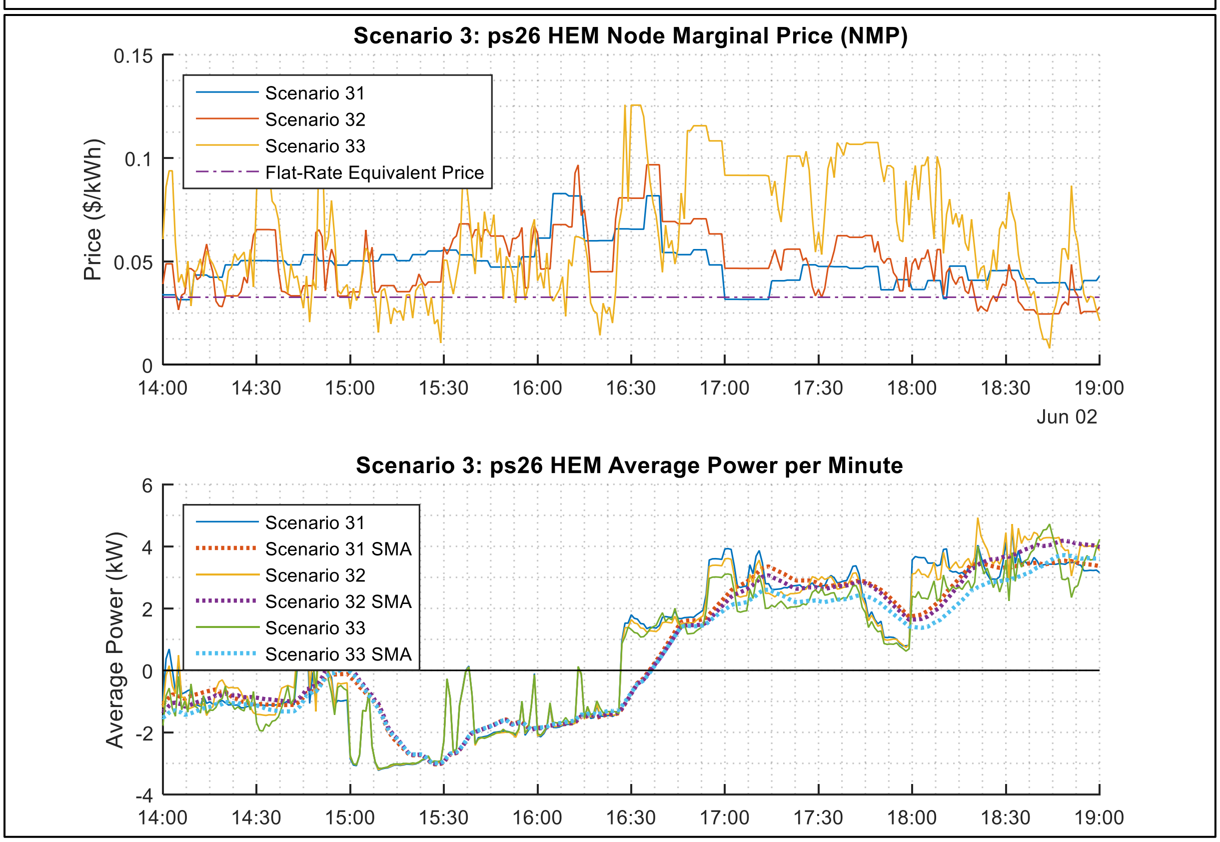


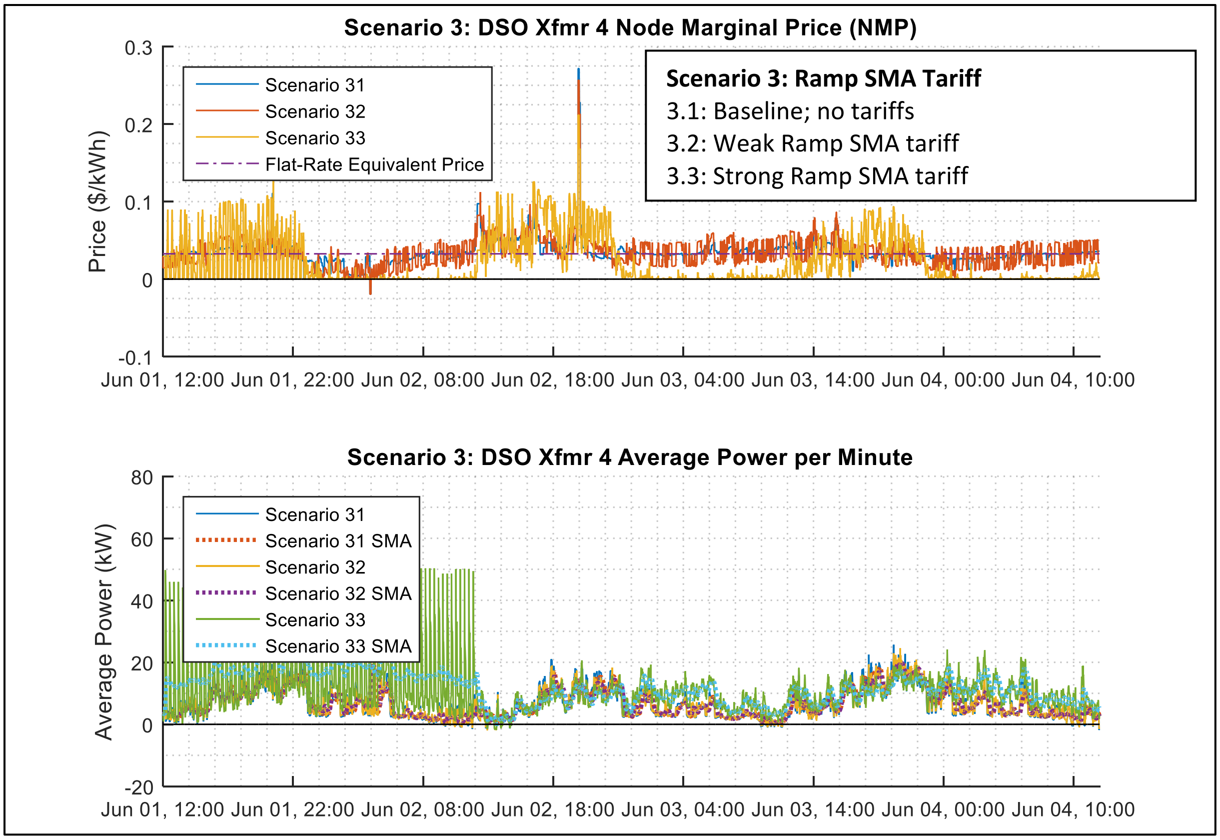
Scenario 3: DSO Market Node Marginal Price (NMP)
Scenario 31
Scenario 32
Scenario 33
Flat-Rate Equivalent Price
Price ($/kWh)
0.1
0.08
0.06
0.04
0.02
0
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00
Jun 02
Scenario 3: DSO Market Average Power per Minute
Scenario 31
Scenario 31 SMA
Scenario 32
Scenario 32 SMA
Scenario 33
Scenario 33 SMA
Average Power (kW)
60
50
40
30
20
10
0
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00


Scenario 3: DSO Xfmr 1 Node Marginal Price (NMP)
Scenario 31
Scenario 32
Scenario 33
Flat-Rate Equivalent Price
Scenario 3: Ramp SMA Tariff
3.1: Baseline; no tariffs
3.2: Weak Ramp SMA tariff
3.3: Strong Ramp SMA tariff
Price ($/kWh)
0.3
0.2
0.1
0
-0.1
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 3: DSO Xfmr 1 Average Power per Minute
Scenario 31
Scenario 31 SMA
Scenario 32
Scenario 32 SMA
Scenario 33
Scenario 33 SMA
Average Power (kW)
250
200
150
100
50
0
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00

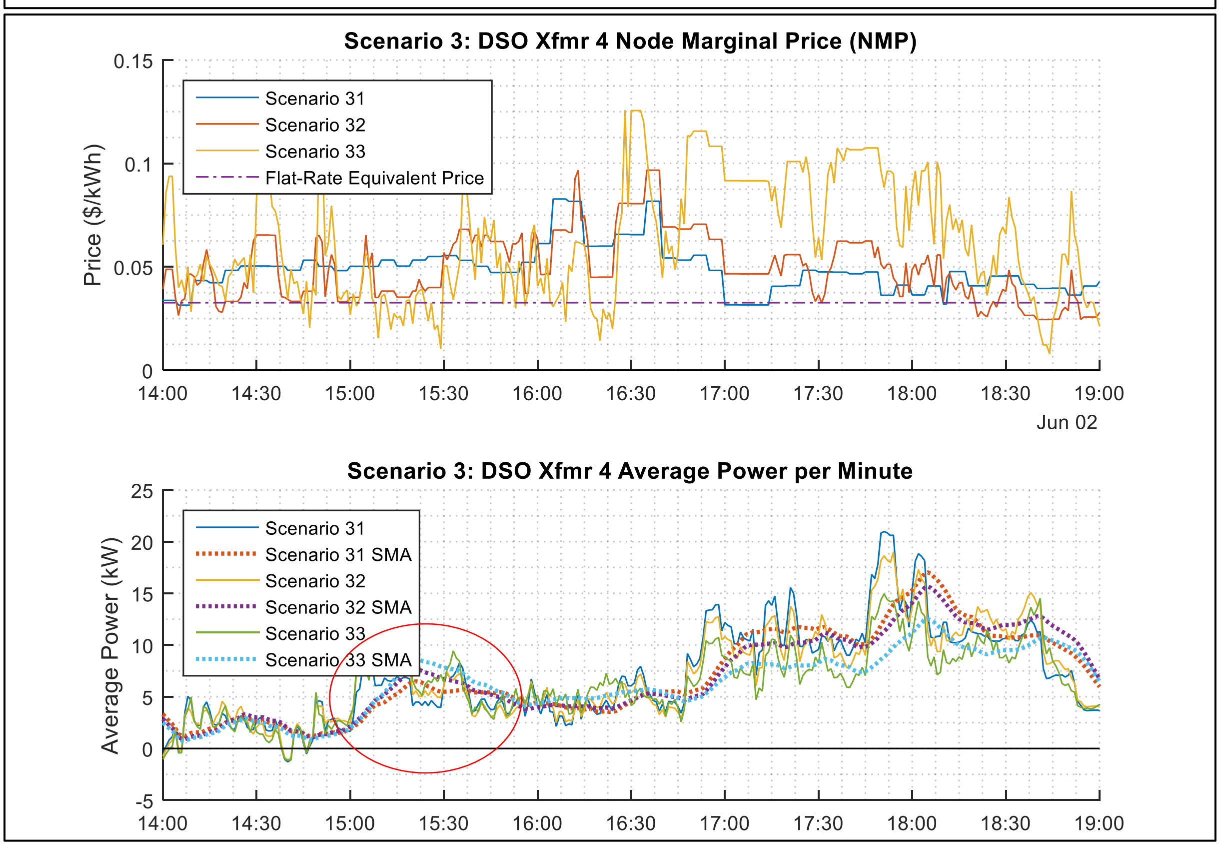

Scenario 3: DSO Xfmr 1 Node Marginal Price (NMP)
Scenario 31
Scenario 32
Scenario 33
Flat-Rate Equivalent Price
Price ($/kWh)
0.15
0.1
0.05
0
14:00
14:30
15:00
15:30
16:00
16:30
17:00
17:30
18:00
18:30
19:00
Jun 02
Scenario 3: DSO Xfmr 1 Average Power per Minute
Scenario 31
Scenario 31 SMA
Scenario 32
Scenario 32 SMA
Scenario 33
Scenario 33 SMA
Average Power (kW)
60
50
40
30
20
10
0
14:00
14:30
15:00
15:30
16:00
16:30
17:00
17:30
18:00
18:30
19:00


Scenario 3: ps26 HEM Node Marginal Price (NMP)
Price ($/kWh)
Scenario 31
Scenario 32
Scenario 33
Flat-Rate Equivalent Price
Scenario 3: Ramp SMA Tariff
3.1: Baseline; no tariffs
3.2: Weak Ramp SMA tariff
3.3: Strong Ramp SMA tariff
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 3: ps26 HEM Average Power per Minute
Average Power (kW)
Scenario 31
Scenario 31 SMA
Scenario 32
Scenario 32 SMA
Scenario 33
Scenario 33 SMA
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00


Scenario 3: ps26 HEM Node Marginal Price (NMP)
Price ($/kWh)
Scenario 31
Scenario 32
Scenario 33
Flat-Rate Equivalent Price
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00
Jun 02
Scenario 3: ps26 HEM Average Power per Minute
Average Power (kW)
Scenario 31
Scenario 31 SMA
Scenario 32
Scenario 32 SMA
Scenario 33
Scenario 33 SMA
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00



Scenario 3: DSO Xfmr 4 Node Marginal Price (NMP)
Price ($/kWh)
Scenario 31
Scenario 32
Scenario 33
Flat-Rate Equivalent Price
Scenario 3: Ramp SMA Tariff
3.1: Baseline; no tariffs
3.2: Weak Ramp SMA tariff
3.3: Strong Ramp SMA tariff
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 3: DSO Xfmr 4 Average Power per Minute
Average Power (kW)
Scenario 31
Scenario 31 SMA
Scenario 32
Scenario 32 SMA
Scenario 33
Scenario 33 SMA
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00



Scenario 3: DSO Xfmr 4 Node Marginal Price (NMP)
Price ($/kWh)
Scenario 31
Scenario 32
Scenario 33
Flat-Rate Equivalent Price
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00
Jun 02
Scenario 3: DSO Xfmr 4 Average Power per Minute
Average Power (kW)
Scenario 31
Scenario 31 SMA
Scenario 32
Scenario 32 SMA
Scenario 33
Scenario 33 SMA
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00

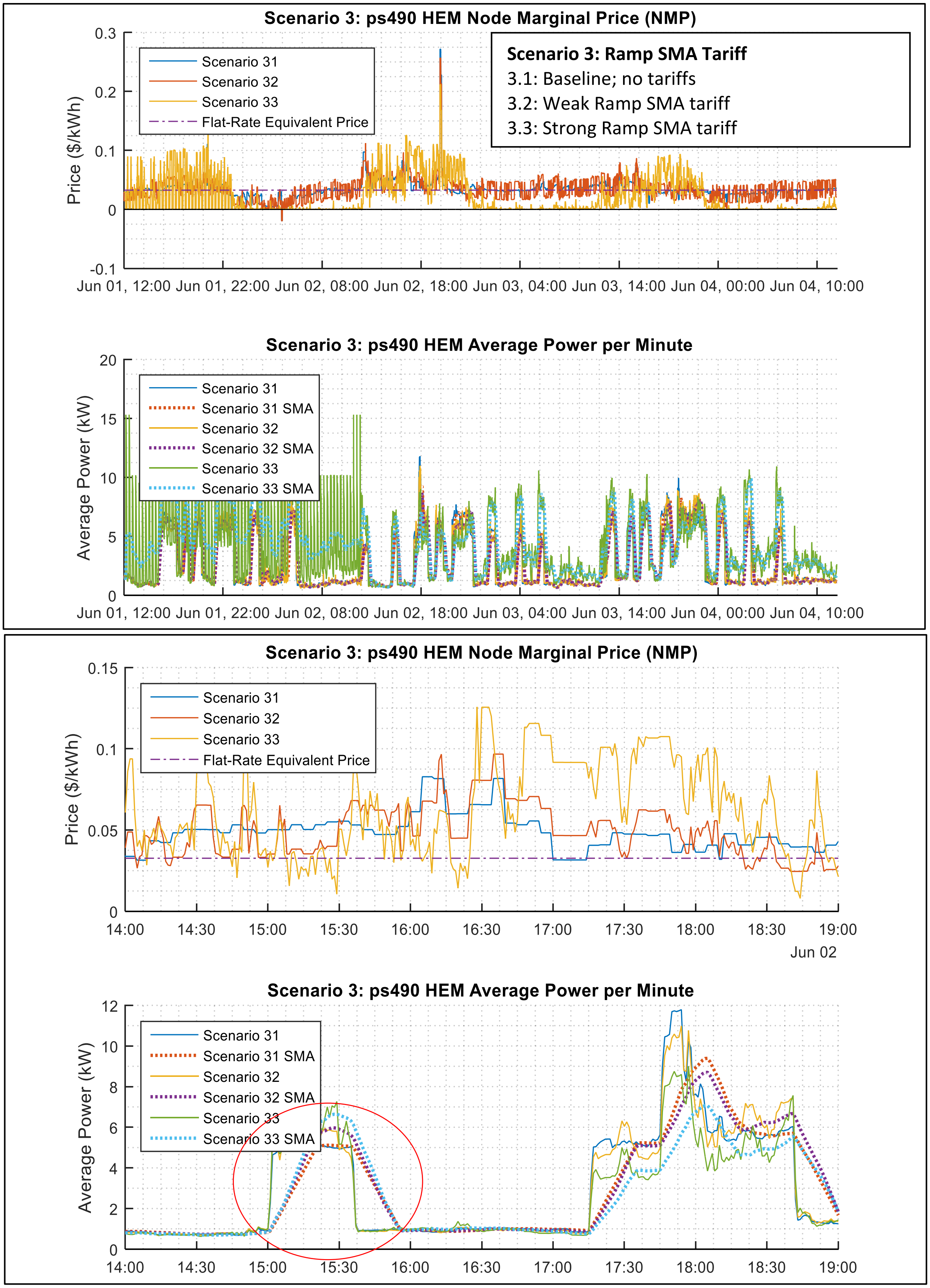


Figure 62: Scenario 3 Results

7.4.3 Observations

There are numerous observations to be made from these results. These observations are highlighted with red circles in Figure 62 above.

