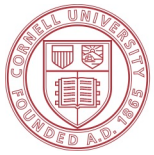


Cornell University

From Deterministic Economic Dispatch to Secure Stochastic Unit Commitment/Optimal Power Flow with

MOST, the new MATPOWER Optimal Scheduling Tool

Ray Zimmerman[1],
Carlos Murillo-Sánchez[3], Daniel Muñoz-Álvarez[1], Alberto Lamadrid[2]

[1]*Cornell University (Ithaca, NY)*
[2]*Lehigh University (Bethlehem, PA)*
[3]*National University of Colombia (Manizales, Caldas, Colombia)*

FERC Technical Conference on
Increasing Market and Planning Efficiency through Improved Software
June 27-29, 2015

CERTS
CONSORTIUM FOR ELECTRIC RELIABILITY TECHNOLOGY SOLUTIONS

Acknowledgements

- Alberto Lamadrid
- Daniel Muñoz-Álvarez
- Carlos Murillo-Sánchez
- Bob Thomas
- C. Lindsay Anderson
- Wooyoung Jeon
- Tim Mount

Outline

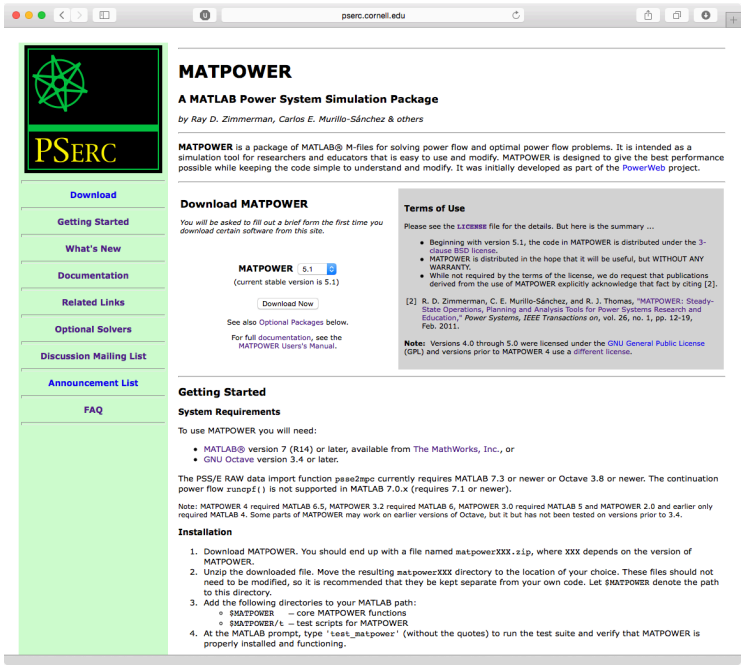
- MATPOWER Background
- MOST Overview
  - What is MOST?
  - Problem Formulation
- MOST Examples
  - Single-period continuous variable problems
  - Multi-interval problems with UC

MATPOWER

Free, open-source power system simulation environment with power flow (PF), Continuation PF, extensible OPF, now with MOST, stochastic unit commitment and multi-interval OPF.

http://www.pserc.cornell.edu/matpower/

- implemented in Matlab language, compatible with free GNU Octave
- interfaces to state-of-the-art solvers
- used worldwide in teaching, research, industry
- research enabling tool, whose momentum & impact continues to grow, as evidenced by
  - **1325 citations** of 2011 MATPOWER paper*
  - roughly **20,000** downloads per year



R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education," *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 12-19, Feb. 2011.

* *Google Scholar, 6/20/16*

Brief MATPOWER History

- MATPOWER began in 1996 as an AC OPF tool used to clear markets for PowerWeb, a power market testing platform.
- Publicly released in 1997 (v1 and v2)
- Development focused on internal use until release of v3 in 2005
- Some of the major features added over time:
  - DC power flow and DC OPF
  - additional power flow algorithms
  - extensible OPF architecture
  - improved OPF solvers, including native Matlab primal-dual interior point
  - continuation power flow
  - explicit open source license (BSD)
  - realistic test cases (Polish, French, European cases up to 13k buses)
- Combination of high quality solvers and ready-to-use cases created a de facto benchmark platform (for optimization & power systems research)
- MATPOWER 6.0b1* includes MOST (**M**ATPOWER **O**ptimal **S**cheduling **T**ool)

**released June 1, 2016 – as of June 18, more than 1,000 downloads*

Outline

- MATPOWER Background
- MOST Overview
  - What is MOST?
  - Problem Formulation
- MOST Examples
  - Single-period continuous variable problems
  - Multi-interval problems with UC

What is MOST?

- **M**ATPOWER **O**ptimal **S**cheduling **T**ool*
- Optimization tool for solving a generalized scheduling problem, which can include:
  - price responsive demand
  - transmission network model (or not)
  - secure dispatch via co-optimized reserves (zonal or endogenously determined from contingencies)
  - stochastic inputs (e.g. renewables, load)
  - integer commitment decisions, startup & shutdown costs
  - multi-interval dispatch with ramping constraints & costs
  - full unit commitment with minimum up- & down-time constraints
  - energy storage and deferrable demand resources

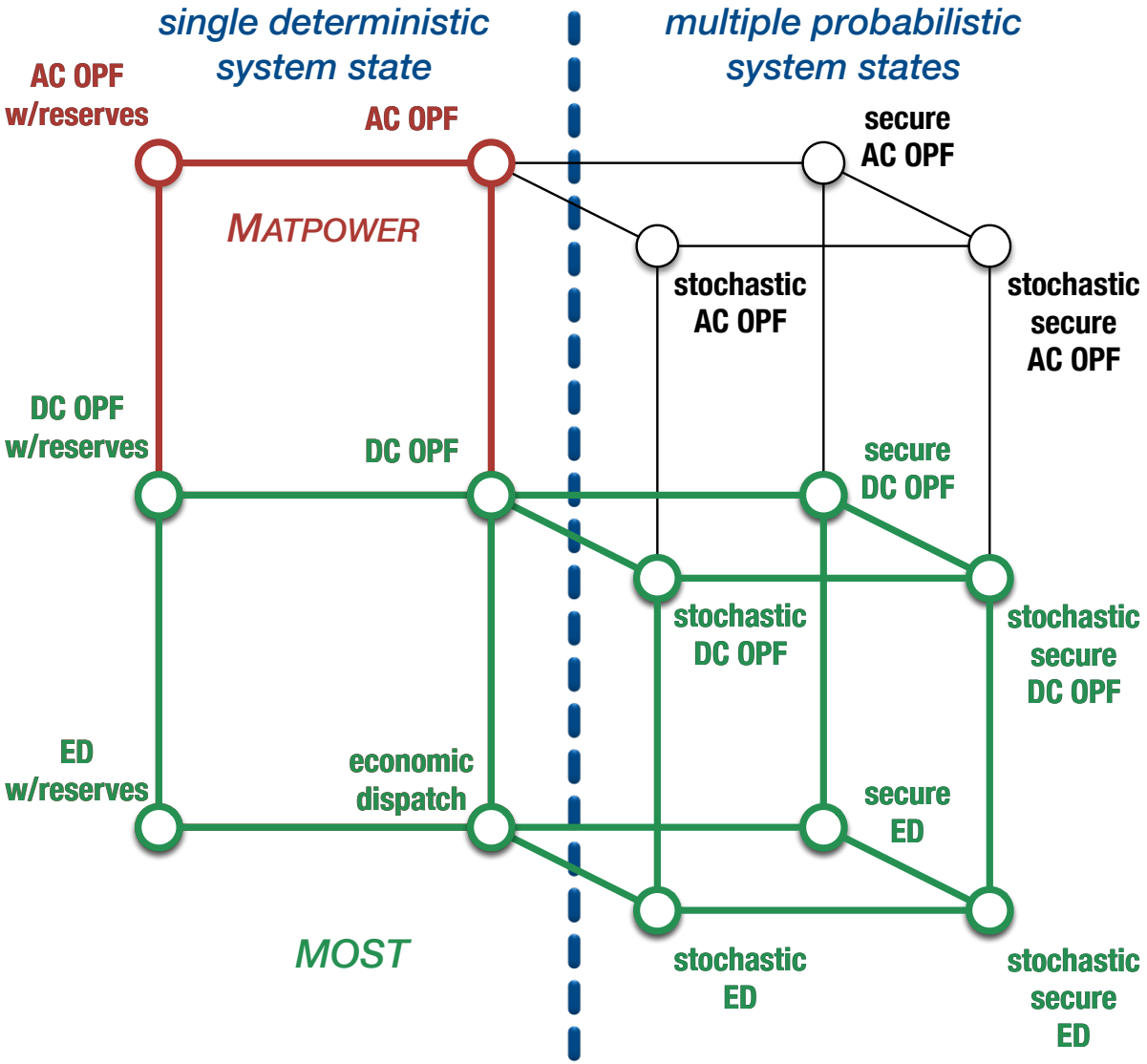
* Previously referred to as MOPS – **M**ATPOWER **O**ptimal **P**ower **S**cheduler

Our Formulation

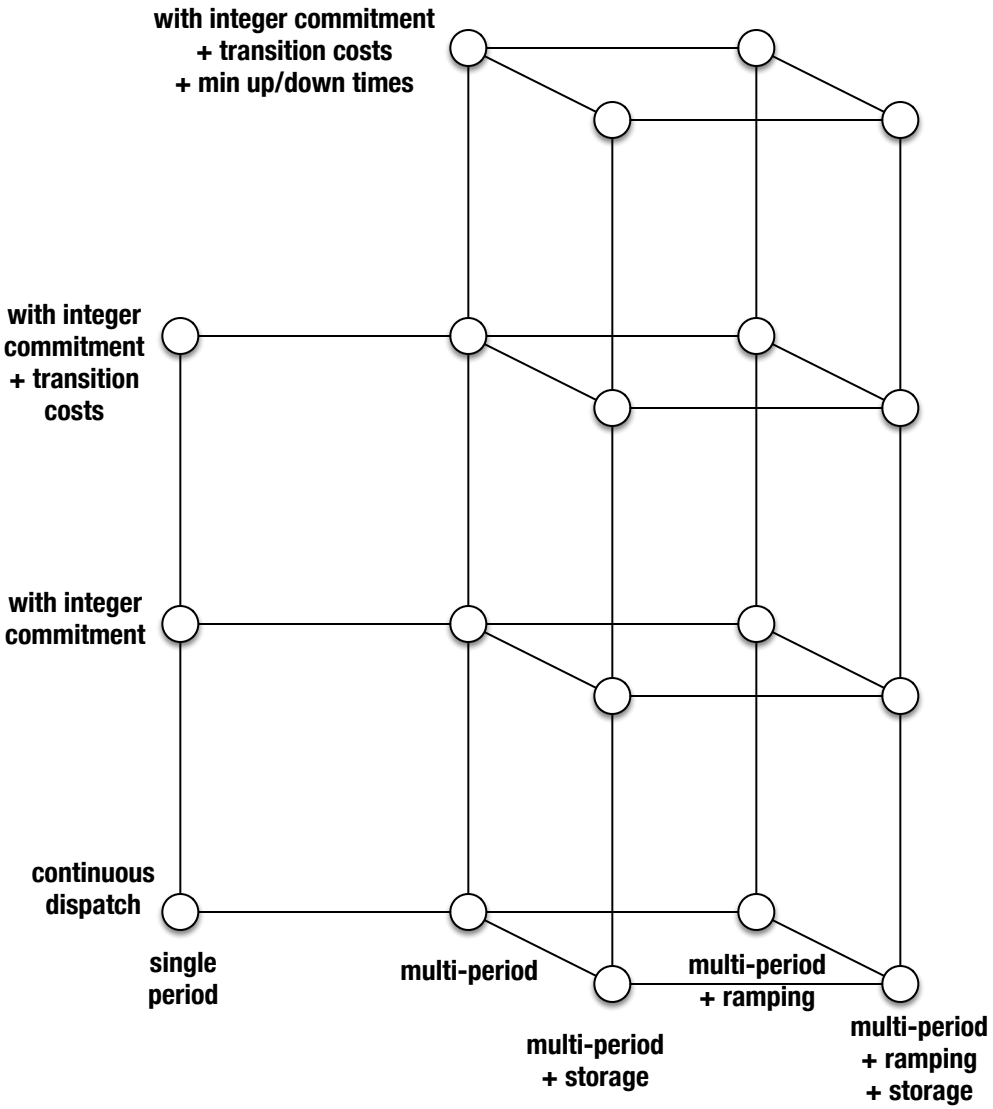
- Generalized extension of combined UC/OPF problem, to include …
  - intertemporal energy constraints for storage, flexible/deferrable demand
  - endogenous, price responsive contingency and ramping reserves
  - multi-stage stochastic approach w/scenario recombination
- Formulation presented here in 2013 and published in [1] and some preliminary testing results presented here in 2015.

[1] Carlos E. Murillo-Sánchez, Ray D. Zimmerman, C. Lindsay Anderson and Robert J. Thomas, "Secure Planning and Operations of Systems with Stochastic Sources, Energy Storage and Active Demand", *Smart Grid, IEEE Transactions on*, vol.4, no.4, pp.2220-2229, Dec. 2013. Available: http://dx.doi.org/10.1109/TSG.2013.2281001

MOST Continuous Single Period Problems



MOPS Mixed Integer and Multi-Period Problems



Outline

- MATPOWER Background
- MOST Overview
  - What is MOST?
  - Problem Formulation
- MOST Examples
  - Single-period continuous variable problems
  - Multi-interval problems with UC

Objective Function

$$\min_x f(x) \quad \text{where}$$

$$f(x) = f_p(p, p_+, p_-)$$ expected active power (re)dispatch costs

$$+ f_z(r_z)$$ zonal reserve costs

$$+ f_r(r_+, r_-)$$ contingency reserve costs

$$+ f_\delta(p)$$ expected ramping wear & tear costs

$$+ f_{\text{lf}}(\delta_+, \delta_-)$$ load following ramp reserve costs

$$+ f_s(s_0, p_{\text{sc}}, p_{\text{sd}})$$ expected value of leftover stored energy

$$+ f_{\text{uc}}(u, v, w)$$ startup, shutdown & no load costs

Constraints

(1) standard OPF constraints
- power balance equations
- transmission flow limits, other OPF inequalities

(2) security constraints
- fixed zonal reserve requirements, or
- contingency constraints
  - reserve, redispatch and contract variables
  - ramping limits on transitions from base to contingency cases

(3) intertemporal constraints
- load following ramping limits and reserves
- energy storage constraints