One, the tariff provides the intended effect. Simply be inspecting the moving average at DSO Market, it is clear that Scenario 3.2 provides less ramping than the baseline established in Scenario 3.1. Additionally, with a stronger tariff, Scenario 3.3 provides a more pronounced impact. Because the tariff penalizes both positive and negative ramping, the overall consumption profile is flattened. Different tariff parameters would be expected to have different results.

Two, ramp mitigation is accomplished by modifying the NMP sent by DSO Market to its subnode; this is observed in the DSO Xfmr 1 price plot. In Scenario 3.2, the price oscillates around the baseline wholesale energy price. In Scenario 3.3 the price oscillations are even larger than in Scenario 3.2; this is due to the larger tariff price parameters. Notice, with no other tariffs in the system, the NMP at DSO Xfmr 1 matches those at all nodes within its subsystem (i.e. below DSO Xfmr 1 in the network). In fact, these prices are differ very slightly, due to loss translation.

Three, the tariff is imposed only on DSO Market. The market action at DSO Market passes along price signals to nodes within the system, to limit the overall system ramp rate. This does not necessarily provide a reduced ramp rate at all nodes in the system. Observe ps490 HEM. From 15:00 to 15:30 on 2-June, this node actually increased its ramp rate with the higher system tariff parameters (i.e. at 15:30, the SMA for Scenario 3.3 has risen higher than the SMA for Scenario 3.2, which has risen higher than then SMA for Scenario 3.1). The same effect is seen at DSO Xfmr 4.

The driver for this effect is provided by examining the plots for DSO Xfmr 1. During that time period, the system sets out to avoid negative ramping. As a result, the market clearing price (the NMP at DSO Xfmr 1) falls below the baseline FRE price during the period. This is an extremely useful result of the DTDM and differentiated price signals. The system as a whole wanted to promote a certain response (avoid negative ramping), which was reflected in the price. However, this price impacts different nodes in unique ways. For ps490 HEM, the price encouraged higher energy consumption, which actually increased the node's local ramp rate. Yet ramp rate at ps490 is not a concern; the only concern is ramp rate at the wholesale grid. This provides a potential win-win for actors in the system: the wholesale grid sees reduced ramp rates; the DSO collects uplift from the ramping tariffs, to be utilized as described in Section 7.3.3; and customers such as ps490 HEM are able to consume energy at a lower effective price. Certainly the detailed cost-benefit analysis would be more nuanced and longer-term. However, this case study demonstrates the fundamental opportunities and value propositions.

Finally, this scenario demonstrates the cycling risk of the SMA-type ramp tariff. As described in Section 6.8, the SMA equation essentially establishes a target quantity tariff, based on the previously recorded quantity that "drops off" from the SMA calculation. This can lead to cycling, as seen at DSO Market for the first 20 hours of simulation. With the strong ramp tariff parameters, the price signals incentivize a system response that oscillates wildly, to keep a constant SMA. This is certainly an unfavorable response. However, after 08:00 on 2-June, this effect is reduced. This indicates that the oscillations are driven by the system initial conditions; once the system preferences became "strong enough" to break the SMA cycle, the system provides more favorable stability.

However, starting at 22:00 on 2-June, the system response for Scenario 3.3 provides a flat SMA. Again, the actual quantities oscillate around this value, with a much smaller deviations. This appears to be a better result, yet it indicates continued system risk. Specifically, if a large, but temporary, preference swing occurs, the system demand will have a temporary “jump”. This “jump” is a deviation from the SMA, and will likely repeat at the frequency of the SMA period. This is because, when the “jump” drops out of the SMA, the same quantity will be incentivized; this process will repeats itself every SMA period.

As described in Section 6.8 and Section 6.9, an alternative to the SMA-calculated ramp tariff is the Exponential Moving Average (EMA), in which historical values are discounted when determining the moving average. This will be used for Scenario 4 and 5.

7.5 Scenario 4: Ramp EMA Tariff Location

7.5.1 Configuration

This scenario is designed to demonstrate the impact of adding a ramping tariff that used the exponential moving average (EMA) at different locations in the network. By comparing the results, observations can be made on the impact of imposing tariffs in a concentrated versus distributed fashion.

For this scenario, the prediction and elasticity values from Scenario 2 are used.

In Scenario 4.1, the base case is established. This uses the network shown in Scenario 1, with no tariffs.

In Scenario 4.2, a ramp EMA tariff is imposed on the DSO Market node. The network configuration matches that illustrated in Section 7.4.1. This ramp EMA tariff uses N = 20 minutes for the EMA calculation, with \$0.005/ΔkW as the penalty for both positive and negative ramping. As a result, this tariff incentivizes steady energy flow.

In Scenario 4.3, a ramp EMA tariff is imposed on all household HEM nodes. There is no longer a ramp EMA tariff imposed on the DSO Market. The network configuration is illustrated below. The parameters are the same as the tariff in Scenario 4.2.

7.5.2 Results

The results of this Scenario are provided below. As in Scenario 3, the SMA of measured power flow is displayed for each node. Note, this is the SMA of the measured quantities, although the tariff is computed using the EMA. The SMA is still used in the plots to improve comparisons to Scenario 3.

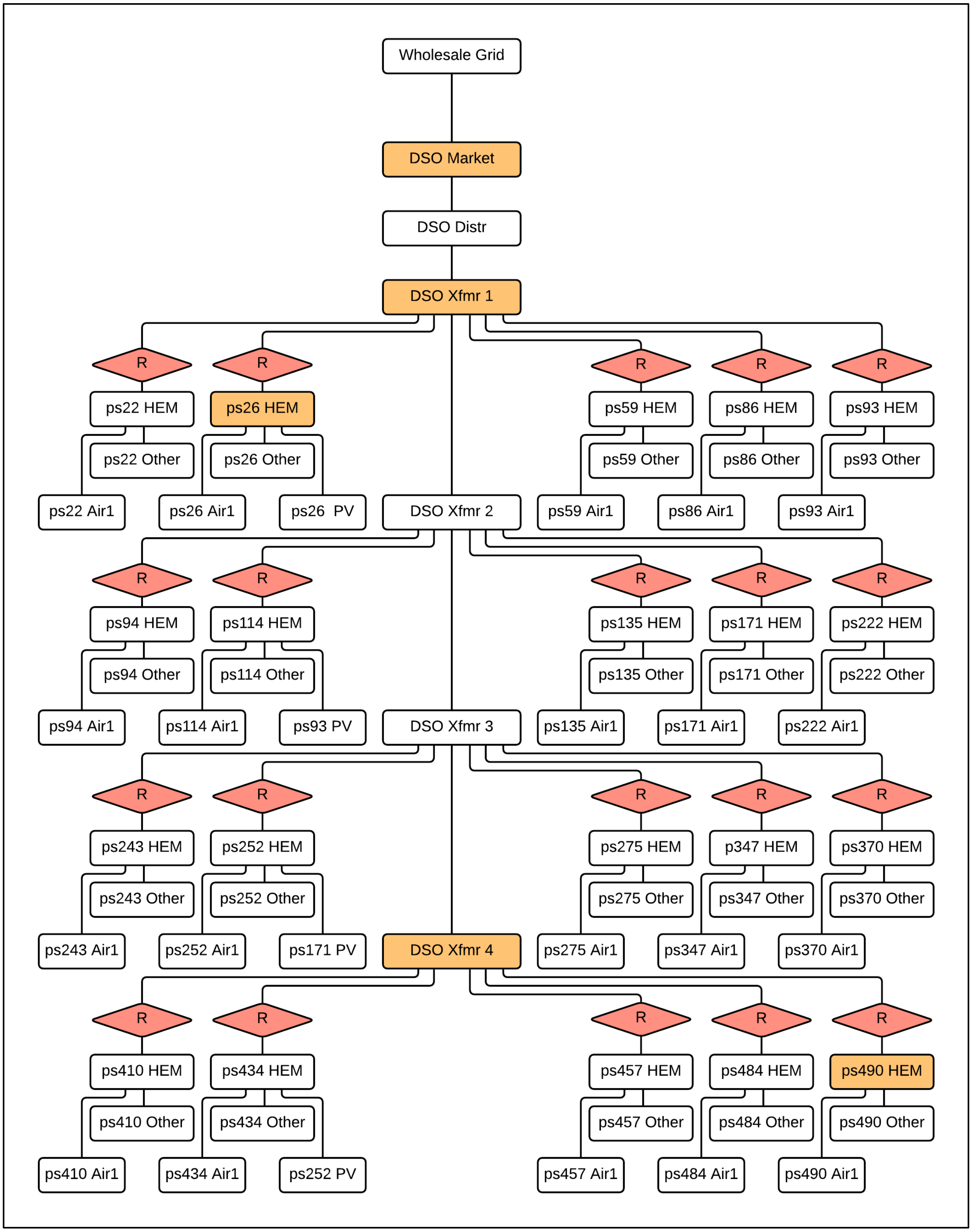
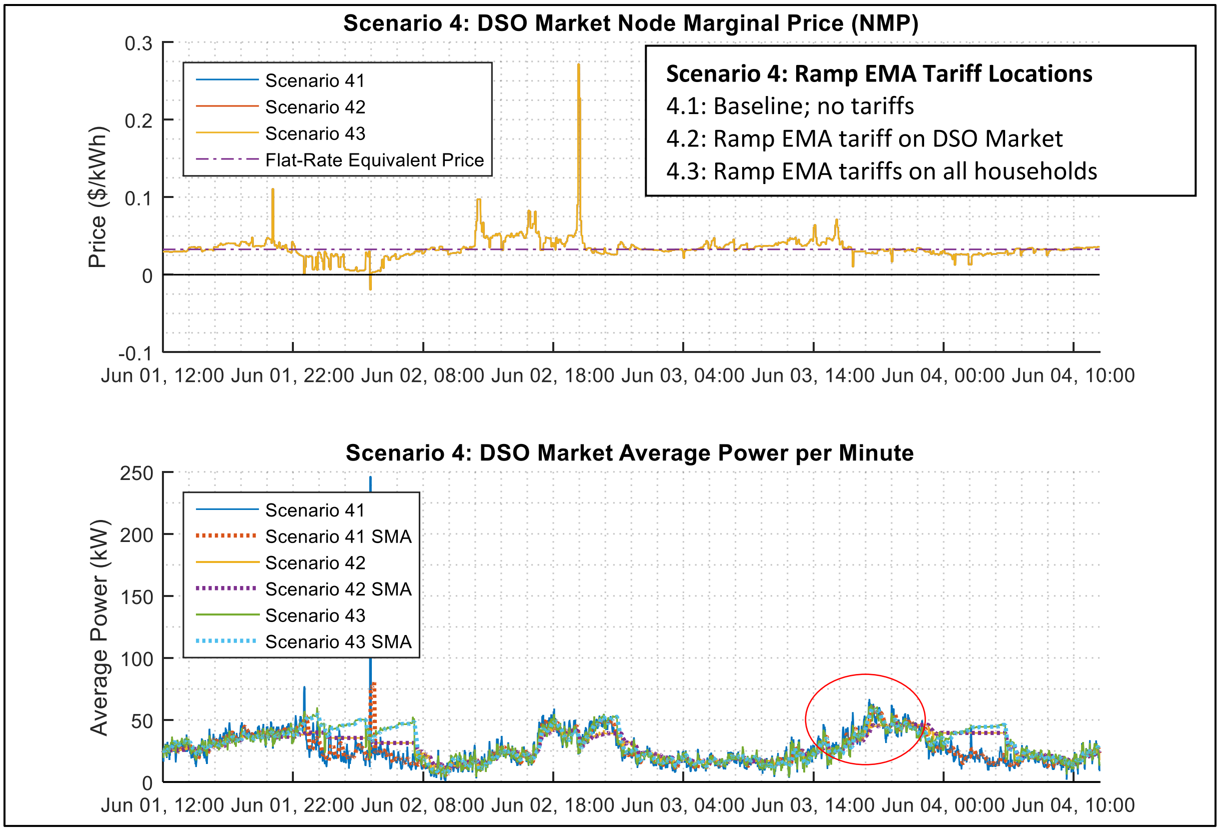
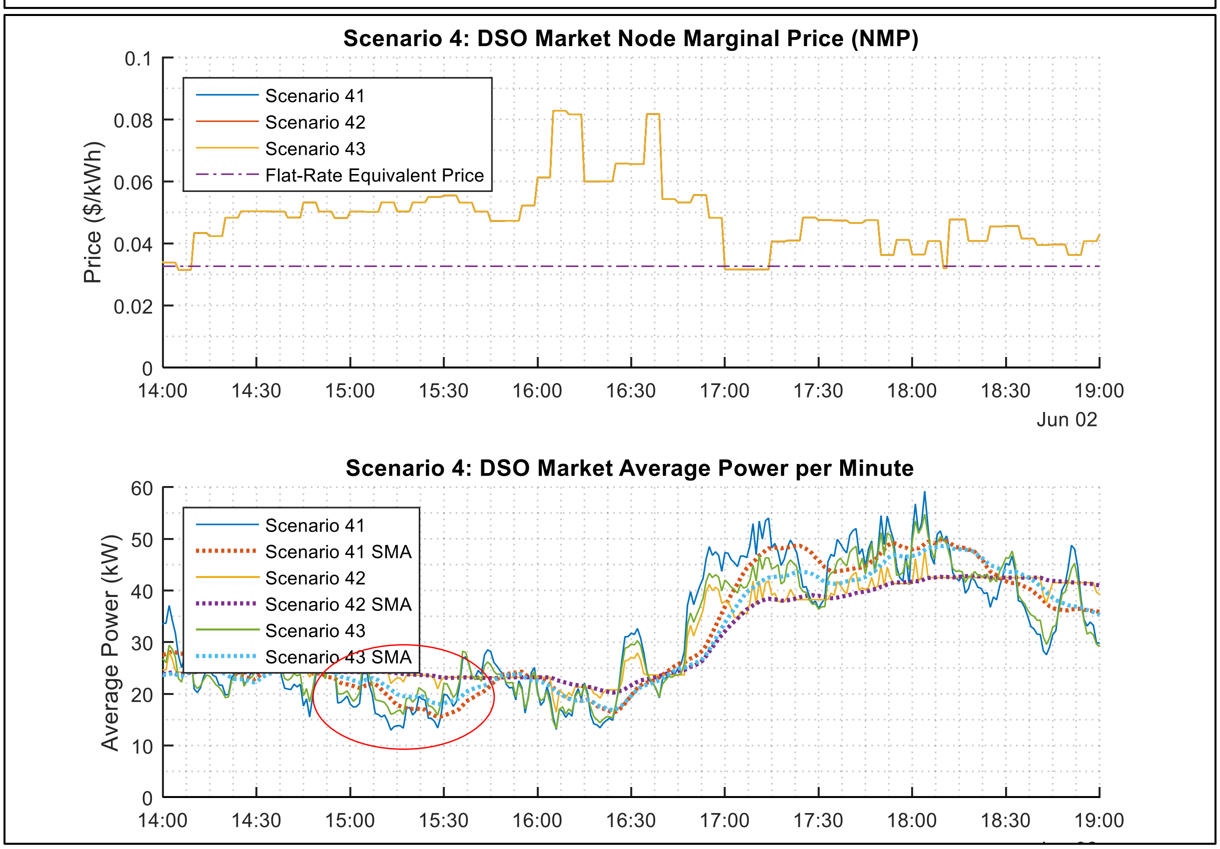