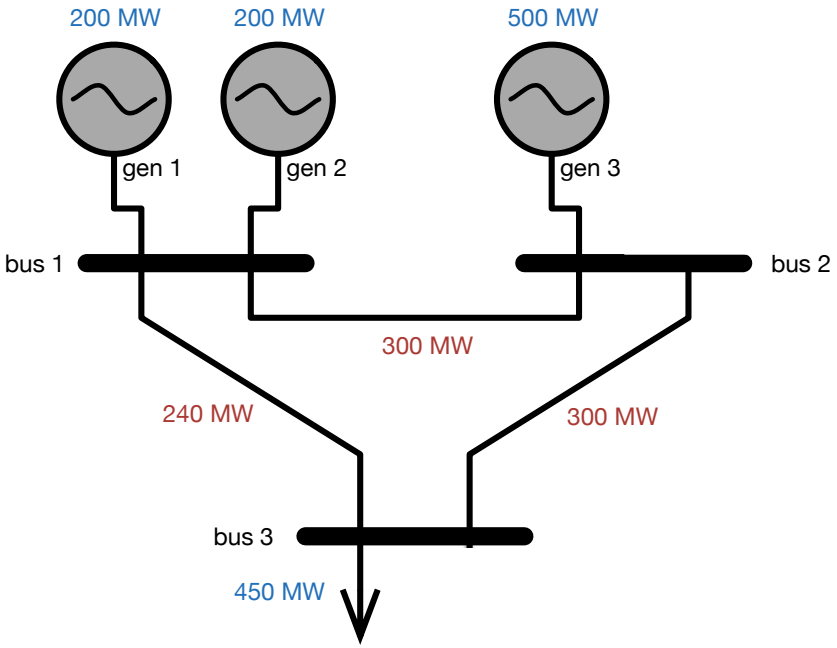
(4) unit commitment constraints
- injection limits vs. commitment variables
- startup/shutdown events
- minimum up/down times

Outline

- MATPOWER Background
- MOST Overview
  - What is MOST?
  - Problem Formulation
- MOST Examples
  - Single-period continuous variable problems
  - Multi-interval problems with UC

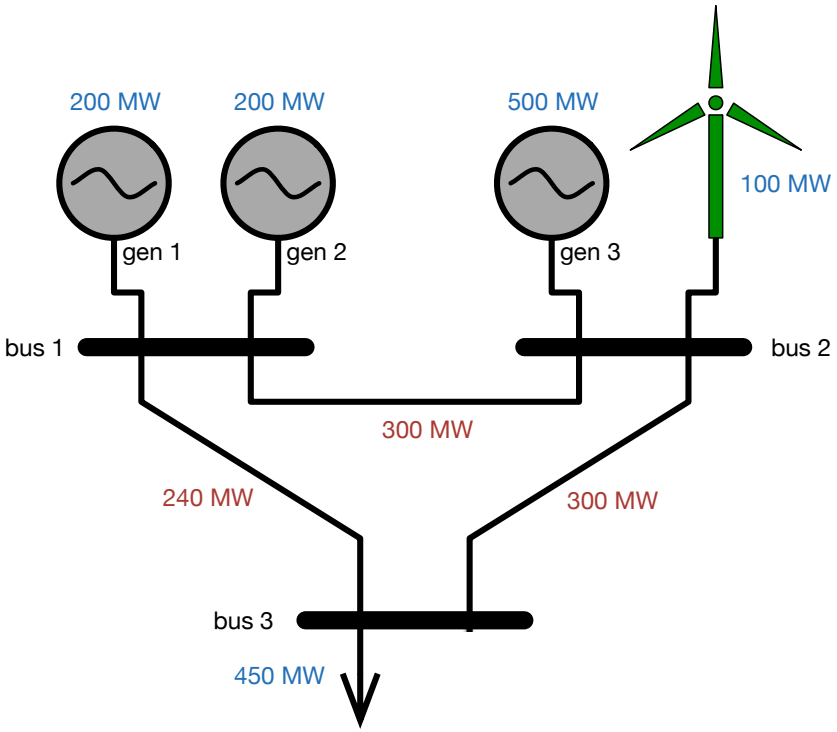
Tutorial Example System

- 3-bus triangle network
- generators
  - 2 identical 200 MW gens at bus 1, diff reserve cost
  - 500 MW gen at bus 2
  - all 3 have identical quadratic generation costs
- load 450 MW at bus 3, curtailable @ $1000
- branches
  - 300 MW limit, line 1–2
  - 240 MW limit, line 1–3
  - 300 MW limit, line 2–3
- adequacy requirement
  - reserve requirement
    - 150 MW limit
  - contingencies
    - generator 2 at bus 1
    - line 1–3