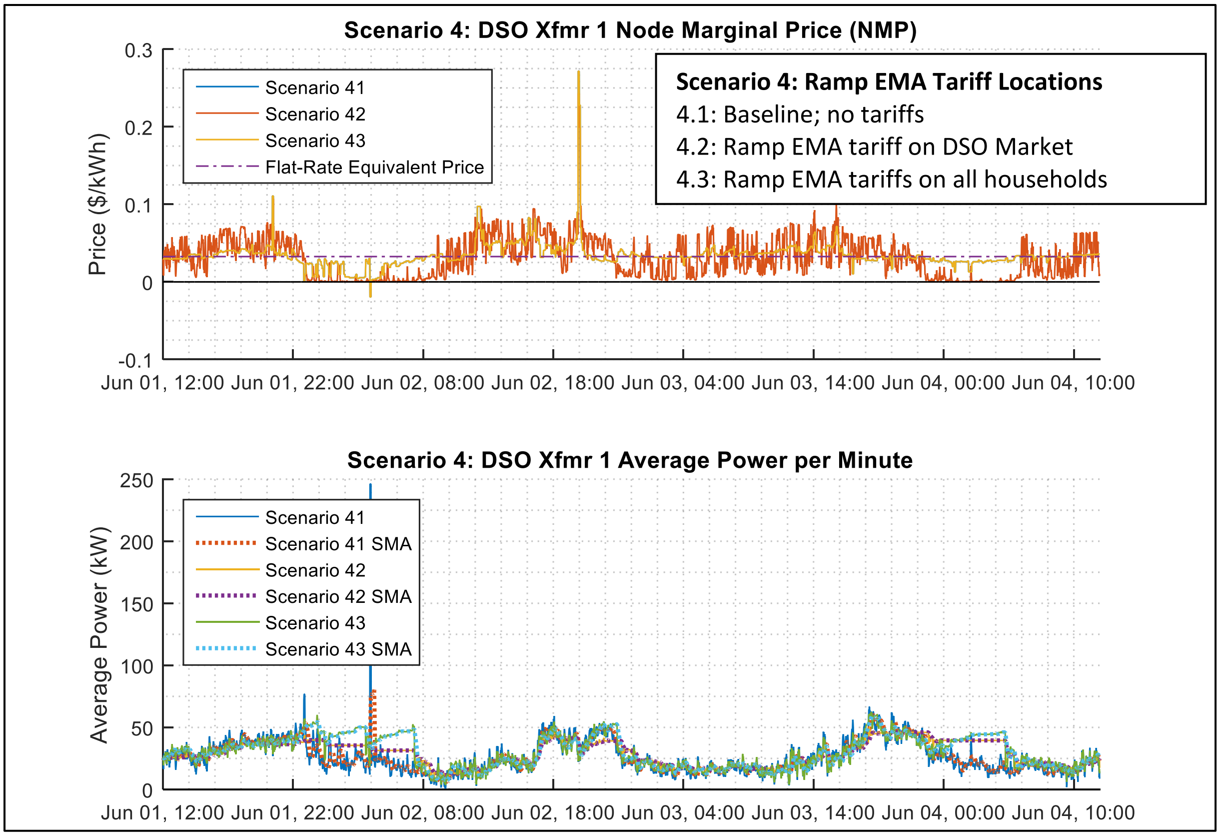
Figure 63: DTDM Network with Distributed Ramp Tariffs (Scenario 4.3)

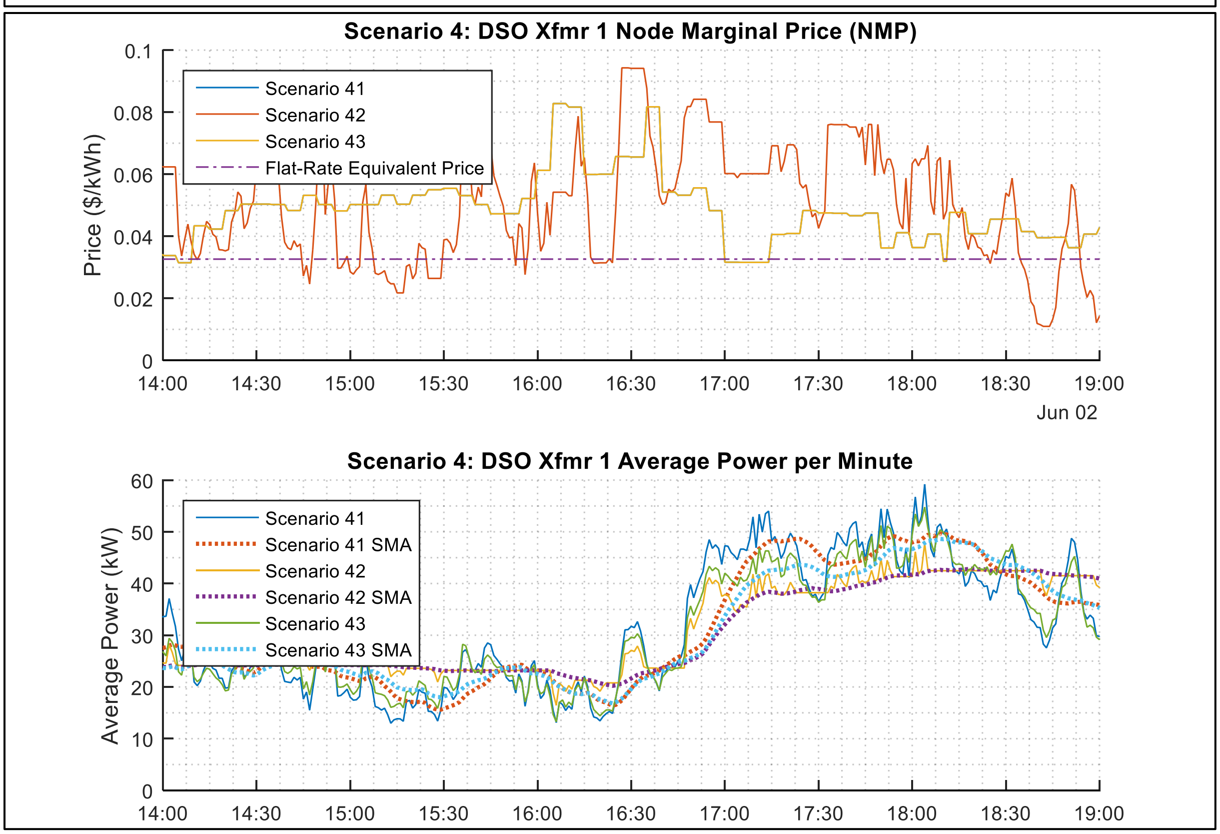


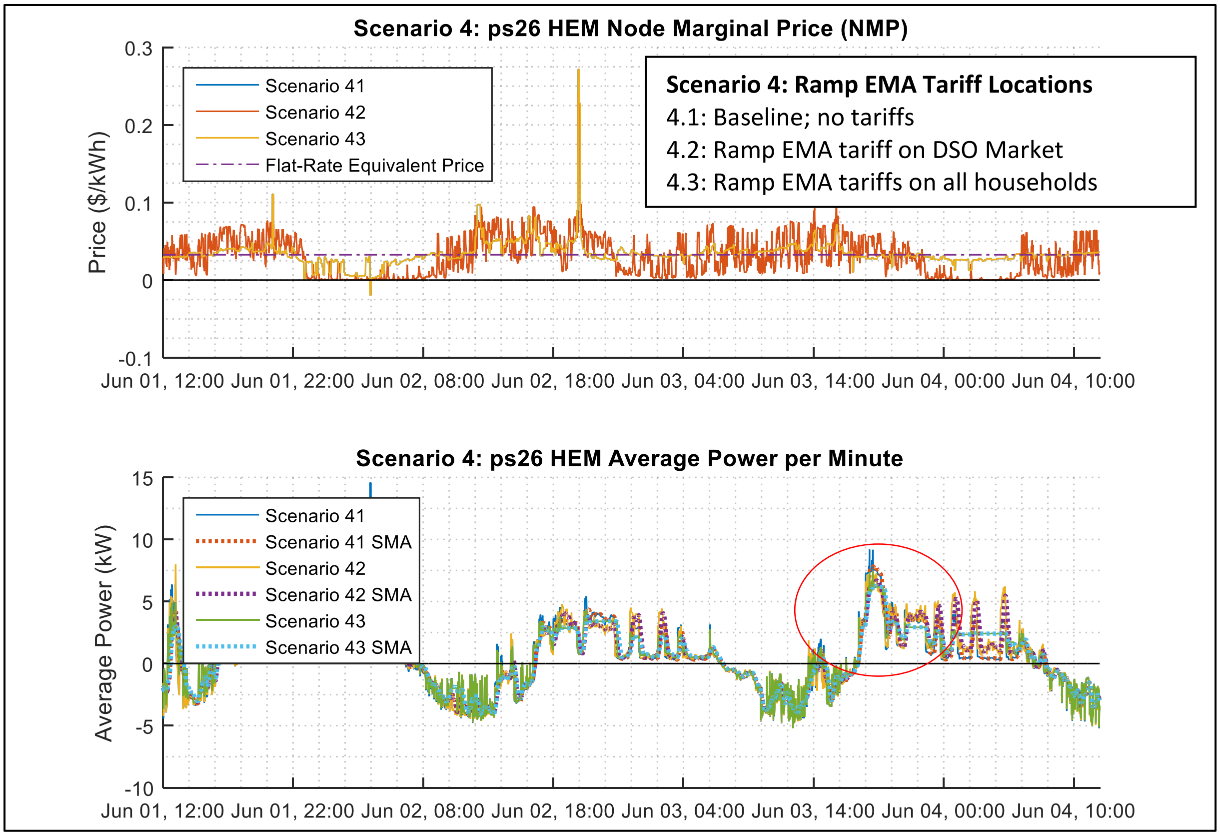
Scenario 4: DSO Market Node Marginal Price (NMP)
Scenario 4: Ramp EMA Tariff Locations
4.1: Baseline; no tariffs
4.2: Ramp EMA tariff on DSO Market
4.3: Ramp EMA tariffs on all households
Price ($/kWh)
Scenario 41
Scenario 42
Scenario 43
Flat-Rate Equivalent Price
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 4: DSO Market Average Power per Minute
Average Power (kW)
Scenario 41
Scenario 41 SMA
Scenario 42
Scenario 42 SMA
Scenario 43
Scenario 43 SMA
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00

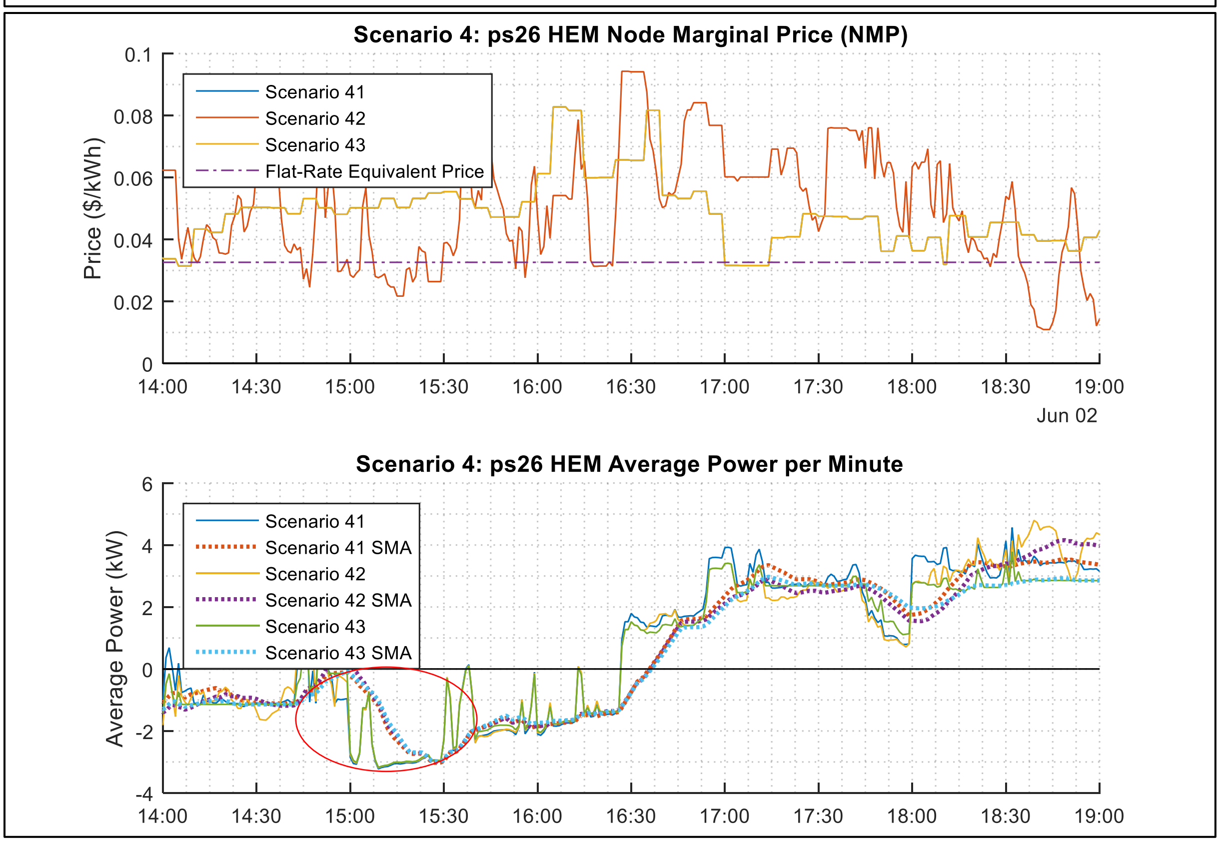


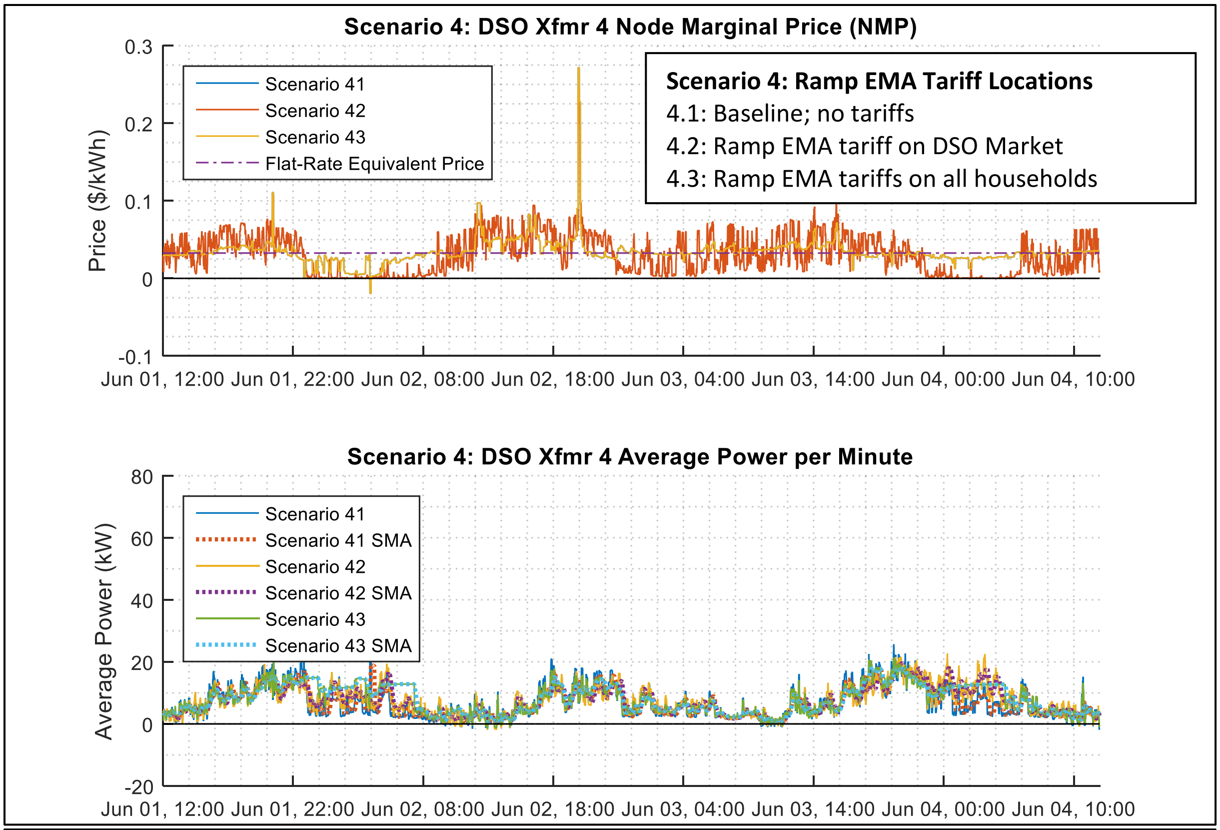
Scenario 4: DSO Market Node Marginal Price (NMP)
Price ($/kWh)
Scenario 41
Scenario 42
Scenario 43
Flat-Rate Equivalent Price
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00
Jun 02
Scenario 4: DSO Market Average Power per Minute
Average Power (kW)
Scenario 41
Scenario 41 SMA
Scenario 42
Scenario 42 SMA
Scenario 43
Scenario 43 SMA
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00