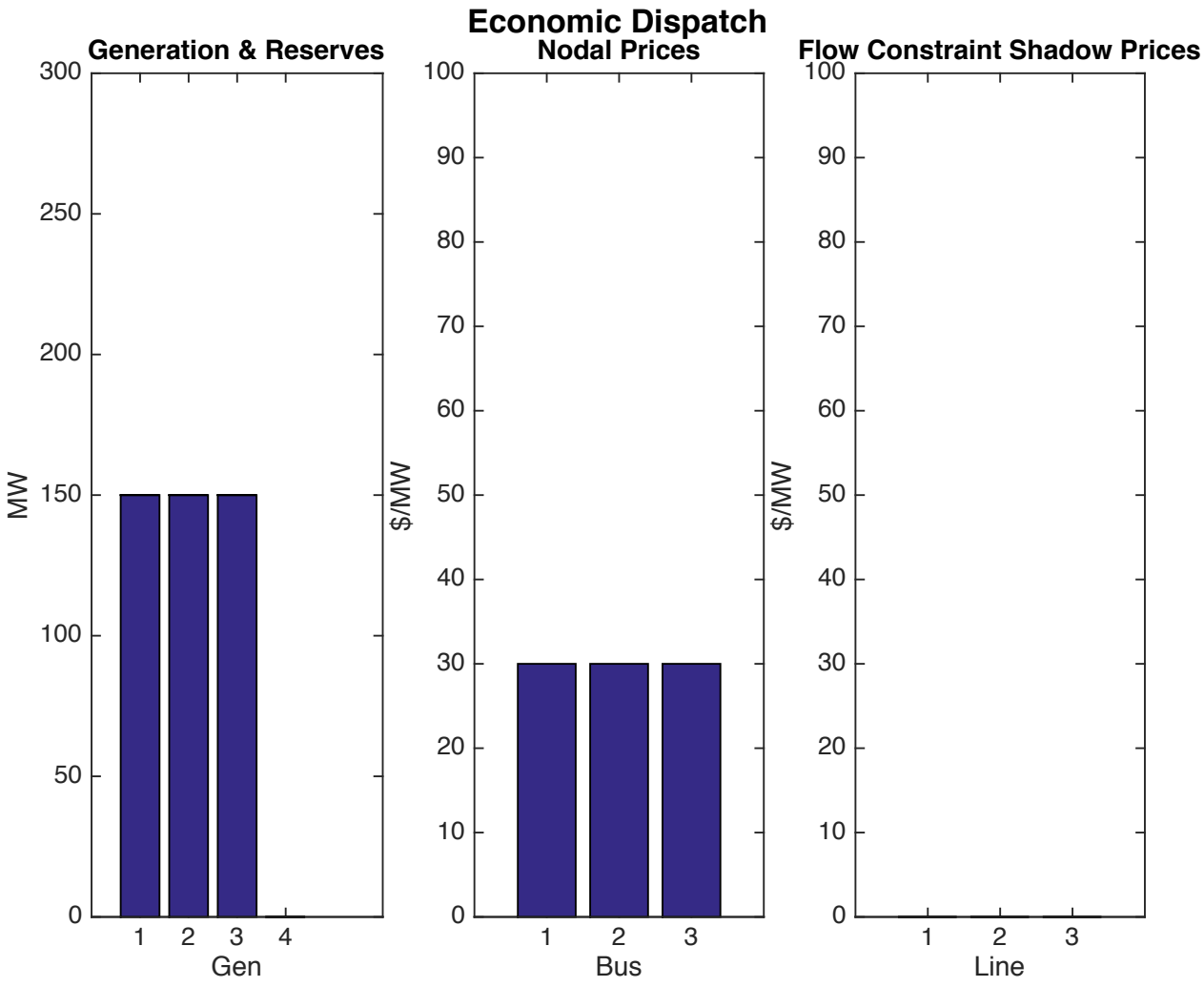


Tutorial Example System

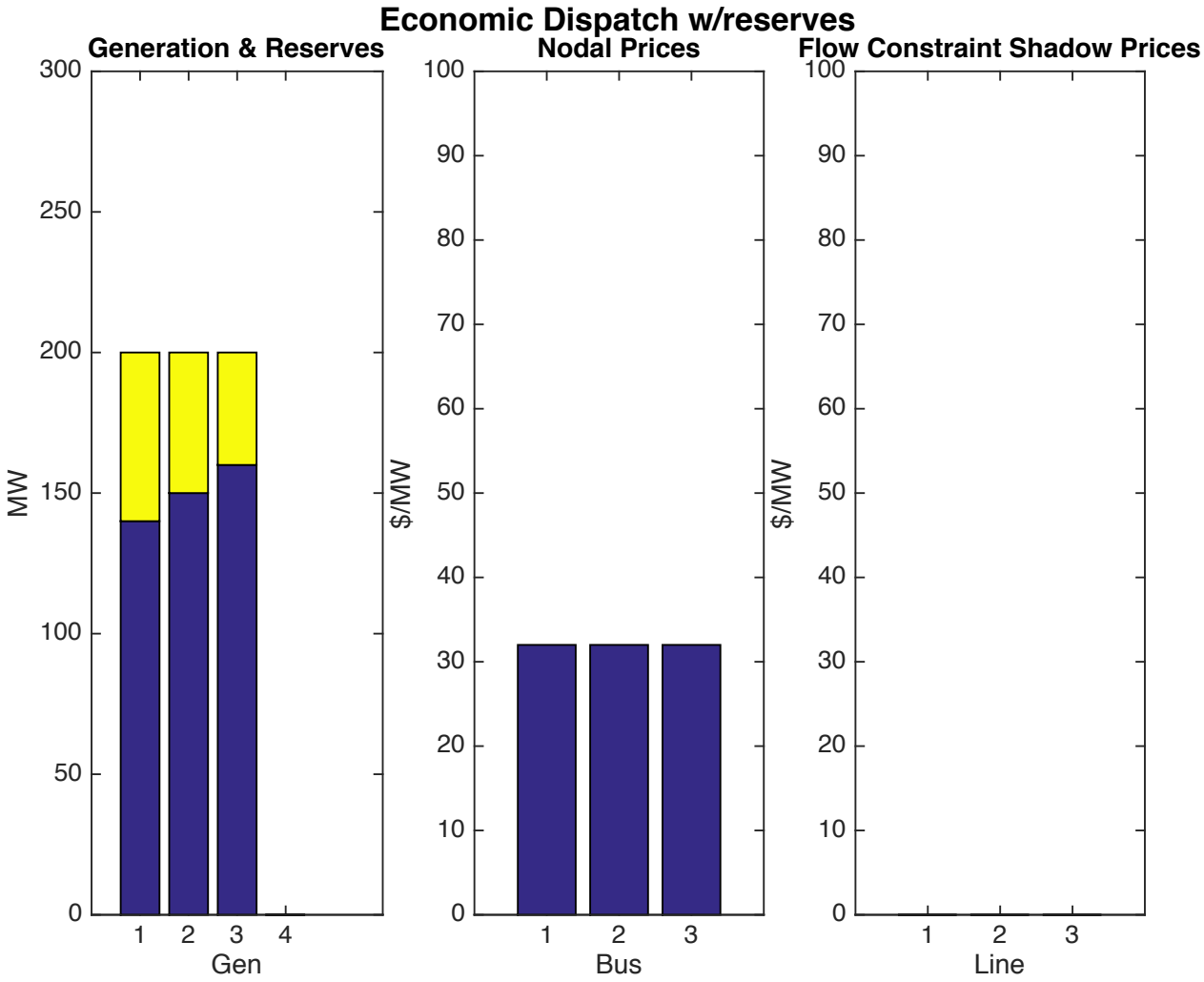
- 3-bus triangle network
- generators
  - 2 identical 200 MW gens at bus 1, diff reserve cost
  - 500 MW gen at bus 2
  - all 3 have identical quadratic generation costs
- load 450 MW at bus 3, curtailable @ $1000
- branches
  - 300 MW limit, line 1–2
  - 240 MW limit, line 1–3
  - 300 MW limit, line 2–3
- adequacy requirement
  - reserve requirement
    - 150 MW limit
  - contingencies
    - generator 2 at bus 1
    - line 1–3
- wind
  - 100 MW unit at bus 2
  - 3 samples of normal distribution around 50 MW