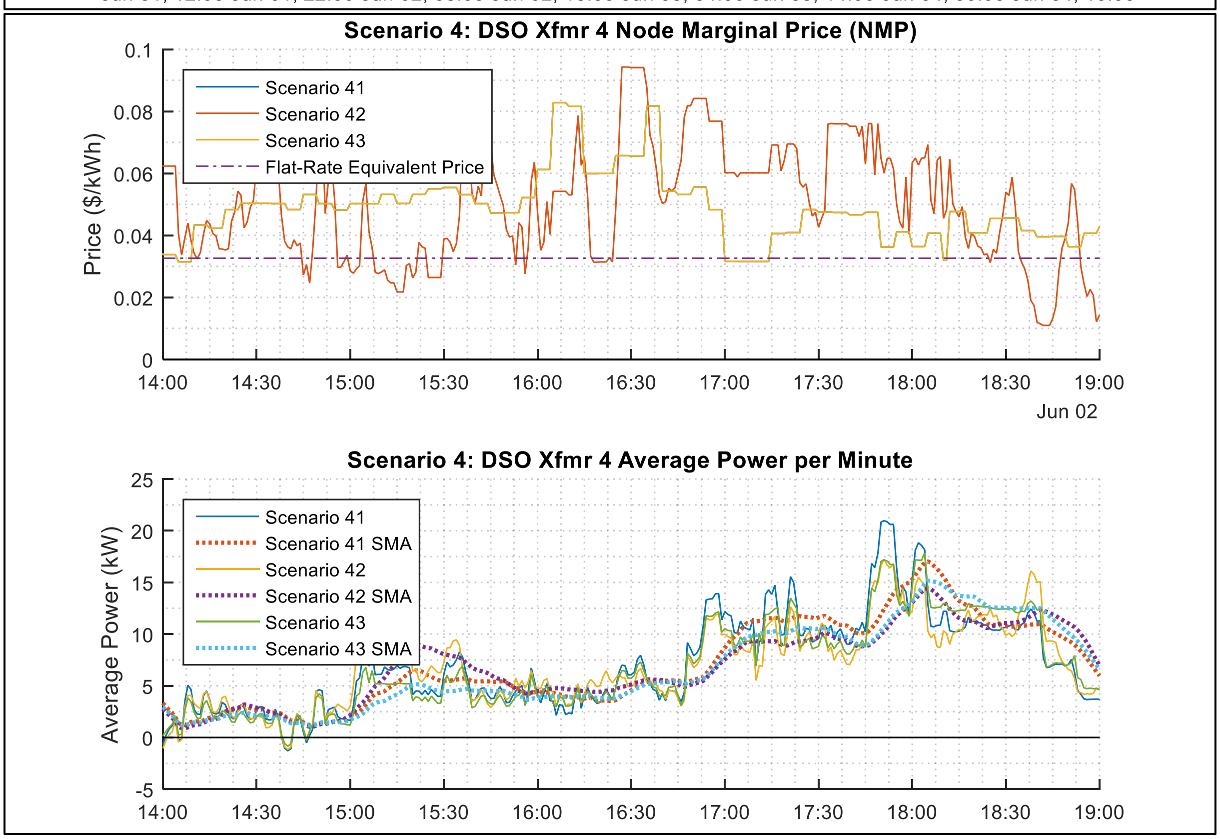

Scenario 4: DSO Xfmr 1 Node Marginal Price (NMP)
Price ($/kWh)
Scenario 41
Scenario 42
Scenario 43
Flat-Rate Equivalent Price
Scenario 4: Ramp EMA Tariff Locations
4.1: Baseline; no tariffs
4.2: Ramp EMA tariff on DSO Market
4.3: Ramp EMA tariffs on all households
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 4: DSO Xfmr 1 Average Power per Minute
Average Power (kW)
Scenario 41
Scenario 41 SMA
Scenario 42
Scenario 42 SMA
Scenario 43
Scenario 43 SMA
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00


Scenario 4: DSO Xfmr 1 Node Marginal Price (NMP)
Price ($/kWh)
Scenario 41
Scenario 42
Scenario 43
Flat-Rate Equivalent Price
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00
Jun 02
Scenario 4: DSO Xfmr 1 Average Power per Minute
Average Power (kW)
Scenario 41
Scenario 41 SMA
Scenario 42
Scenario 42 SMA
Scenario 43
Scenario 43 SMA
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00


Scenario 4: ps26 HEM Node Marginal Price (NMP)
Price ($/kWh)
Scenario 41
Scenario 42
Scenario 43
Flat-Rate Equivalent Price
Scenario 4: Ramp EMA Tariff Locations
4.1: Baseline; no tariffs
4.2: Ramp EMA tariff on DSO Market
4.3: Ramp EMA tariffs on all households
0.3
0.2
0.1
0
-0.1
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 4: ps26 HEM Average Power per Minute
Average Power (kW)
Scenario 41
Scenario 41 SMA
Scenario 42
Scenario 42 SMA
Scenario 43
Scenario 43 SMA
15
10
5
0
-5
-10
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00


Scenario 4: ps26 HEM Node Marginal Price (NMP)
Price ($/kWh)
Scenario 41
Scenario 42
Scenario 43
Flat-Rate Equivalent Price
0.1
0.08
0.06
0.04
0.02
0
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00
Jun 02
Scenario 4: ps26 HEM Average Power per Minute
Average Power (kW)
Scenario 41
Scenario 41 SMA
Scenario 42
Scenario 42 SMA
Scenario 43
Scenario 43 SMA
6
4
2
0
-2
-4
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00


Scenario 4: DSO Xfmr 4 Node Marginal Price (NMP)
Scenario 41
Scenario 42
Scenario 43
Flat-Rate Equivalent Price
Scenario 4: Ramp EMA Tariff Locations
4.1: Baseline; no tariffs
4.2: Ramp EMA tariff on DSO Market
4.3: Ramp EMA tariffs on all households
Price ($/kWh)
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 4: DSO Xfmr 4 Average Power per Minute
Scenario 41
Scenario 41 SMA
Scenario 42
Scenario 42 SMA
Scenario 43
Scenario 43 SMA
Average Power (kW)
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00


Scenario 4: DSO Xfmr 4 Node Marginal Price (NMP)
Scenario 41
Scenario 42
Scenario 43
Flat-Rate Equivalent Price
Price ($/kWh)
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00
Jun 02
Scenario 4: DSO Xfmr 4 Average Power per Minute
Scenario 41
Scenario 41 SMA
Scenario 42
Scenario 42 SMA
Scenario 43
Scenario 43 SMA
Average Power (kW)
14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30 18:00 18:30 19:00

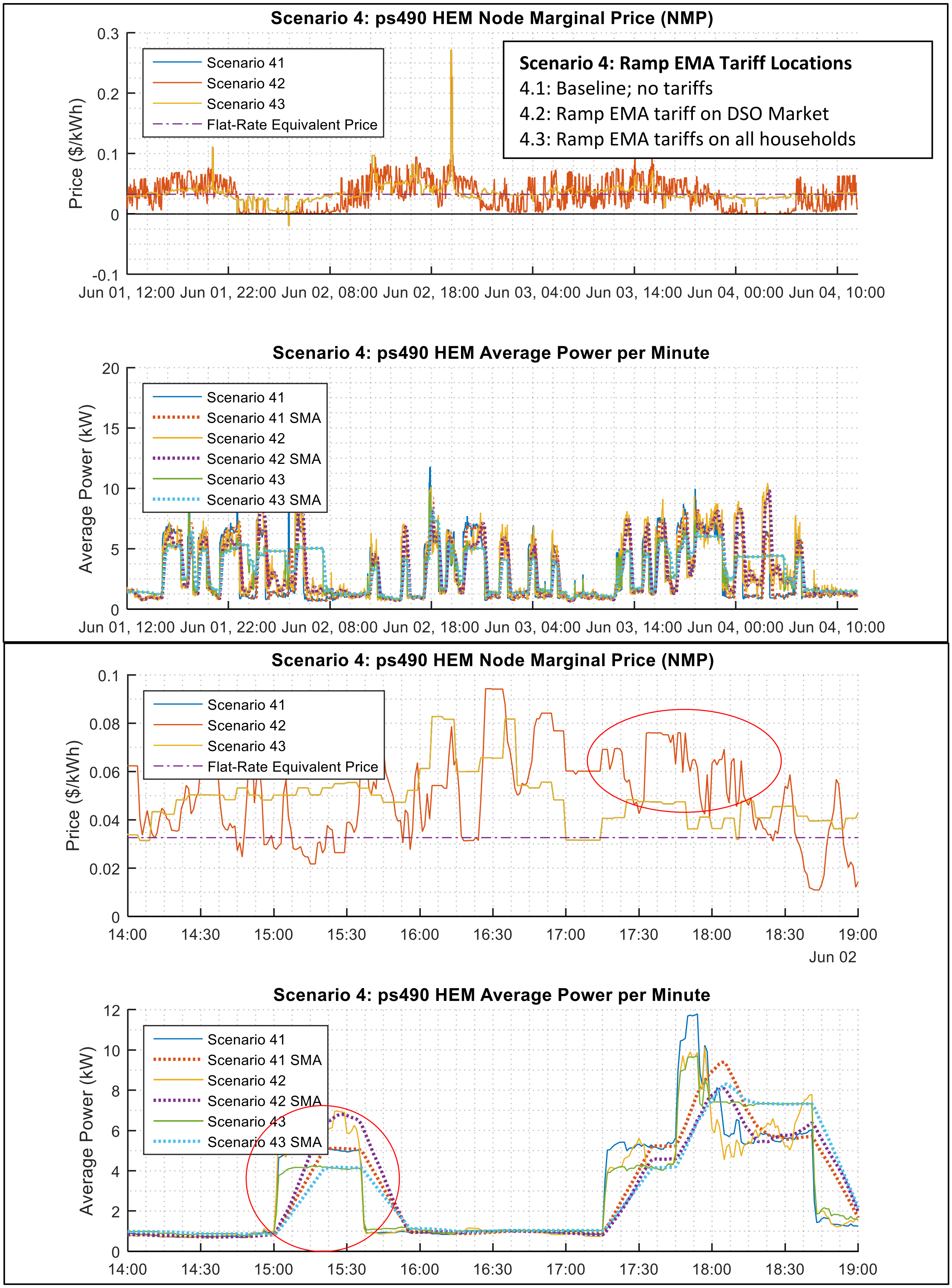


Figure 64: Scenario 4 Results

7.5.3 Observations

There are numerous observations to be made from these results. These observations are highlighted with red circle in Figure 64 above.

One, the location of the tariffs impacts the local and system-wide effect. This occurs even when tariffs in both Scenario 4.2 and 4.3 use the same ramp penalties parameters. Additionally, observe that the households in both scenarios only have one “upstream” tariff; there is no double-collection by tariffs in either scenario.

In Scenario 4.2, the tariff is concentrated at DSO Market. As seen in Scenario 3, this may or may not incentivize constant energy consumption at each individual node. This is seen from 15:00 to 15:30 for both ps26 HEM and ps490 HEM. However, this tariff is successful at reducing system-wide ramping, as measured at DSO Market.

In Scenario 4.3, the tariffs is distributed at each HEM node. Now both ps26 HEM and ps490 HEM can be seen to limit their local ramping at all time periods, including 15:00 to 15:30. Because the tariff is configured to penalize both positive and negative ramping, this results in both higher and lower consumption than the baseline quantities, depending on the current EMA. However, this does not translate directly to ramping reduction at DSO Market. Inspection of the results provides a smoother consumption curve than the baseline case, but less smooth than that in Scenario 4.2.

This provides an important observation and conclusion. Tariffs should be imposed where the controlled externality occurs. If the concern is ramping at the wholesale grid connection, that should be the location of the tariff, like in Scenario 4.2. Distributing the tariffs will still result in a system response, but the impact may not be as strong. This does not preclude business case reasons to distribute tariffs. Yet it does illustrate the downside to such an approach: customers will modify their behavior in ways that may be unnecessary or even counterproductive to system goals.

Two, each HEM node determines its own clearing price, beyond the NMP provided by DSO. In Scenario 4.3, the NMP provided to each HEM node is nearly identical that provided in Scenario 4.1. This is because the system above the HEM is identical, but each HEM provides a slightly different demand curve submission during Market Operation. Thus, in Scenario 4.3, the NMP at each node is still controlled by the wholesale energy price, with only very slight adjustment due to the impact of the modified consumption on the loss translation of NMPs.

Yet, in Scenario 4.3, when the HEM nodes consume energy during Real-Time actions, it is driven solely by the NMP. The nodes take into account the impact of the local ramping tariff. Thus energy consumption differs from the baseline, even when the NMP is essentially the same. This demonstrates the individual response of actors in the system, to their locally-imposed tariffs.

7.6 Scenario 5: Dispatcher Node

7.6.1 Configuration

This scenario is designed to demonstrate the impact of adding Dispatcher Node to the system. A Dispatcher Node is used to measure subsystem energy flow and dispatch a storage node to provide optimal energy flow leaving the Dispatcher Node. The existing node DSO Distr is set as a Dispatcher Node. A new node, DSO Storage, is added as DSO Distr’s dispatchable storage.

For a Dispatcher Node to have an effect on the system, the subsystem must have prediction error. To this end, all parameters are used for elasticity and prediction error:

```
elastAir1 = 0.6;          % Elasticity of air1 loads
predictAir1 = 0.2;        % Prediction error for air1 loads
elastOther = 0.2;         % Elasticity of all other loads
predictOther = 0.2;       % Prediction error for all other loads
predictPV = 0.2;          % Prediction error for PV generation
```

Additionally, DSO Storage, is added to the system, for all scenarios. It uses the following parameters:

```
DDS.addNode('DSO Storage','DSO','DSO Distr');
      DDS.recent.type = 'Storage';
      DDS.recent.storageEnergyCap = 200;
      DDS.recent.storageMaxDischarge = 12;
      DDS.recent.storageMaxCharge = 12;
      DDS.recent.storageEnergyPct = 0.5;
      DDS.recent.storageControl = 'dynamic';
      DDS.recent.storageControlSlope = ...
          (0.025/(12/60))/(DDS.Pmin/DDS.Qbin); % for +-$0.05 range
      DDS.recent.storageDynEMAperiod = 10;
```

Thesee parameter were selected to provide clear illustration of the impact of the Dispatcher Node. As a result, the storage unit has the ability to overwhelm the rest of the system, for both power and energy. This will provide clear indications of the Dispatcher Node's methods and interactions, but it will not necessarily represent a well-deployed energy storage system.

Additionally, the storage node's control method is dynamic, with an EMA n-value of 10 minutes. This means the node will use the EMA of the NMP to determine its charge/discharge price threshold. Additionally, the parameter *storageControlSlope* is set to provide a demand curve with prices ranging from ±$0.05 for the calculated NMP EMA. This was selected to provide the

Dispatcher Node with a wide range of control options. If the node was instead operated for profit maximization, a different configuration would likely be implemented.

Finally, a Dispatcher Node will have more impact when adjusting energy flow to meet the constraints imposed by a tariff. To this end, a ramp EMA tariff is imposed on the DSO Market node. This is the same tariff used in Scenario 4.2: it uses N = 20 minutes for the EMA calculation, with $0.005/ΔkW as the penalty for both positive and negative ramping. As a result, this tariff incentivizes steady energy flow.

In Scenario 5.1, the base case is established. This uses the network shown below. For the base case, DSO Distr does not function as a Dispatcher Node. Thus, the node DSO Storage is not set to dispatchable mode and will function strictly based on its dynamic control behavior model. In Scenario 5.2, DSO Distri set to be a Dispatcher Node with dispatchType ‘contact’. In this mode, it will seek to dispatch DSO Storage to exactly meet the quantity specified by the NMP on its demand curve submission. There will be no consideration of the measured energy’s impact on the NMP and tariff imposed upon DSO Market.

In Scenario 5.3, DSO Distri is set to dispatchType ‘dynamic’. In this mode, it will seek to dispatch DSO Storage at the economically efficient point. In doing so, it will consider the measure subsystem energy quantity as an inelastic demand and will run a market using the storage node’s demand curve, imposed tariffs, and wholesale supply node. The clearing quantity from this market action will set the dispatched quantity from DSO Storage.

7.6.2 Results

For these results, an additional plot is provided: that of the newly added node, DSO Storage.