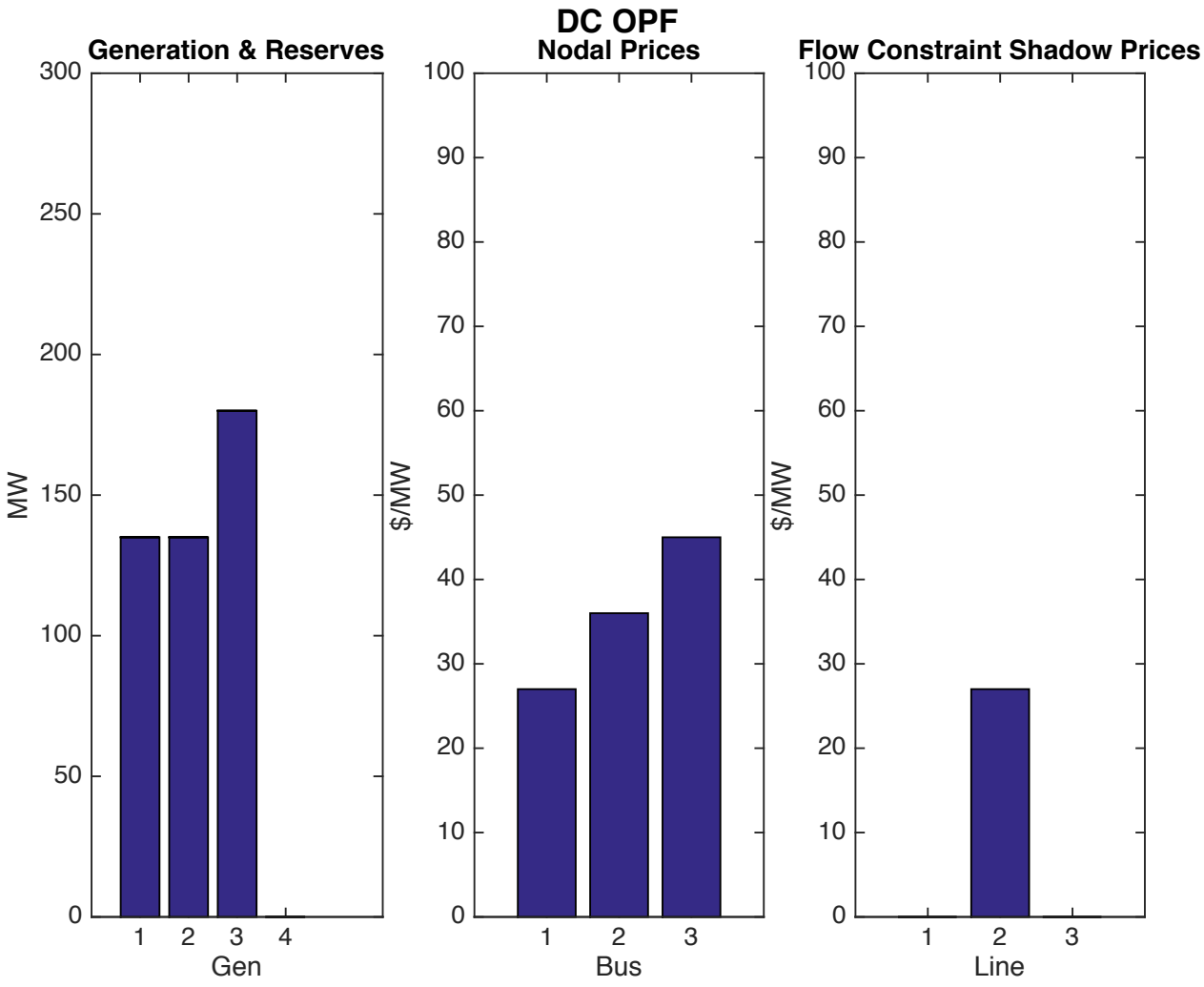
Single Period Continuous Examples



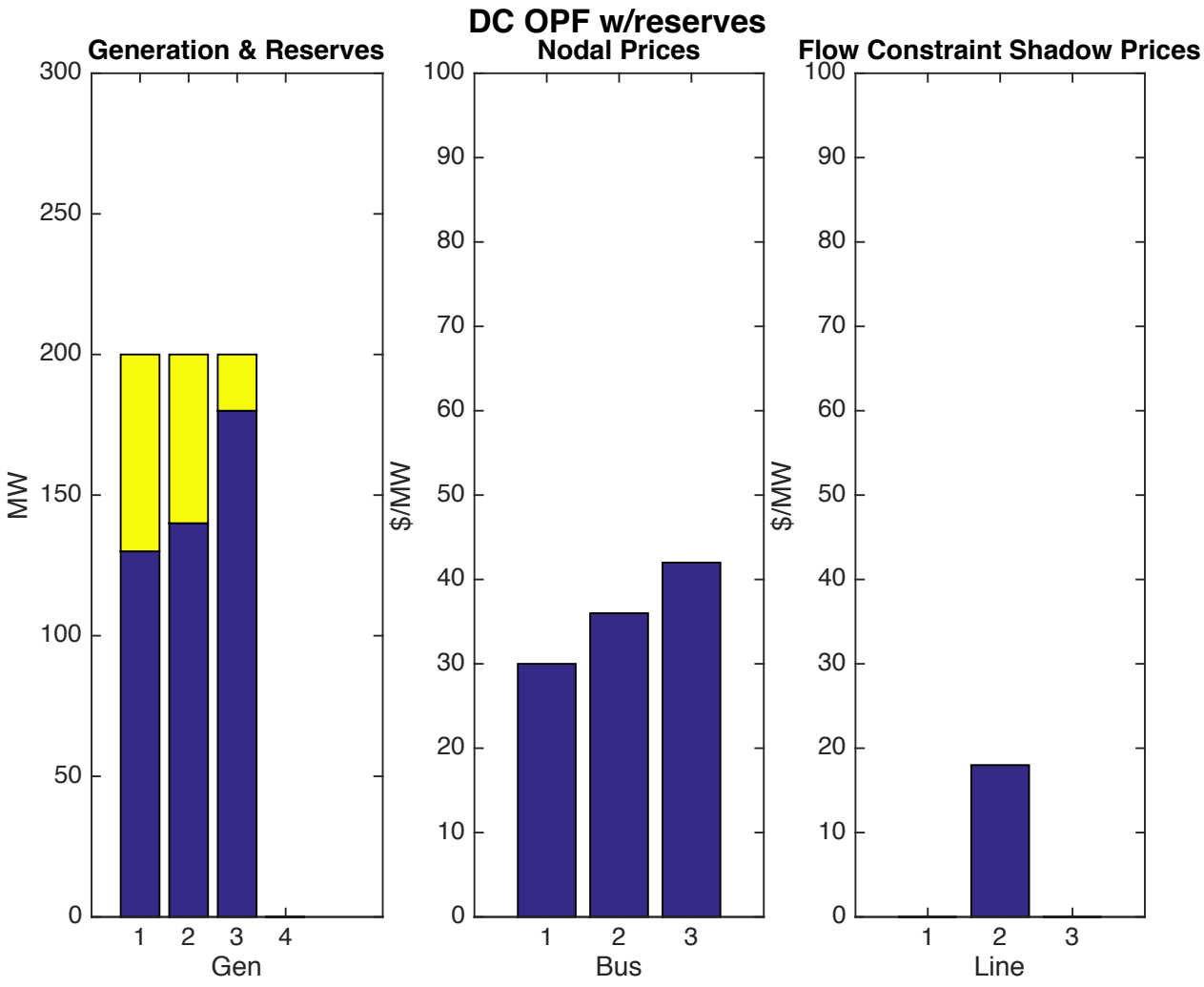
Single Period Continuous Examples



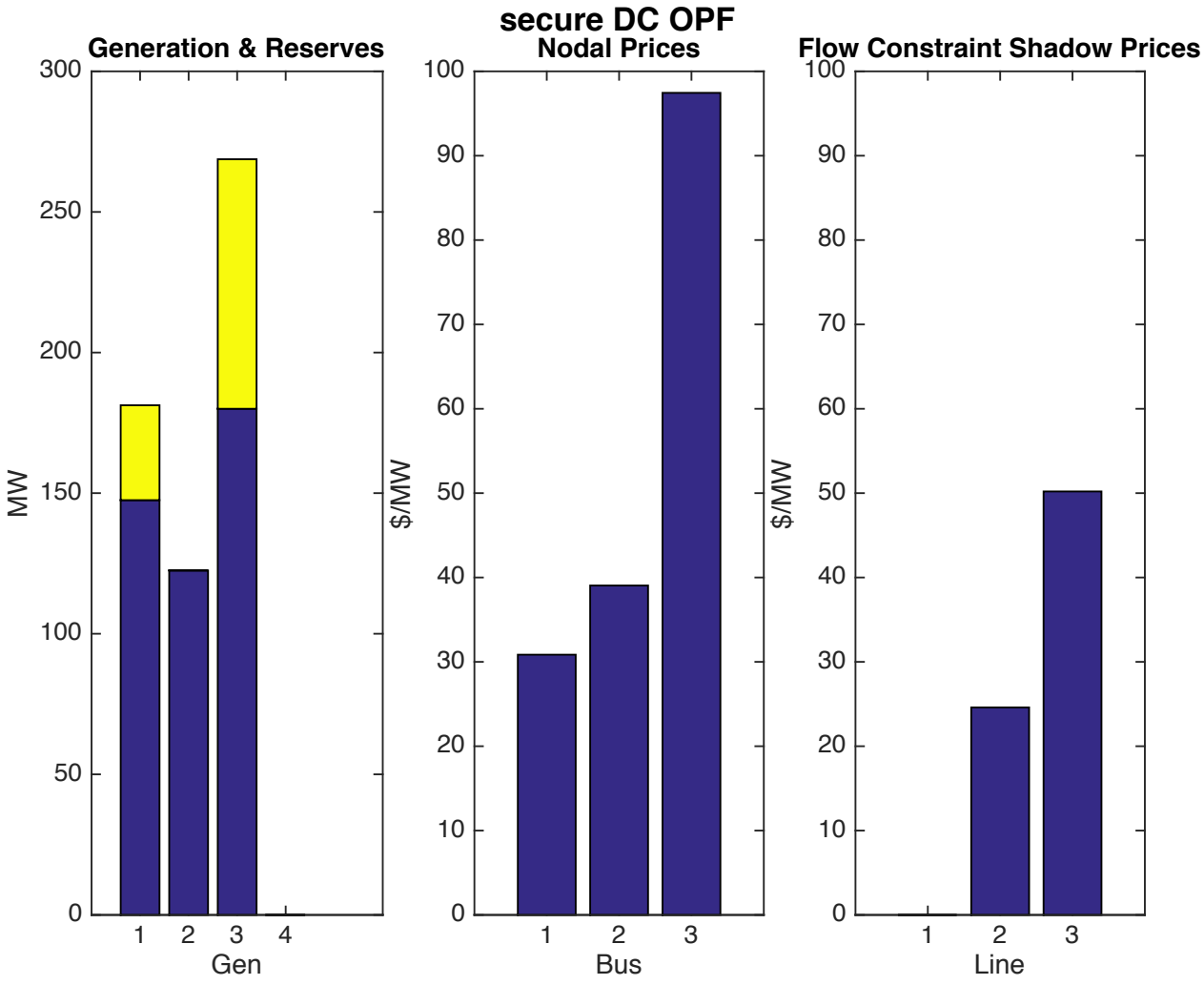
Single Period Continuous Examples



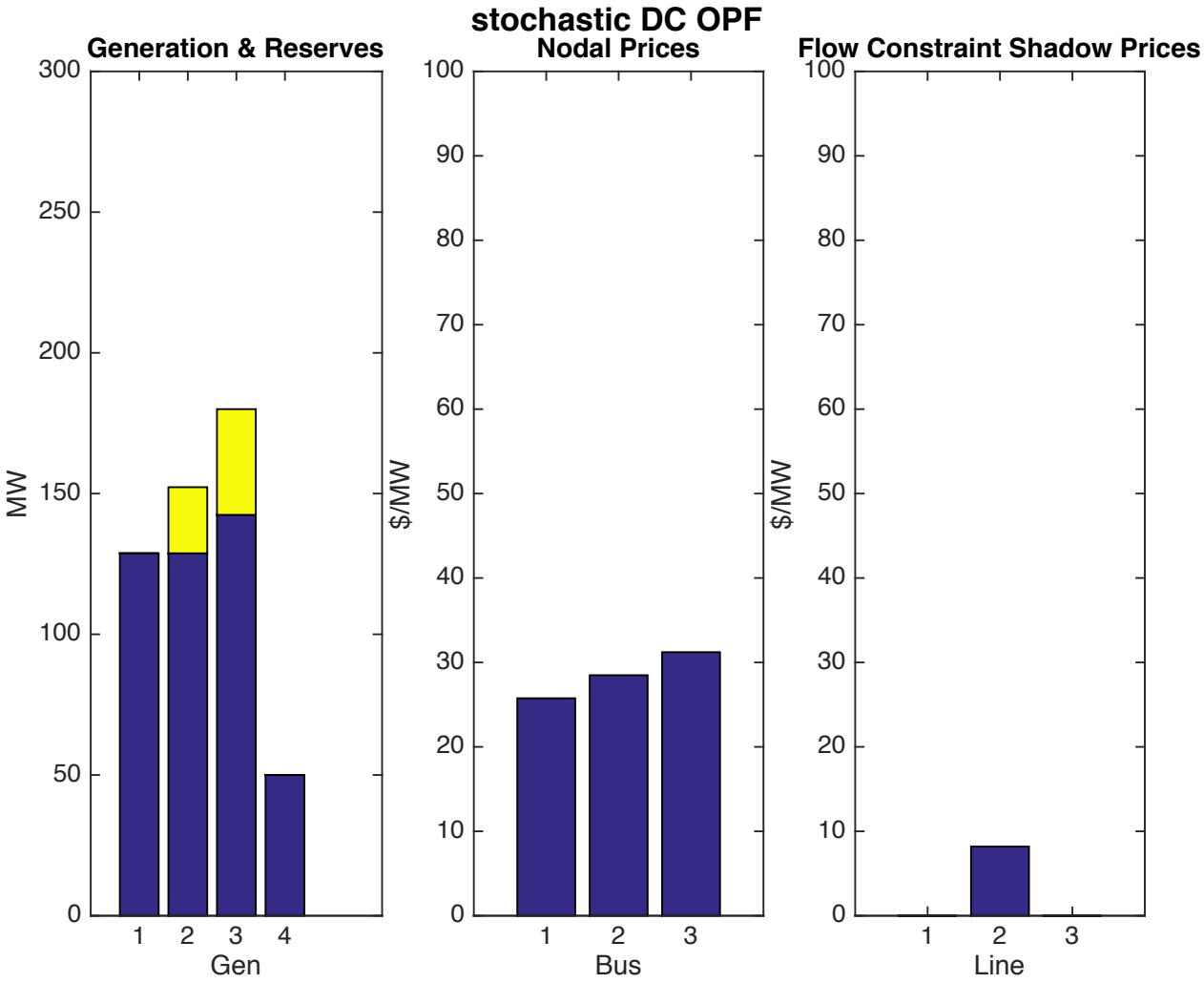
Single Period Continuous Examples



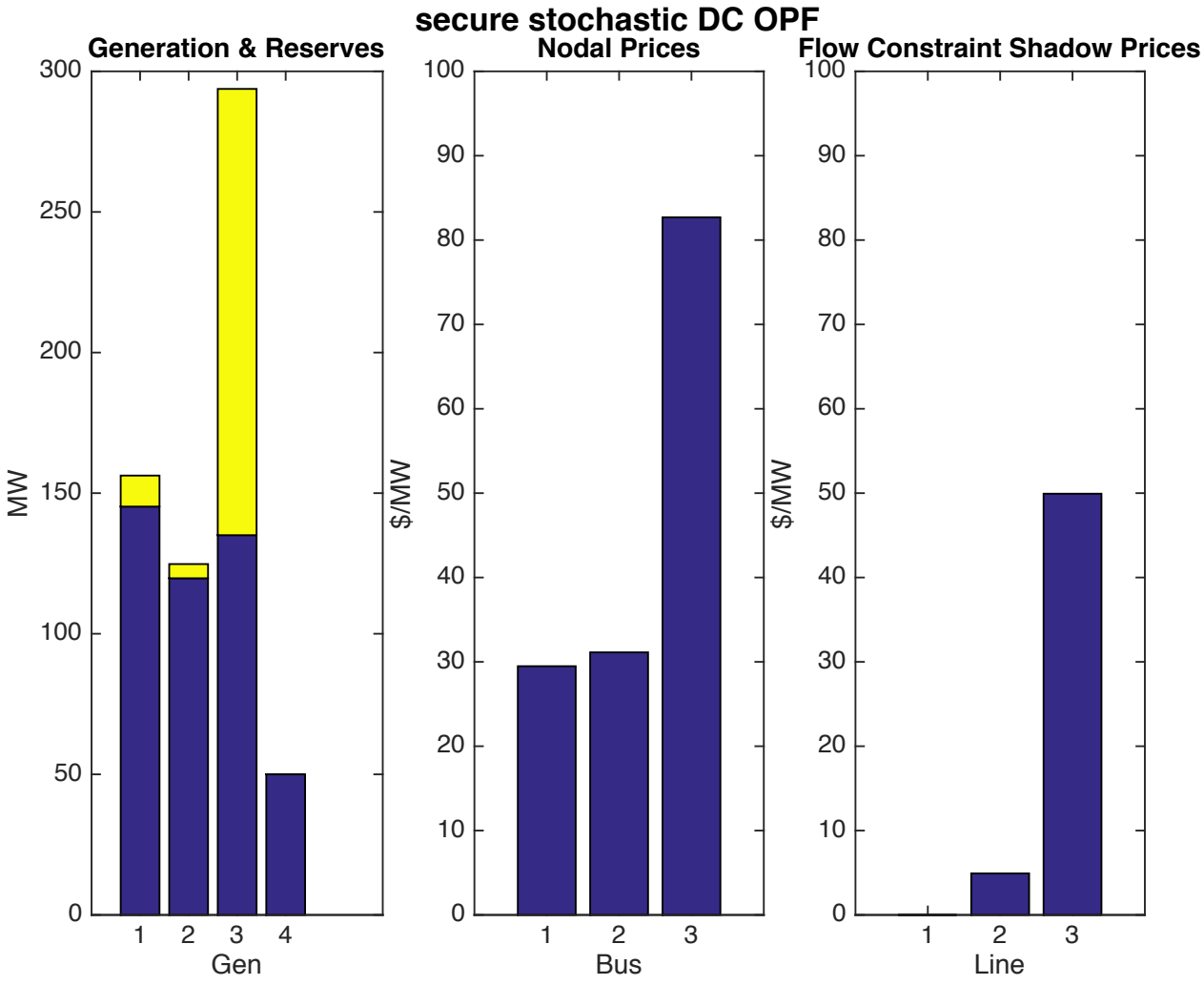
Single Period Continuous Examples



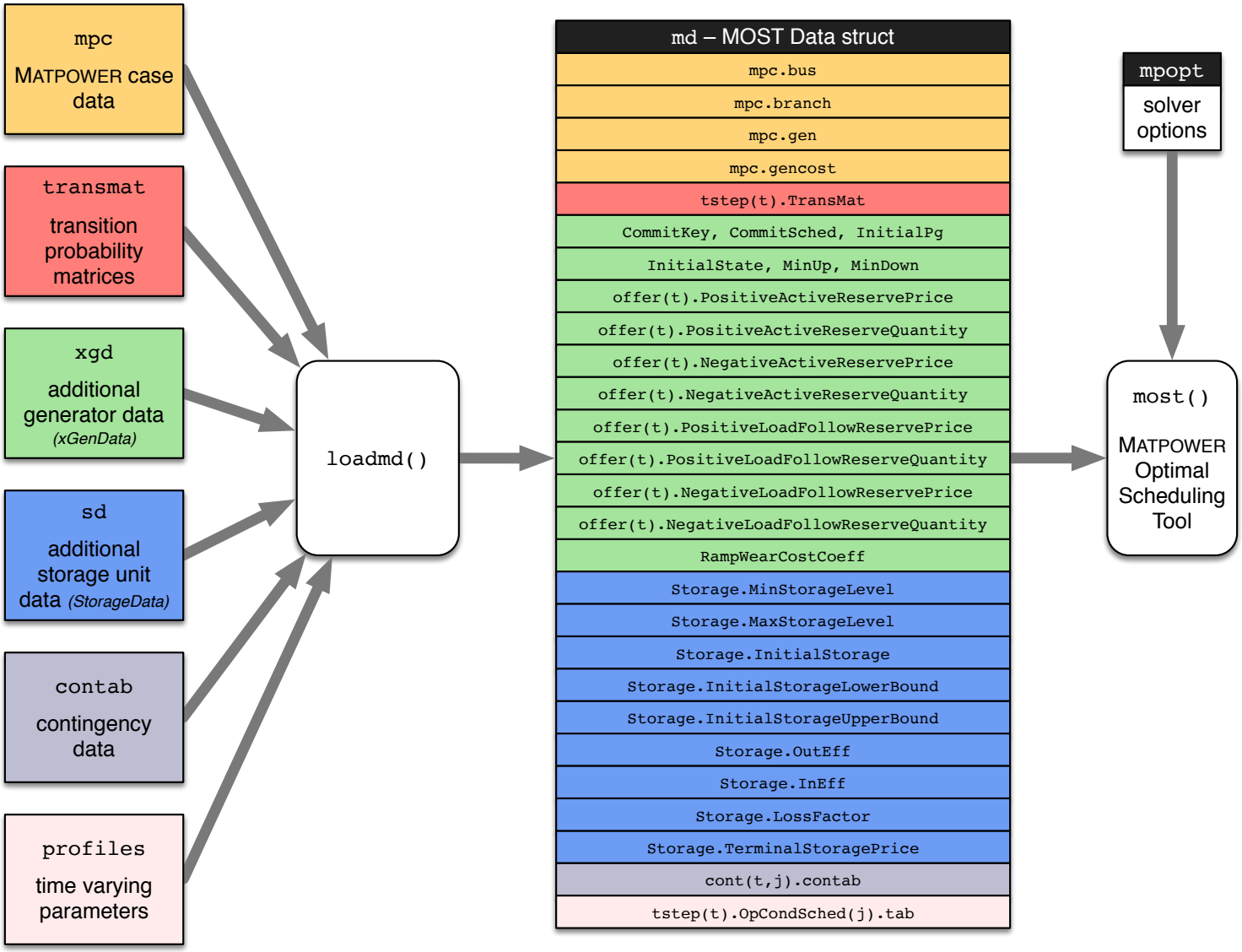
Single Period Continuous Examples



Single Period Continuous Examples