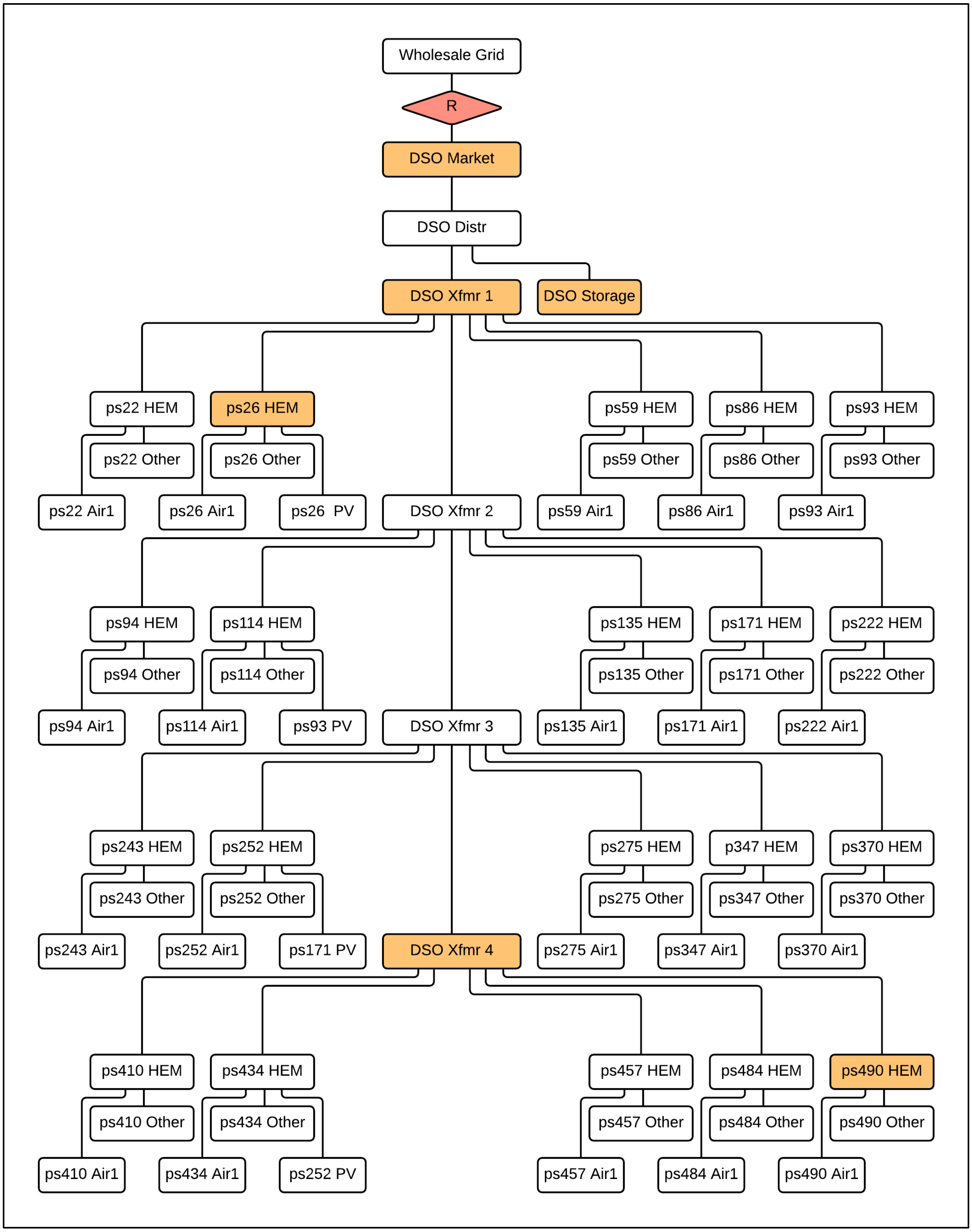
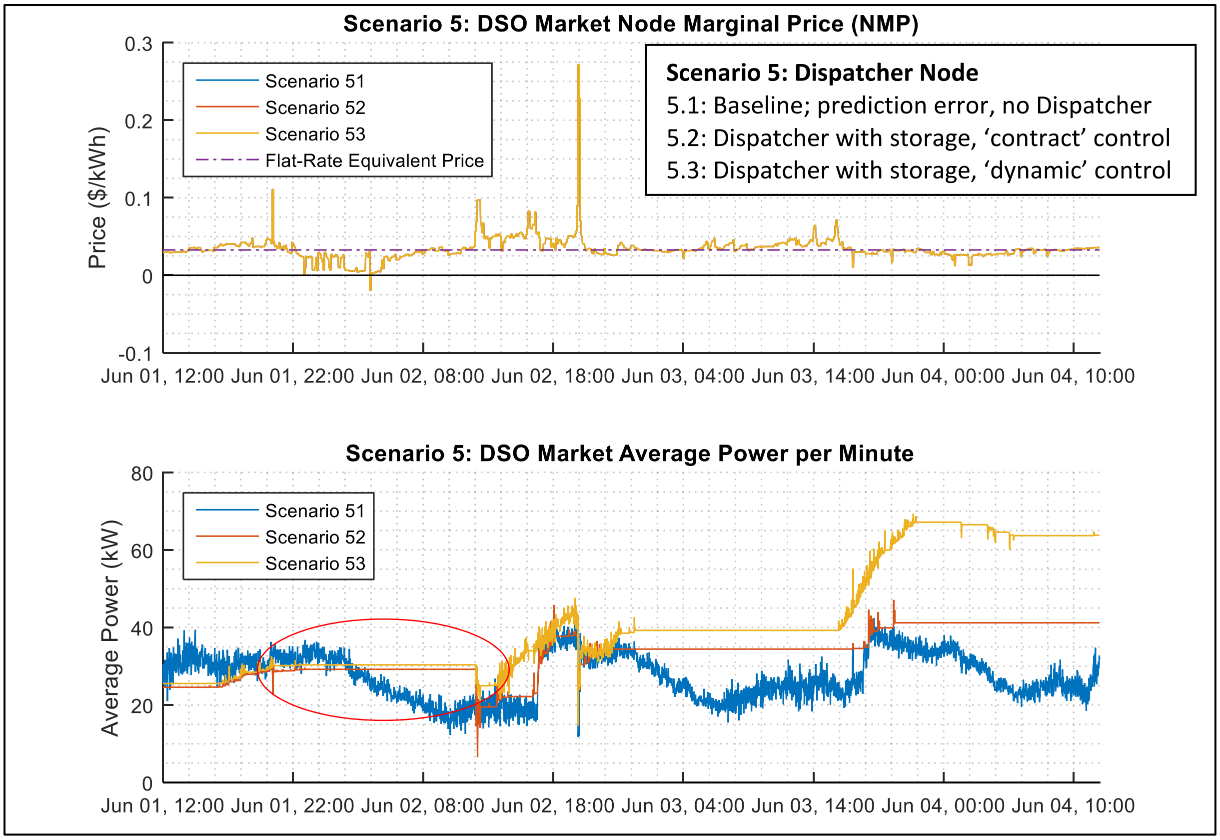
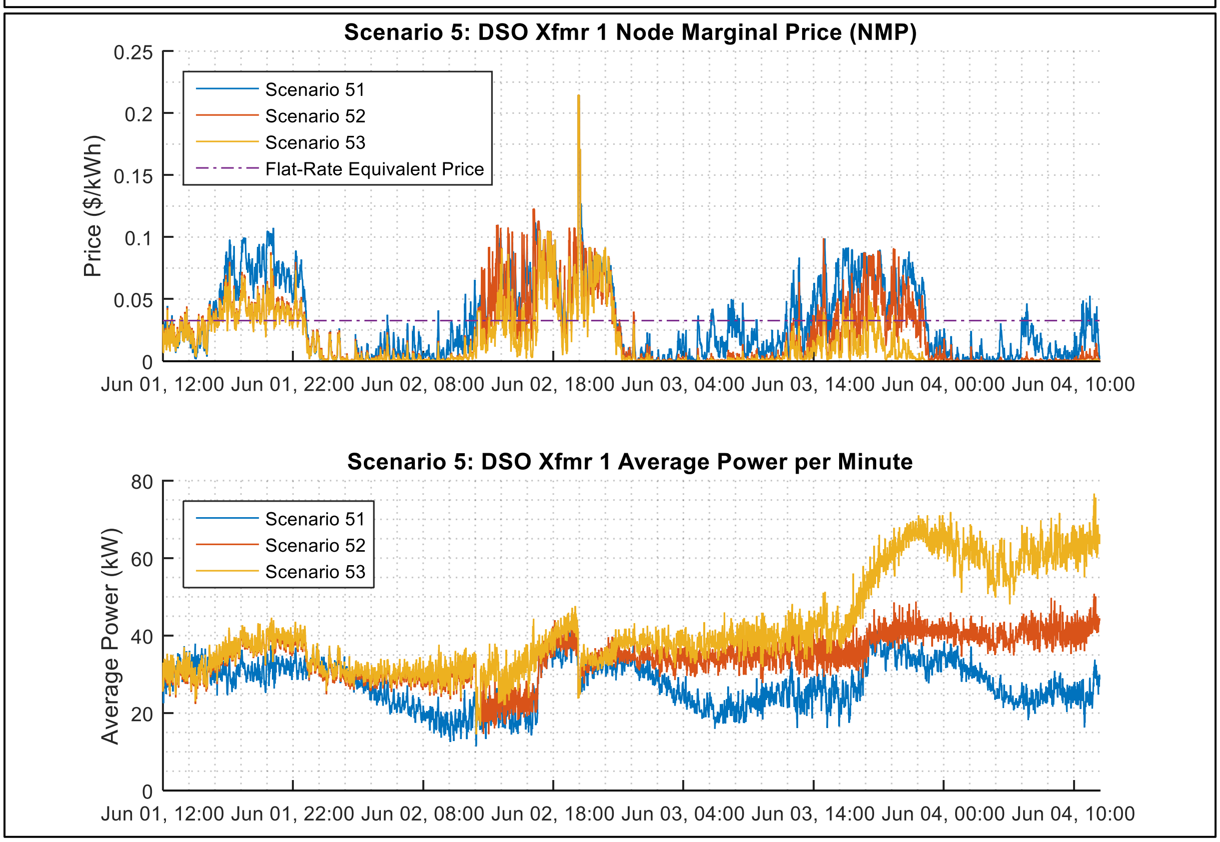
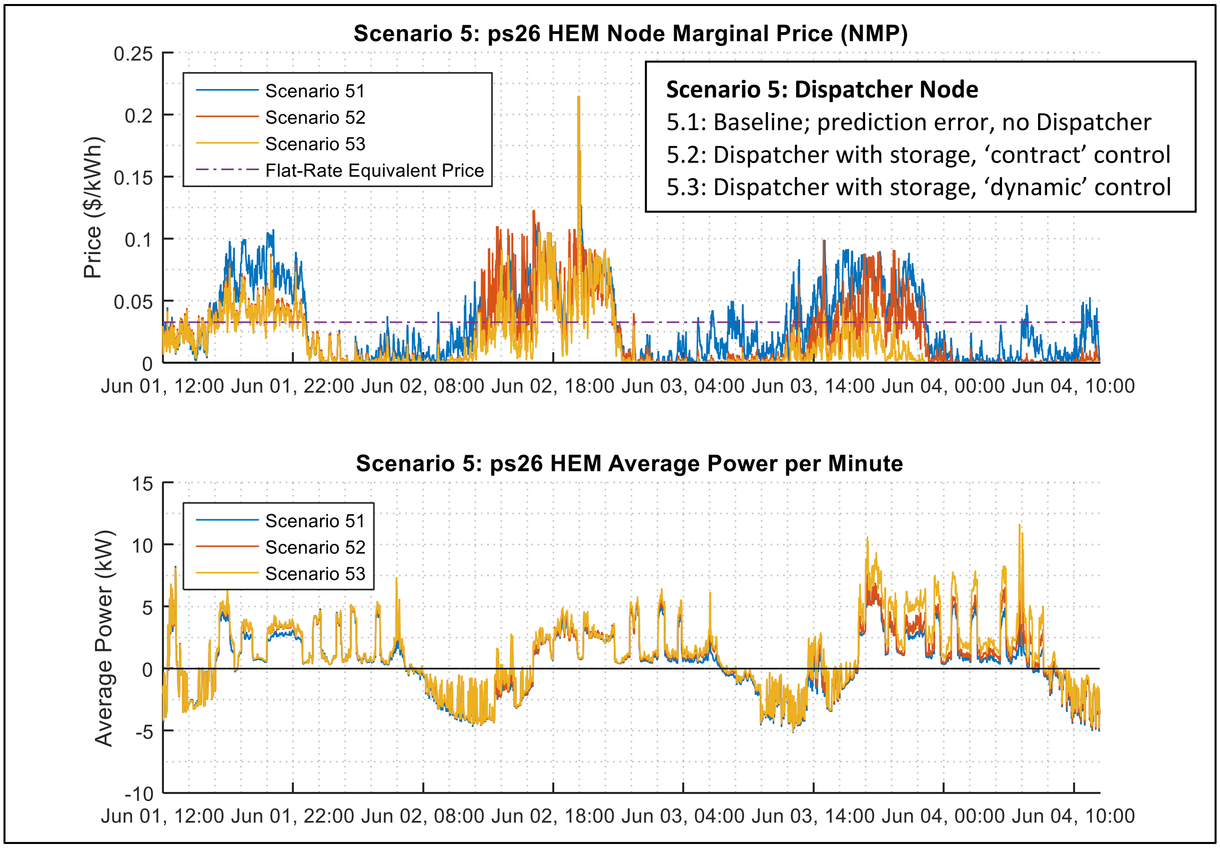
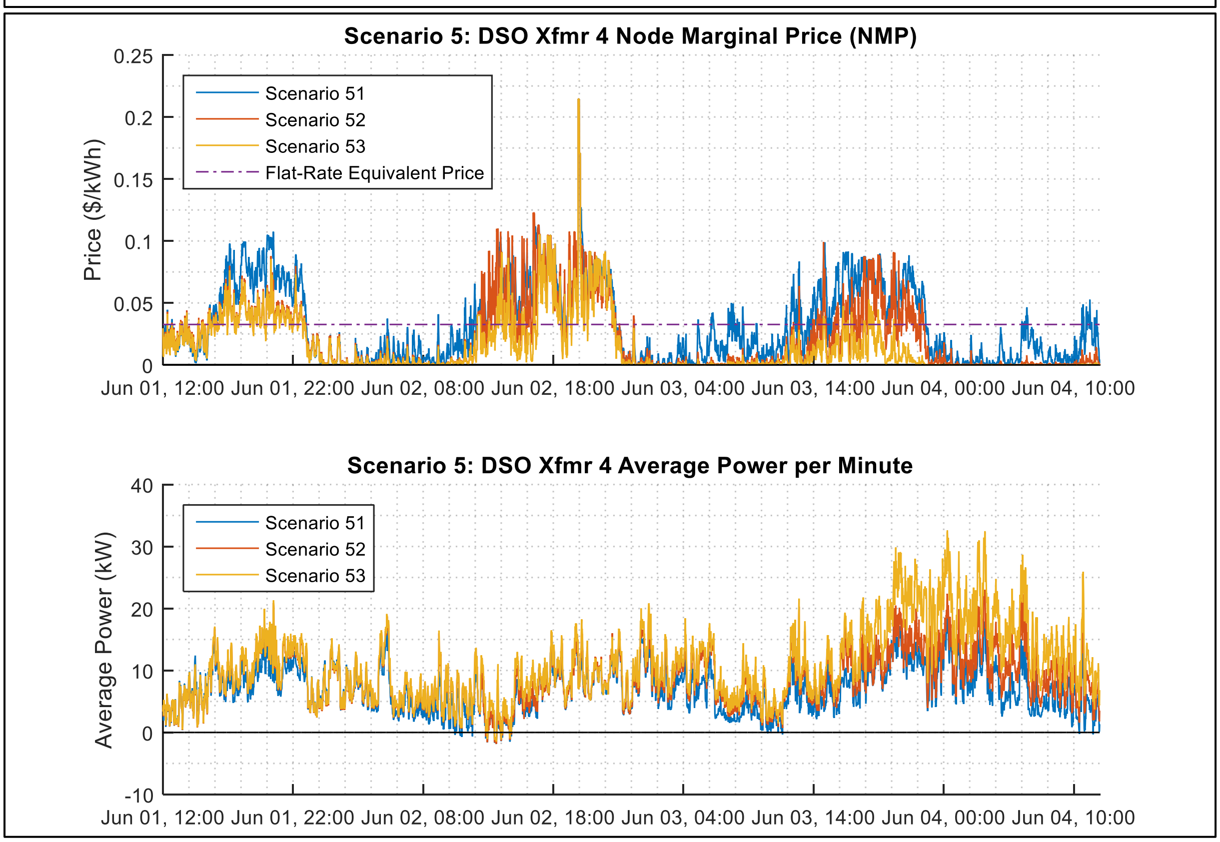


Figure 65: DTDM Network with Dispatchable Storage (Scenarios 5.1, 5.2, 5.3)


Scenario 5: DSO Market Node Marginal Price (NMP)
Price ($/kWh)
Scenario 51
Scenario 52
Scenario 53
Flat-Rate Equivalent Price
Scenario 5: Dispatcher Node
5.1: Baseline; prediction error, no Dispatcher
5.2: Dispatcher with storage, 'contract' control
5.3: Dispatcher with storage, 'dynamic' control
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 5: DSO Market Average Power per Minute
Average Power (kW)
Scenario 51
Scenario 52
Scenario 53
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00