Input Data


mpc
MATPOWER case data
transmat
transition probability matrices
xgd
additional generator data
(xGenData)
sd
additional storage unit data (StorageData)
contab
contingency data
profiles
time varying parameters
loadmd()
md – MOST Data struct
mpc.bus
mpc.branch
mpc.gen
mpc.gencost
tstep(t).TransMat
CommitKey, CommitSched, InitialPg
InitialState, MinUp, MinDown
offer(t).PositiveActiveReservePrice
offer(t).PositiveActiveReserveQuantity
offer(t).NegativeActiveReservePrice
offer(t).NegativeActiveReserveQuantity
offer(t).PositiveLoadFollowReservePrice
offer(t).PositiveLoadFollowReserveQuantity
offer(t).NegativeLoadFollowReservePrice
offer(t).NegativeLoadFollowReserveQuantity
RampWearCostCoeff
Storage.MinStorageLevel
Storage.MaxStorageLevel
Storage.InitialStorage
Storage.InitialStorageLowerBound
Storage.InitialStorageUpperBound
Storage.OutEff
Storage.InEff
Storage.LossFactor
Storage.TerminalStoragePrice
cont(t,j).contab
tstep(t).OpCondSched(j).tab
mpopt
solver options
most()
MATPOWER Optimal Scheduling Tool