Scenario 5: DSO Xfmr 1 Node Marginal Price (NMP)
Price ($/kWh)
Scenario 51
Scenario 52
Scenario 53
Flat-Rate Equivalent Price
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 5: DSO Xfmr 1 Average Power per Minute
Average Power (kW)
Scenario 51
Scenario 52
Scenario 53
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00


Scenario 5: ps26 HEM Node Marginal Price (NMP)
Price ($/kWh)
Scenario 51
Scenario 52
Scenario 53
Flat-Rate Equivalent Price
Scenario 5: Dispatcher Node
5.1: Baseline; prediction error, no Dispatcher
5.2: Dispatcher with storage, 'contract' control
5.3: Dispatcher with storage, 'dynamic' control
0.25
0.2
0.15
0.1
0.05
0
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 5: ps26 HEM Average Power per Minute
Average Power (kW)
Scenario 51
Scenario 52
Scenario 53
15
10
5
0
-5
-10
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00


Scenario 5: DSO Xfmr 4 Node Marginal Price (NMP)
Price ($/kWh)
Scenario 51
Scenario 52
Scenario 53
Flat-Rate Equivalent Price
0.25
0.2
0.15
0.1
0.05
0
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00
Scenario 5: DSO Xfmr 4 Average Power per Minute
Average Power (kW)
Scenario 51
Scenario 52
Scenario 53
40
30
20
10
0
-10
Jun 01, 12:00 Jun 01, 22:00 Jun 02, 08:00 Jun 02, 18:00 Jun 03, 04:00 Jun 03, 14:00 Jun 04, 00:00 Jun 04, 10:00

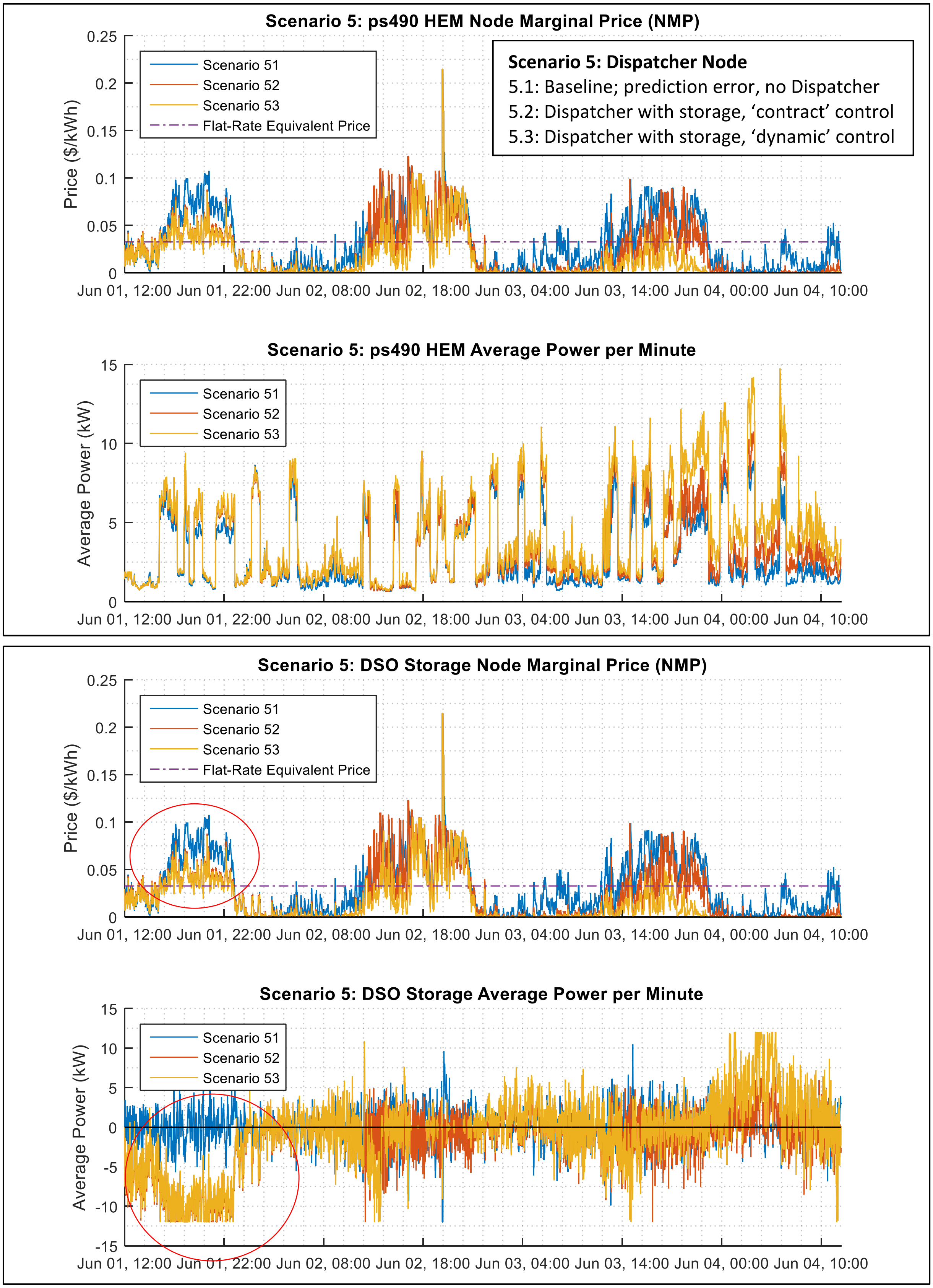


Figure 66: Scenario 5 Results

7.6.3 Observations

There are numerous observations to be made from these results. These observations are highlighted with red circles in Figure 66 above.

One, the Dispatcher Node is able to provide the optimal system behavior, despite prediction error of all other nodes in the system. The ramp EMA tariff is designed to incentivize constant energy consumption. As seen in Scenario 5.2 and 5.3, this is possible with Dispatchable Storage. However, the energy consumption is not constant is Scenario 5.1, despite the same configuration for the tariff, storage node, and all bottom-level nodes. This difference is due to the prediction error at the bottom-level nodes. In Scenarios 5.2 and 5.3, this error is overcome by the Dispatcher Node's fast-loop energy measurements and dispatch.

Two, when set to dispatchable mode, the NMP observed by DSO Storage does not actually control its energy consumption. Recall that a dispatchable storage node quantity is driven solely by the Dispatcher Node. The NMP is a reflection of Market Process, which occurs before Real-Time Actions. When set to dispatchable mode, the DSO Storage quantities are determined during Real-Time Action, based on the unanticipated deviations from the Market Process clearing quantity. This is observed between 12:00 - 22:00 on 1-June. The NMP is very similar for all three scenarios; however, the actual power flow is very different for Scenarios 5.2 and 5.3, when the DSO Storage is in dispatchable mode.

Three, DSO Storage obeys its 12 kW power flow limit. This demonstrates successful communication between the DSO Distr, the Dispatcher Node, and DSO Storage, the dispatchable storage. The Dispatcher Node was unable to command an energy quantity beyond the storage unit's limits.

This concludes the DTDM case study simulations. These case studies were presented to demonstrate the interactions between behavior models and tariff designs within the DTDM. It has been shown that the DTDM is able to facilitate energy transactions and provide nodes differentiated prices, based on energy, losses, and externalities. This demonstration should be considered the first step in advanced DTDM modeling and implementation.

8 Conclusion

8.1 Answers to Research Questions

In Section 1.2, six research questions were posed. They were as follows:

How should an energy marketplace be designed to support a Dynamic Distribution System?
What are the marketplace goals and design principles?
In what ways can a marketplace design be evaluated?
For a proposed market design, what are the marketplace rules?
For a proposed market design, how are externalities described and captured?
For a proposed market design, how do actors participate in the market?

Answers to these questions were suggested over the course of the literature review, proposed Dynamic Tariff Distribution Marketplace (DTDM) market design, simulation configuration, behavior model and proposed tariff design, and case studies. In this section, each will be addressed specifically, with commentary on how the DTDM provides those results.

How should an energy marketplace be designed to support a Dynamic Distribution System?

The marketplace must support all manner of DER, including demand- and supply-side technologies that support both energy and reliability system needs. This includes technologies providing generation, demand response, storage, and ancillary services. To this end, the DTDM is designed without restrictions on the participating DER. The DTDM marketplace process is generalized for any manner of market entry, in so much as the market entrant can meet the protocols described during Market Operation.

Additionally, the marketplace is one of four components of the larger Dynamic Distribution System. The marketplace must support the other three components: power system control, revised utility business models, and updated regulatory policies. To this end, the DTDM provides both minute-scale clearing energy to support sub-second power quality control and a fundamental market configuration for analysis in business model and regulatory assessments.

The DDS marketplace should seek to pass along the costs and benefits imposed by externalities to the parties responsible for causing the externalities. The DTDM specifically accomplishes this through the use of tariffs. Tariffs provide a direct way to describe and capture externalities as a cost component beyond energy and line loss costs. Actors in the system have the ability to locate and configure these tariffs for the benefit of the system as a whole.

What are the marketplace goals and design principles?

This research question is fundamentally linked to the first. However, it provides an opportunity to describe the specifics of the DTDM approach to a DDS marketplace. The DTDM seeks to provide minute-scale clearing energy for autonomous, independent actors using differentiated prices based on energy, losses, and externalities.

The stated goals of the DTDM are as follows. These goals were selected to meet the design principles of indirect control, scalability, flexibility, and efficient communication.

Support interconnection with a larger energy system. This is accomplished by providing the means for a DTDM to use the larger energy system as a supply source. Additionally, the flexibility of tariffs enables the DTDM system operator to provide the larger energy system with a financial commitment to maintaining desired system parameters. As an example, if the whole grid is concerned with evening ramp rates, the DSO can quantify this concern and implement the cost as a tariffs. This provides opportunities for the DDS to provide additional benefits to the larger energy system.

Provide intra-system control capabilities. This is accomplished through the dynamic tariffs, which can be updated during Market Operation. The system operator sets the level of incentives needed to match the desired outcome, based on their system analysis and acceptable risk levels. This level of capability may be less than that provided by a direct control scheme, but this still provides new levels of distribution-level control, while maintaining actor independence.

Quantify externalities. This is accomplished directly through dynamic tariffs. By placing tariffs at the location in which they are incurred, the costs and benefits of the externalities are borne by the parties responsible. This is demonstrated in case study Scenario 2. This goal serves to mitigate or eliminate the disadvantages associated with DER deployment.

Support islanded operation. This is accomplished by enabling inherent flexibility in the marketplace process. When the DTDM is connected to a larger energy system, it uses the larger energy system's pricing as its supply. However, when this connection does not exist, the DTDM is able to find its local energy clearing point. Additionally, the flexibility of tariffs enable a DSO to increases the incentives for behaviors that are more critical in islanding: accurate prediction and stable energy flows.

In what ways can a marketplace design be evaluated?

Many of the metrics for evaluating a DDS deployment depend on the specific instance in which it is applied. Thus, the primary method for evaluating a DDS marketplace is the flexibility it provides the DDS system operator.

A DDS marketplace should seek to mitigate or eliminate the disadvantages of DER. The DTDM accomplishes this by providing dynamic tariffs, which have been demonstrated to incentivize desired system behavior.

Additionally, RMI identified two relevant criteria for evaluating a revised business model: a customer's electricity bill should reflect the value of services imported and exported by the customer; a rate structure should pay for services, promote value, and be flexible. The DTDM meets both these objective by providing differentiated price signals with specific, defensible cost allocations.

The following three research questions are answered directly by the proposed DTDM.

For a proposed market design, what are the marketplace rules?

The more detailed answer to this question is provided in Section 4. In brief, the DTDM network consists of nodes, tariffs, and linkages. Nodes represent the consumption, generation, and transportation of energy. Linkages connect nodes. Tariffs impose the costs of externalities on actors within the system.

During the marketplace process, three steps occur for each market duration.

In Market Operation, bottom-level nodes first express their preferences as demand curve. A demand curve is a series of energy quantities and the price at which the node is willing to purchase that level of energy. Next, tariffs update their parameters. Tariffs represent the externalities tied to energy transport. Next, demand curves are consolidated as they pass to the marketplace node. This includes curve tariff adjustment, aggregation, and translation to capture the impact of physical losses. At the Marketplace Node, the market function determines the clearing price and quantity; this is a single-price auction. Finally, differentiated prices then propagate through the network to all nodes; this is each node's Node Marginal Price.

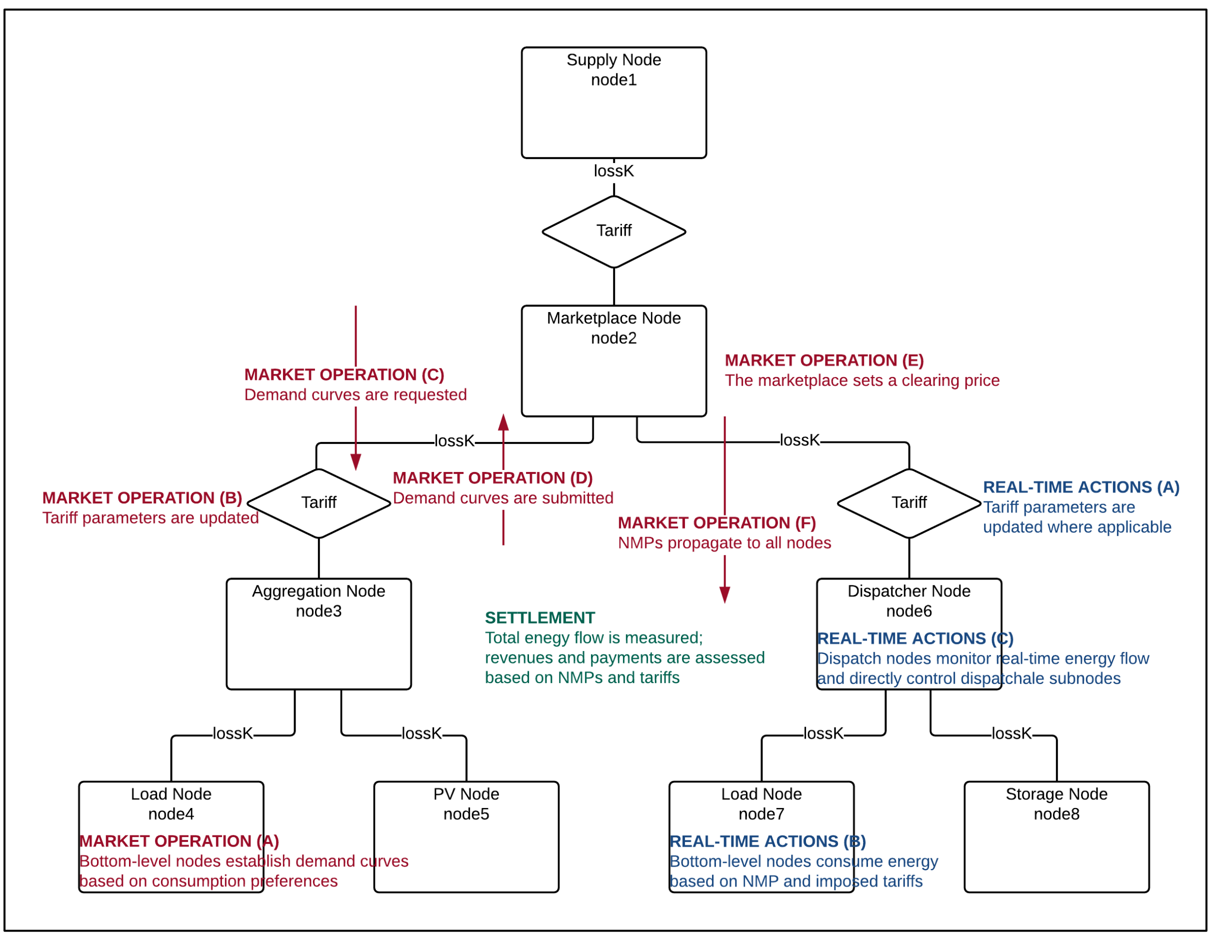


Figure 67: Market Process Overview

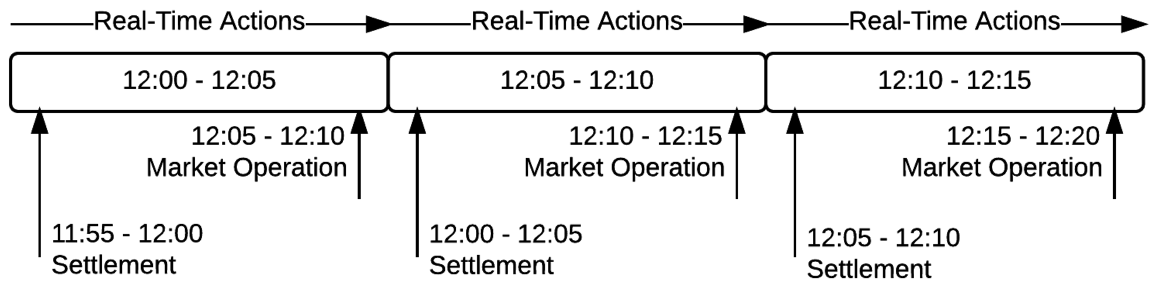


Figure 68: Market Process Timeline (5-Minute Market Interval)

In Real-Time Actions, bottom-level nodes consume energy based on their revised preferences, imposed tariffs, and the Node Marginal Price. Dispatcher Nodes monitor real-time energy flow and directly dispatch energy storage or dynamic loads to meet their target energy flows. During Settlement, energy flow is recorded and payments are assessed based on Node Marginal Prices and tariffs.

Within this process are some limited, basic restrictions on the communication and expectations between actors in the system. This includes the level of precision expected in demand curve submission and acceptable construction for the tariff revenue function.

For a proposed market design, how are externalities described and captured?

In the DTDM, tariffs explicitly describe externalities. By placing a tariff at the location in which the externality is incurred, the tariff is borne by the actors imposing the externality on the system. The DTDM provides ultimate flexibility in locating and configuring tariffs to the system operator and actors within the system.

For a proposed market design, how do actors participate in the market?

In the DTDM, actors must express their anticipated energy consumption preference in terms of a demand curve. When submitted to their supernode, the demand curve is a contractual offer. When a Node Marginal Price arrives, it sets the price of energy in addition to any pre-established tariffs. From this information, the actor determines their real-time consumption of energy. After consumption, settlement allocates revenues and payments between actors in the system.

8.2 Contributions Made

In this thesis, a market design for use in a dynamic distribution system has been proposed. This Dynamic Tariff Market Design (DTDM) uses a single-price double auction as its market clearing action, which parallels that used by many wholesale energy markets. All actors within the DDS are able to participate in the DTDM. They do so by submitting demand curves as contractual offers to buy and sell energy within a given time period. This general approach enables the DTDM to apply to any conceivable DER, insomuch as the DER can express its energy consumption (or generation) in terms of potential market prices. This process is accomplished for discrete market periods, each in the range or one to ten minutes long.

Further, energy consumption dynamics can be influenced by system actors through dynamic tariffs. Dynamic tariffs are pre-established at locations within the DTDM network and are updated before each market period. This approach allows indirect control of energy consumption by the DDS system operator; prices will reflect the impact of the tariffs, but will not directly control the behavior of actors within the system. This enables externalities to be described and borne by the actors responsible for imposing the externalities in the system. Like the rest of the DTDM, this approach is extremely general, providing flexibility for market participants.

In the development of the DTDM, numerous observations have been made on the complexities and requirements for system actors communicating demand curves in the market process. This includes rounding, curve aggregation, consideration of physical energy losses, and tariff revenue function limitations. Each of these decisions is a trade-off between multiple options; the discussions presented in this thesis outline the rationale used in the selection of each.

In this thesis, the DTDM market process was developed and simulated in Matlab to include simplified behavior models for household loads, PV generation, and energy storage. Example DTDM tariffs were developed in this simulation as well, including tariffs to prevent exceeding infrastructure capacity, and tariffs to limit energy ramp rates. The market process, behavior models, and tariffs were demonstrated to interact successfully in five case studies.

This thesis provides a starting point and insights into the challenges and opportunities for using a DDS marketplace to indirectly control electrical energy dynamics through the medium of price signals. The DTDM has demonstrated itself as a valid approach as a market design for use in a single-phase electrical system, with many independent actors.

8.3 Future Work

The DTDM is designed to support the overall development of the DDS concept. As such, the proposed market design is a starting point for further research. Topics for continued investigation are broken into six categories. Each is highlighted below, with observations and commentary.

8.3.1 Improved Modeling Flexibility

Currently, modeling implementation in Matlab includes: market rules and protocols, simulation analysis, pre-defined actor behavior models, and pre-defined tariff structures. A more robust and flexible software interface would separate these components to provide flexibility to DTDM modelers.

Users should be free to develop new behavior models without modifying the fundamental DTDM simulation code. Behavior models would then be assigned to nodes in a DTDM structure. This would enable comparison on different behavior models across otherwise similar scenarios. Additionally, code should include validation for a proposed behavior model. Prior to running a DTDM simulation, the validation would verify each behavior model is complete and non-contradictory.

The same approach should be used for implementing tariffs. The definition of tariff structures and parameters should be accomplished outside of the core DTDM simulation code. This would provide flexibility in developing and testing new proposed tariff parameters. Additionally, during demand curve adjustment and submission, the node should request updated parameters, not the actual tariff curve. Interpretation of the tariff instance into a practical tariff curve is the node's responsibility, not the tariff's responsibility.

Tracking and managing power factor may be an important component of DDS deployment. If so, the DTDM model should include power factor estimation and submission with demand curves. This would also require power factor adjustment, aggregation, and translation.

Finally, the current DTDM simulation includes a simplified energy flow model. In fact, this uses the same linkage loss factor used in the DTDM market. It would be beneficial to separate the market simulation from the energy flow simulation. For example, the OpenDSS platform can be linked to Matlab to perform quasistatic power flow calculation for each timestep. Both the DTDM market processes and system power flow have dynamic characteristics. Ideally, both would be modelled separately but linked through a series of common variables.

8.3.2 Physical Implementation

It would be beneficial to implement the DTDM concepts on physical devices. This would provide market participation by small-scale devices and/or virtual loads. This would provide insight into the practical capabilities of nodes to predict future consumption and how demand curves represent that consumption. Realistic demand curves for loads, storage devices, and generation could then be incorporated into the DTDM as improved behavior models. Physical implementation could also indicate the path-dependencies of energy consumption and price elasticity.

Additionally, physical implementation would require refinement of DTDM communication requirements. By necessity, this would consider competing communication processes and standards. Practical considerations of communication security and reliability would also be highlighted.

8.3.3 Improved Simulation Behavior Models

In addition to physical implementation, DTDM could be improved with direct focus on the simulation behavior models. Each behavior model includes a list of known limitations. This includes incorporating the “lumpiness” of actual energy consumption, actor risk aversion, and improved prediction error modeling. One large area for improvement is dynamic and time-varying price elasticity and the impact that previous consumption has on future preferences.

Improvements can also be made at the system-level for behavior modeling. If nodes represent households in the same neighborhoods, then there are common dependencies, such as weather patterns, that could be incorporated into the behavior models.

Finally, additional behavior models could be added to the simulation. This would include consumption preference prediction, demand curve generation, and real-time actions. Additional node types might include: non-thermal, schedulable loads; small- and medium-scale wind turbines (10 kW – 2 MW); combined cooling, heat, and power (CCHP) units; traditional, stand-alone fossil fuel generators; and electric vehicles as a specific application of energy storage.

8.3.4 Advanced Market Rules

The DTDM provides a basis for the DDS marketplace. However, practical implementation within a utility business model may necessitate additional market rules.

As one example, the current DTDM provides energy balance for a single phase. However, a practical implementation must allow for three-phase loads, unless deployment is limited to certain networks. This provides a challenge: how should markets interact for the three phases? For example, a capacity tariff would only apply to a single phase, so combining all three phases into a single market may not be useful. However, if a market operates for each phase, then three-phase loads may be faced with three different price signals. Perhaps controlling this is the responsibility of the end-user. Or perhaps the marketplace node should pick the same price for all three phases, optimizing among the three curves. Each approach has potential challenges and opportunities, and further analysis is warranted.

Additionally, the DTDM is currently designed for radial systems. This is useful for most distribution systems, but loop configurations exist. How can the DTDM be modified to support such systems?

The DTDM must support ancillary markets. This could include black-start capabilities, on-demand reserves, and voltage support. In general, these would be related, but separate from the primary DTDM market. Designing and incorporating these ancillary markets requires more analysis.

Finally, consideration must be made for non-participating customers and traditional DER aggregation actors. A realistic deployment of the DTDM in a DDS would include accommodation for households that desire to remain with their traditional rate structures. Additionally, DER aggregation firms present an existing stakeholder in the utility markets. Practical implementation must address the value they derive from the system and a method by which they participate in the marketplace.

8.3.5 System Operator Optimization

With the DTDM providing the framework, further research should consider the methods in which a system operator can optimize their behavior.

For example, what are the heuristics for tuning tariff parameters? Where should tariffs be located? What are the ideal tariff designs and where do they apply? It is observed that there is perhaps an inherent trade-off between system stability and economic burden when tuning tariff parameters. A system with high tariffs may have predictable dynamic characteristics, but the actors within the system may be exposed to high effective energy prices. Alternatively, low tariffs may not induce a predictable and acceptable level of dynamic system stability. This trade-off is ripe for further analysis.

Additionally, the system operator should be concerned with unfavorable feedback loops. Is it possible for a price signal to induce system instability? How is this prevented? Similarly, what types of system responses indicate high probability of unfavorable sub-second electrical dynamics?

A system operator has flexibility in sizing their network. How many nodes should participate in a market? Should there be markets-within-markets?

What level of "observability" does the DSO require? The DTDM assumes all non-owed nodes are "black boxes". Is this sufficient? What about measurement points for the DSO's portion of the DTDM network?

8.3.6 Utility Business Model Analysis

Finally, the DTDM will not be successful if it does not support revised business models. To this end, there is extensive analysis that should be performed, using the DTDM as the framework marketplace in the DDS.

Specifically, the current electrical distribution system includes existing technologies, customer preferences and expectations, and vested interests. In general, consideration should be given to enable current systems to transition to the DDS with DTDM with as little disruption as possible. In Section 2, it was observed that the DDS is most closely aligned with RMI's Network Utility business model approach to DER integration. To that end, the DTDM can serve as a target end-goal for business model transition. However, the DTDM may also prove useful for the transitional process itself. For example, a currently operating utility may elect to implement the fundamental framework of the DTDM, but with all actor interactions controlled by the utility itself. For example, all customers may be considered inelastic loads, with only utility-owned assets considered elastic (such as stationary storage or dispatchable load centers). With this approach, the utility could implement ramp and capacity tariffs to improve system performance. Such a “safe deployment” would be invisible to end-use customers, but would enable the utility to test the DTDM approach with a future transition in mind.

Regulated utility providers will need to propose modified rate structures that meet the goals of regulators. For example, regulators place limits on the prices to which customers are exposed. One question will be where such restrictions would end; e.g., are subnodes of the regulated utility subject to the same restrictions? Additionally, the ability of retail providers to deploy their own DER assets will affect the tariff parameters needed to provide required levels of control.

The demand curves used in the DTDM Market Operation are based on marginal costs and values. This most accurately reflects variable revenues and costs. However, there are fixed costs associated with electrical networks and variable costs that are very well approximated with fixed costs. These could be incorporated as revenue offset parameters in tariff structures. The balance between fixed costs, such as connection fees, and tariff designs must be examined.

What is the traditional utility’s role in managing the DTDM? If the DSO is a new business entity, what information can the traditional utility provide to assist the DTDM deployment? How should they be compensated for this information?

For example, perhaps a traditional utility could analyze their distribution networks and load profiles, characterize the network, and divide it into the most favorable zones for DDS deployment. They could also set or influence pricing by characterizing penalties and incentives for undesired and desired behavior. Alternatively, the traditional utility may transition to be the DSO itself.

Finally, a successful business model will provide value to all stakeholders. The DTDM should be used as a framework for determining how the DDS provides this value. This potentially includes: a lower effective cost of energy for end-users; more predictable revenues for distribution system owners; more reliable energy consumption for wholesale grid operators; more use cases for DER manufacturers; reduced environmental impacts for advocates; and increased customer access to energy for regulators.

8.4 Summary

This thesis describes a proposed marketplace for use in a Dynamic Distribution System (DDS). The Dynamic Tariff Distribution Marketplace (DTDM) seeks to provide minute-scale clearing energy for autonomous, independent actors using differentiated prices based on energy, losses, and externalities.

The DTDM was designed to mitigate the disadvantages associated with high levels of distributed energy resource (DER) penetration. It supports the closely-related elements of the DDS: power system control, revised business models, and updated regulations. The market rules of the DTDM were developed through an iterative process of simulation and market rule development. The DTDM network consists of nodes, tariffs, and linkages. Nodes represent the consumption, generation, and transportation of energy. Linkages connect nodes. Tariffs impose the costs of externalities on actors within the system.

During the marketplace process, three steps occur for each market duration. In Market Operation, bottom-level nodes express their preferences as demand curve, tariff update their parameters, and demand curves are consolidated as they pass to the marketplace node. There, the marketplace node determines the clearing price and quantity; differentiated prices then propagate through the network to all nodes. In Real-Time Actions, bottom-level nodes consume energy based on their revised preferences, imposed tariffs, and the Node Marginal Price. Dispatcher Nodes monitor real-time energy flow and directly dispatch energy storage or dynamic loads to meet their target energy flows. During Settlement, energy flow is recorded and payments are assessed based on Node Marginal Prices and tariffs.

This process was modeled in Matlab, to include all market processes. To support simulation, behavior models were developed for Load Nodes, Storage Nodes, and PV Nodes. Each provides a basis for behavior simulation in consumption preference prediction and real-time consumption. These behavior models and market rules were then demonstrated in a series of case studies.

The case studies demonstrate the fundamental dynamic interactions between actors and tariffs in the system. Not only do actors within the system respond to price signals, but tariffs can be used to provide differentiated price signals to market participants. This process enables the system operator to describe externalities and impose the costs on those responsible for causing the externalities. These results meet the stated goals of the DTDM marketplace.

9 References

[1] B. Beihoff, T. Jahns, R. Lasseter and G. Radloff, "Transforming the Grid from the Distribution System Out," Wisconsin Energy Institute, University of Wisconsin-Madison, 2014.

[2] Rocky Mountain Institute, "Net Energy Metering, Zero Net Energy and the Distributed Energy Resource Future: Adapting Electric Utility Business Models for the 21st Century," 2012.

[3] California ISO, *What the duck curve tells us about managing a green grid,* Folsom, CA, 2013.

[4] U.S. Energy Information Administration, "Electric Power Monthly with Data for September 2015," U.S. Department of Energy, Washington, DC, 2015.

[5] Madison Gas and Electric Company, *Time-of-use Rates for MGE Residential Customers,* 2105.

[6] Rocky Mountain Institute, "New Business Models for the Distribution Edge: The Transition From Value Chain to Value Constellation," 2013.

[7] S. Stoft, Power System Economics, Piscataway, NJ: IEEE Press, 2002.

[8] D. Pearce, The MIT Dictionary of Modern Economics, The MIT Press, 1992.

[9] EU EcoGrid, "EcoGrid EU: From Design to Implementation," 2013.

[10] New York Independent System Operator, *Market Participants User's Guide,* 2012.

[11] P. L. Joskow and C. D. Wolfram, "Dynamic Pricing of Electricity," *American Economic Review,* vol. 102, no. 3, pp. 381-385, 2012.

[12] A. Faruqui and S. Sergici, "Household response to dynamic pricing of electricity: A survey of 15 experiments," *Journal of Regulatory Economics,* vol. 38, pp. 193-225, 2010.

[13] S. P. Holland and E. T. Mansur, "The Short-Run Effects of Time-Varying Prices in Competitive Electricity Markets," *Energy Journal,* pp. 127-155, 2006.

[14] S. Borenstein, "The Long-Run Efficiency of Real Time Electricity Pricing," *The Energy Journal,* pp. 93-116, 2005.

[15] S. Borenstein, "Time-Varying Retail Electricity Prices: Theory and Practice," 2010.

[16] The Gridwise Architecture Council, "GridWise Transactive Energy Framework Version 1.0," 2015.

[17] Pecan Street Inc. Dataport, 2015.

[18] NYISO, "Markets & Operations," [Online]. Available: http://www.nyiso.com/public/markets_operations/market_data/pricing_data/index.jsp. [Accessed 6 February 2015].

[19] Tesla Motors, "Telsa Powerwall," [Online]. Available: https://www.teslamotors.com/POWERWALL. [Accessed 5 December 2015].

[20] Advanced Research Projects Agency - Energy, *Funding Opportunity Announcement - Network Optimized Distributed Energy Systems,* U.S. Department of Energy, 2015.

[21] National Institute of Standards and Technology, NIST/SEMATECH e-Handbook of Statistical Methods, 2012.

[22] National Fire Protection Association, NFPA 70: National Electrical Code, 2014 Edition, Quincy, MA: National Fire Protection Association, 2013.

Appendix A: MATLAB Case Study Initialization

```
% Thesis Case Studies
tic;
rng(1);     % Set seed value for randn random number generation

scenario = 53;

% For all scenarios
marketDuration = 1;         % Marketplace duration
simStart = 218161;          % Reference time 01-Jun-2014 12:00:00
simDuration = 3*24*60;      % Three days

% Load data sources
load('caseStudyNYISO.mat'); % Import ISO RT data for 2014 ($/kWh per min)
load('caseStudyPS.mat');    % Import Pecan Street use, air1, and gen data
    psUse = max(psUse,0);   % Clean data to remove negative use loads
    psAir1 = max(psAir1,0); % Clean data to remove negative air1 loads
    psGen = max(psGen,0);   % Clear data to remove negative PV generation

% Linkage parameters
lossFeeder = 60*(.806/1000)*500/(1000*7200^2*.9^2);
lossService = 60*(.201/1000)*50/(1000*120^2*.9^2);

% Common Node Parameters
switch scenario
    case 11
        elastAir1 = 0;          % Elasticity of air1 loads
        predictAir1 = 0;        % Prediction error for air1 loads
        elastOther = 0;         % Elasticity of all other loads
        predictOther = 0;       % Prediction error for all other loads
        predictPV = 0;          % Prediction error for PV generation
    case 12
        elastAir1 = 0.4;        % Elasticity of air1 loads
        predictAir1 = 0.05;     % Prediction error for air1 loads
```

```
        elastOther = 0;         % Elasticity of all other loads
        predictOther = 0.2;     % Prediction error for all other loads
        predictPV = 0.05;       % Prediction error for PV generation
    case 13
        elastAir1 = 0.8;        % Elasticity of air1 loads
        predictAir1 = 0.05;     % Prediction error for air1 loads
        elastOther = 0.4;       % Elasticity of all other loads
        predictOther = 0.2;     % Prediction error for all other loads
        predictPV = 0.05;       % Prediction error for PV generation
    case {51,52,53}
        elastAir1 = 0.6;        % Elasticity of air1 loads
        predictAir1 = 0.2;      % Prediction error for air1 loads
        elastOther = 0.2;       % Elasticity of all other loads
        predictOther = 0.2;     % Prediction error for all other loads
        predictPV = 0.2;        % Prediction error for PV generation
    otherwise
        % all scenarios 2-4
        elastAir1 = 0.6;        % Elasticity of air1 loads
        predictAir1 = 0;        % Prediction error for air1 loads
        elastOther = 0.2;       % Elasticity of all other loads
        predictOther = 0;       % Prediction error for all other loads
        predictPV = 0;          % Prediction error for PV generation

end

% Common Tariff Parameters
rampWindow = 20;          % Ramp rate and analysis running average frame

%---INITIALIZE SYSTEM---
% Create DDS object
DDS = classDDS;
    DDS.timeseries = (datetime('01-Jan-2014 00:00:00'):minutes(1):...
        datetime('01-Jan-2015 00:00:00'))'; % Set DDS timeseries to 2014
    DDS.pFRE = median(nyisoRT2014kWh(simStart:simStart+simDuration));
    DDS.Qbin = 0.0001;
    DDS.Pmin = 1e-12;
```

```
% Create Node Objects
% Node connection to Wholesale/Grid
DDS.addNode('Wholesale Grid','ISO');
    DDS.recent.supplyDataSource = nyisoRT2014kWh;
    clear nyisoRT2014kWh;
% DDS Marketplace Node
DDS.addNode('DSO Market','DSO','Wholesale Grid');
    DDS.recent.setMarket(1,marketDuration);
% DSO Infrastructure
DDS.addNode('DSO Distr','DSO','DSO Market');
DDS.addNode('DSO Xfmr 1','DSO','DSO Distr');
    DDS.recent.lossK = lossFeeder;
DDS.addNode('DSO Xfmr 2','DSO','DSO Xfmr 1');
    DDS.recent.lossK = lossFeeder;
DDS.addNode('DSO Xfmr 3','DSO','DSO Xfmr 2');
    DDS.recent.lossK = lossFeeder;
DDS.addNode('DSO Xfmr 4','DSO','DSO Xfmr 3');
    DDS.recent.lossK = lossFeeder;

% Add Households and Bottom-level Nodes
xfmrs = 4;          % Number of transformers along feeder
servicePer = 5;     % Number of households per transformer
for n = 1:xfmrs
    for m = 1:servicePer
        index = (n-1)*servicePer + m;
        strID = char(psUserID(index));
        DDS.addNode(strcat(strID,' HEM'),strID,['DSO Xfmr ',num2str(n)]);
            DDS.recent.lossK = lossService;
        DDS.addNode(strcat(strID,' Air1'),strID,strcat(strID,' HEM'));
            DDS.recent.loadDataSource = psAir1(:,index);
            DDS.recent.loadElast = elastAir1;
            DDS.recent.predictErrorSD = predictAir1;
        DDS.addNode(strcat(strID,' Other'),strID,strcat(strID,' HEM'));
            DDS.recent.loadDataSource = psUse(:,index) - psAir1(:,index);
            DDS.recent.loadDataSource = max(0,DDS.recent.loadDataSource);
```

```
            DDS.recent.loadElast = elastOther;
            DDS.recent.predictErrorSD = predictOther;
        if m == 2
            DDS.addNode(strcat(char(psGenUserID(n)),' PV'),strID,...
                    strcat(strID,' HEM'));
                DDS.recent.type = 'PV';
                DDS.recent.pvDataSource = psGen(:,n);
                DDS.recent.predictErrorSD = predictPV;
        end
    end
end

switch scenario
    case 52
        DDS.nodes(3).setDispatcher;
        DDS.nodes(3).dispatchType = 'contract';
    case 53
        DDS.nodes(3).setDispatcher;
        DDS.nodes(3).dispatchType = 'dynamic';
    otherwise
        % no dispatcher nodes
end

if scenario > 50
    DDS.addNode('DSO Storage','DSO','DSO Distr');
        DDS.recent.type = 'Storage';
        DDS.recent.storageEnergyCap = 200;          % kWh
        DDS.recent.storageMaxDischarge = 12;        % kW
        DDS.recent.storageMaxCharge = 12;           % kW
        DDS.recent.storageEnergyPct = 0.5;          % initial SOC 50%
        DDS.recent.storageControl = 'dynamic';
        DDS.recent.storageControlSlope = ...
            (0.025/(12/60))/(DDS.Pmin/DDS.Qbin);    % for +-$0.05 range
        DDS.recent.storageDynEMAperiod = 10;
    if scenario > 51
        DDS.recent.storageDispatchFlag = 1;
```

```
    end
end

% With nodes complete, clear datasource and non-simulation variables
clear xfmrs servicePer n m index strID
clear psAir1 psGen psGenUserID psUse psUserID

% Add Tariffs
switch scenario
    case 22
        DDS.addTariff('DSO Xfmr 4','Capacity');
            DDS.recent.capPwrLimit = 20;
    case 23
        DDS.addTariff('DSO Xfmr 4','Capacity');
            DDS.recent.capPwrLimit = 20;
            DDS.recent.capRampType = 'linear';
            DDS.recent.capPpeak = .05;
            DDS.recent.capQrampPct = .8;
    case 32
        DDS.addTariff('DSO Market','Ramping');
            DDS.recent.rampWindow = rampWindow;
            DDS.recent.rampPriceRise = 0.005;
            DDS.recent.rampPriceFall = 0.005;
    case 33
        DDS.addTariff('DSO Market','Ramping');
            DDS.recent.rampWindow = rampWindow;
            DDS.recent.rampPriceRise = 0.020;
            DDS.recent.rampPriceFall = 0.020;
    case 42
        DDS.addTariff('DSO Market','RampEMA');
            DDS.recent.rampEMAnPeriod = rampWindow;
            DDS.recent.rampEMApriceRise = 0.005;
            DDS.recent.rampEMApriceFall = 0.005;
    case 43
        for ownerIndex = 3:22
            DDS.addTariff([DDS.owners(ownerIndex).name,' HEM'],'RampEMA');
```

```
                DDS.recent.rampEMAnPeriod = rampWindow;
                DDS.recent.rampEMApriceRise = 0.005;
                DDS.recent.rampEMApriceFall = 0.005;
        end
    case {51,52,53}
        DDS.addTariff('DSO Market','RampEMA');
            DDS.recent.rampEMAnPeriod = rampWindow;
            DDS.recent.rampEMApriceRise = 0.01;
            DDS.recent.rampEMApriceFall = 0.01;
    otherwise
        % no tariffs
end

%---RUN SIMULATION---
% Run simulation
DDS.run(simStart,simDuration);

%---SAVE RESULTS---
save(['caseStudy',num2str(month(datetime)),'-',...
    num2str(day(datetime)),'-',num2str(hour(datetime)),...
    num2str(minute(datetime)),'scen',num2str(scenario)]);

toc;
```

Appendix B: MATLAB Object Classes and Parameters

```
classdef classDDS < handle
    properties
        % Object Parameters
        nodes@classNode;          % Node objects in the DDS
        tariffs@classTariff;      % Tariff objects in the DDS
        owners@classOwner;        % Owner objects in the DDS
        markets@classNode;        % Market nodes in the DDS
        dispatchers@classNode;    % Dispatcher nodes in the DDS
        % Time Parameters
        t = 0;                    % current timestep
                                  %   (last DDS.run iteration)
        timeseries = datetime.empty(0,1);   % System datetime values
        tRefOffset = 0;           % Reference index for staring time
                                  %   (t = T + tRefOffset)
        % System Parameters
        Pcap = 100;               % Price cap; used as "infinite price"
                                  %   for boundary conditions
        pFRE = 0.10;              % Flat-rate equivalent price ($/kWh)
                                  %   used as default in load curves
        Qbin;                     % Qbin for DDS market communication
        Pmin;                     % Pmin for top-level node
        dispatchLevels = 0;       % Total number of dispatch levels
        % UI Parameters
        recent;                   % last object (node, owner, tariff)
                                  %   added or found with obj.find()
        % Meter Parameters
        meterTime;                % Array of simulation timestep values
    end

    properties (Dependent)
        tRef;                     % Reference index for timeseries
                                  %   (set with obj.t and obj.tRefOffet)
    end
```

```
classdef classNode < handle & matlab.mixin.CustomDisplay
    properties
        % GENERAL PARAMETERS; DISPLAY WHEN CALLING ALL NODES
        name;                     % Must be unique
        type = 'Load';            % Default node type
        % Relational Parameters
        DDS @classDDS;            % DDS containing this object
        owner @classOwner;        % Owner of this node
        super @classNode;         % Node above in hierarchy
        tariffs @classTariff;     % Tariffs between node and its supernode
        sub @classNode;           % Nodes below in hierarchy
        % General Parameters
        lossK;                    % Loss translation constant
        Pmin;                     % Pmin (set by supernode)
        % Communication and Transation Parameters
```

```
priceContr;              % Node Marginal Price for current
                         %   market interval
quantContrAvg;           % Contracted quantity (measured
                         %   at node), averaged per minute
priceClear;              % Clearing price at this node, to
                         %   clear WRT tariffs and priceContr
quantClearAvg;           % Clearing quantity, averaged per minute
quantActual;             % Actual demand (kWh)
% Demand Curve Parameters
curve;                   % Demand curve for this node, actual
curveSubmit;             % Demand curve for this node, adjusted
                         %   by tariffs for supernode submission
curveSubmitTrans;        % Demand curve submission, translated
                         %   to incorporate losses
predictErrorSD = 0;      % Standard deviation of error for demand
                         %   prediction (expressed as a percent of
                         %   actual demand; must be positive)
% NODE TYPE-SPECIFIC PARAMETERS
% Load Node Parameters
loadDataSource;      % Data source for estimated load in avgKW/min
loadElast = 0;       % Price elasticity of demand (unsigned value)
loadPanchor;         % Anchor price for elastic curve
loadQanchor;         % Anchor quantity from loadDataSource
loadQmax1min;        % Max load quantity, from loadDataSource
% PV Node Parameters
pvDataSource;        % Data source for output in avgKW/min
% Storage Node Parameters
storageDispatchFlag = 0;     % 1 if storage unit is controlled
                             %   by Dispatcher node
storageMaxDischarge;         % maximum discharge rate (avgKW)
storageMaxCharge;            % maximum charge rate (avgKW)
storageEnergyCap;            % available energy storage (kWh)
storageEnergyPct = 0;        % current SOC; percent of EnergyMax
storageControl = 'manual';   % 'manual': set all price thresholds;
                             %   'dynamic': uses priceEMA;
                             %   'dispatchcontract': dispatched
storageControlSlope = 1;     % for use with Dynamic control;
                             %   1 is min for Pmin/Qbin slope,
                             %   increase for greater price range
storagePriceMaxDischarge;    % price for maximum discharge
storagePriceMinDischarge;    % price for minimum discharge
storagePriceMinCharge;       % price for minimum charge
storagePriceMaxCharge;       % price for maximum charge
storageDynEMAperiod;         % for use with dynamic control;
                             %   N-period for EMA
storageDynEMAcurrent = 0;    % for use with dynamic control;
                             %   stores current EMA (initially 0)
% Market Node Parameters
marketNext;          % next market start (index WRT obj.DDS.t)
marketStart;         % market start (index WRT obj.DDS.t)
marketDuration;      % market period length (minutes)
% Dispatcher Node Parameters
dispatchSupply @classNode;   % Supply supernode (intramarket)
dispatchLevel;               % Dispatch level used in simulation
dispatchNode;                % Node controlled by the dispatcher
```

```
        dispatchType;                   % 'contract': match contracted Q;
                                        %   'dynamic': contract P and tariffs
        dispatchQuantRequest;           % Quantity requested by Dispatcher
        % Supply Node Parameters
        supplyPrice;            % current flat-rate price
        supplyDataSource;       % Data source for pricing
    end
```

```
classdef classTariff < handle
    properties
        type;                       % capacity, ramping, rampEMA, flat
        DDS@classDDS;               % DDS containing this object
        owner@classOwner;           % Owner of the tariff
        revenue = 0;                % last revenue calculated in obj.settle
        % Meter Parameters (row for each timestep)
        mFlag = 1;                  % 1 if this node records meter values
        meterQuantActual;           % quantity at subnode
        meterRevenue;               % revenue
        meterPriceEff;              % effective price
        settleLast = 0;             % DDS.t for last settlement
        % Type: Capacity Parameters
        capPwrLimit;                % Maximum power flow (kW)
        capFlow = 'bi';             % Acceptable direction(s) of energy
                                    %   flow; select bi, pos, or neg
        capRampType = 'flat';       % select flat, linear, quadratic
        capPpeak = [];              % Peak price at curve limit
                                    %   (non-flat curves)
        capQrampPct = [];           % Percent of limit where ramp begins
        capQlimit;                  % Maximum energy flow quantity
        % Type: Ramping (SMA) Parameters
        rampWindow;                 % Running average window (min)
        rampPriceRise;              % Node payment to the tariff for
                                    %   positive change in avgKW
        rampPriceFall;              % Node payment to the tariff for
                                    %   negative change in avgKW
        % Type: RampEMA Parameters
        rampEMAnPeriod;             % N-period for EMA
        rampEMApriceRise;           % Node payment for positive change
                                    %   of deltaEMA*60
        rampEMApriceFall;           % Node payment for negative change
                                    %   of deltaEMA*60
        rampEMAcurrent;             % current running EMA
        % Type: Flat Parameters
        flatDataSource;             % source of flat rate price
        flatRate;                   % current flat rate

        indexTariff;                % index of tariff in obj.DDS.tariffs
  end
```

```
classdef classOwner < handle
    properties
        name;                % Must be unique
        DDS @classDDS;       % DDS containing this object
        nodes @classNode;    % Nodes controlled by this owner
        tariffs @classTariff;    % Tariffs controlled by this owner
        % Edge object indices
        edgeNodesA @classNode;      % a) no supernode (i.e. top node)
        edgeNodesB @classNode;      % b) different owner supernode
        edgeNodesC @classNode;      % c) no subnodes (i.e. end node)
        edgeNodesD @classNode;      % d) 1+ different owner subnode
        edgeNodesDSubs @classNode;  % index of Case D subnodes
        edgeNodesE @classNode;      % e) different owner tariff
        edgeNodesETariffs @classTariff; % index of Case E tariffs
        edgeTariffs @classTariff;   % Owned tariff objects that connect
                                    %   to a different owner node
        edgeSetFlag = 0;            % flag indicating edges set
        edgeMeterQuantIn;    % energy into owner's subsystem
        edgeMeterQuantOut;   % energy out of owner's subsystem
        edgeMeterQuantLoad;  % energy consumed/generated by owner
        edgeMeterRevenue;    % revenue into owner's subsystem
        edgeMeterPriceEff;   % Minute-by-minute effective price for
                             %   consumption/gen (-revenue/quantLoad)
        edgeAggPriceEff;     % Aggregate effective price
                             %   (-sum(revenue)/sum(quantLoad))
        indexOwner;          % index of owner in obj.DDS.owners
    end
```