Example Code

- DC OPF

```
mdi = loadmd('ex_case3a');
mdo = most(mdi);
```

- Economic Dispatch

```
mpopt = mpoption('most.dc_model', 0);
mdo = most(mdi, mpopt);
```

- With zonal reserves

```
reserves.zones  = [1 1 1 0];
reserves.req    = 150;
reserves.cost   = [1; 3; 5];
reserves.qty    = [100; 100; 200];
mdi.FixedReserves  = reserves;
mpopt = mpoption('most.dc_model', 1);
mdo = most(mdi, mpopt);
```

Secure DC OPF

- Defining contingencies: example contab

```
function contab = ex_contab
define_constants;
% label probty  type         row column      chgtype newvalue
contab = [
    1   0.06    CT_TGEN      2   GEN_STATUS   CT_REP  0;    %% gen 2 at bus 1
    2   0.04    CT_TBRCH     2   BR_STATUS    CT_REP  0;    %% line 1-3
];
```

- Same mechanism used to modify a base case for each period in multi-interval optimization

“Extra” Generator Data - `xGenData`

- Generator parameters not included in base case data, e.g. unit commitment data, reserve offers, ramping costs, and more

```
function xgd_table = ex_xgd_res(mpc)
xgd_table.colnames = {
    'PositiveActiveReservePrice', ...
            'PositiveActiveReserveQuantity', ...
};
xgd_table.data = [
    5       250;
    0       100;
    1.5     600;
    0       800;
];
```

- Run the Secure DC OPF

```
mpc = loadcase('ex_case3a');
xgd = loadxgendata('ex_xgd_res', mpc);
mdi = loadmd('ex_case3a', [], xgd, [], 'ex_contab');
mdo = most(mdi, mpopt);
```

Outline

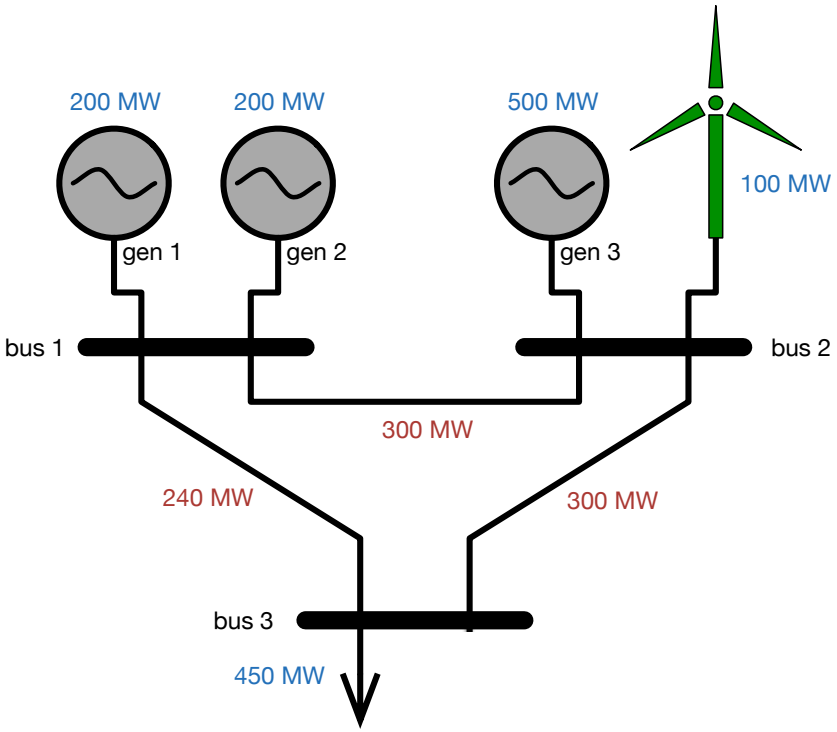
- MATPOWER Background
- MOST Overview
  - What is MOST?
  - Problem Formulation
- MOST Examples
  - Single-period continuous variable problems
  - Multi-interval problems with UC

Tutorial Example System

- 3-bus triangle network
- generators
  - 2 identical 200 MW gens at bus 1, diff reserve cost
  - 500 MW gen at bus 2
  - gen costs marginal startup shutdown

1	$25	$ 0	$ 0
2	$30	$ 200	$200
3	$40	$3000	$600

- load 300-540 MW at bus 3, curtailable @ $1000
- branches
  - 300 MW limit, line 1–2
  - 240 MW limit, line 1–3
  - 300 MW limit, line 2–3
- adequacy requirement
  - reserve requirement
    - 150 MW limit
  - contingencies
    - generator 2 at bus 1
    - line 1–3
- wind
  - 0-166 MW unit at bus 2
  - 3 samples of normal distribution around varying mean

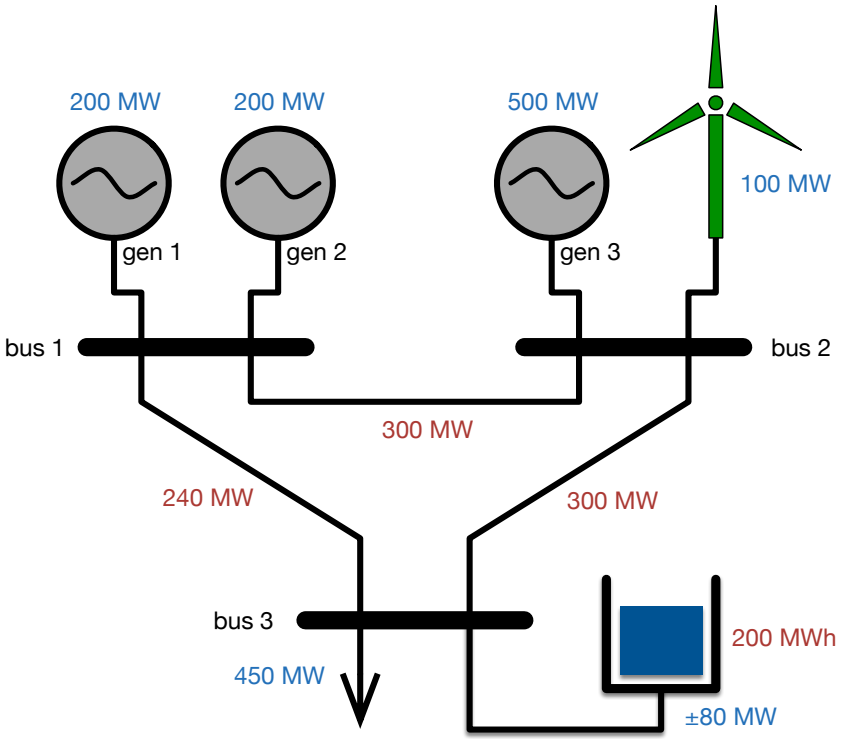


Tutorial Example System

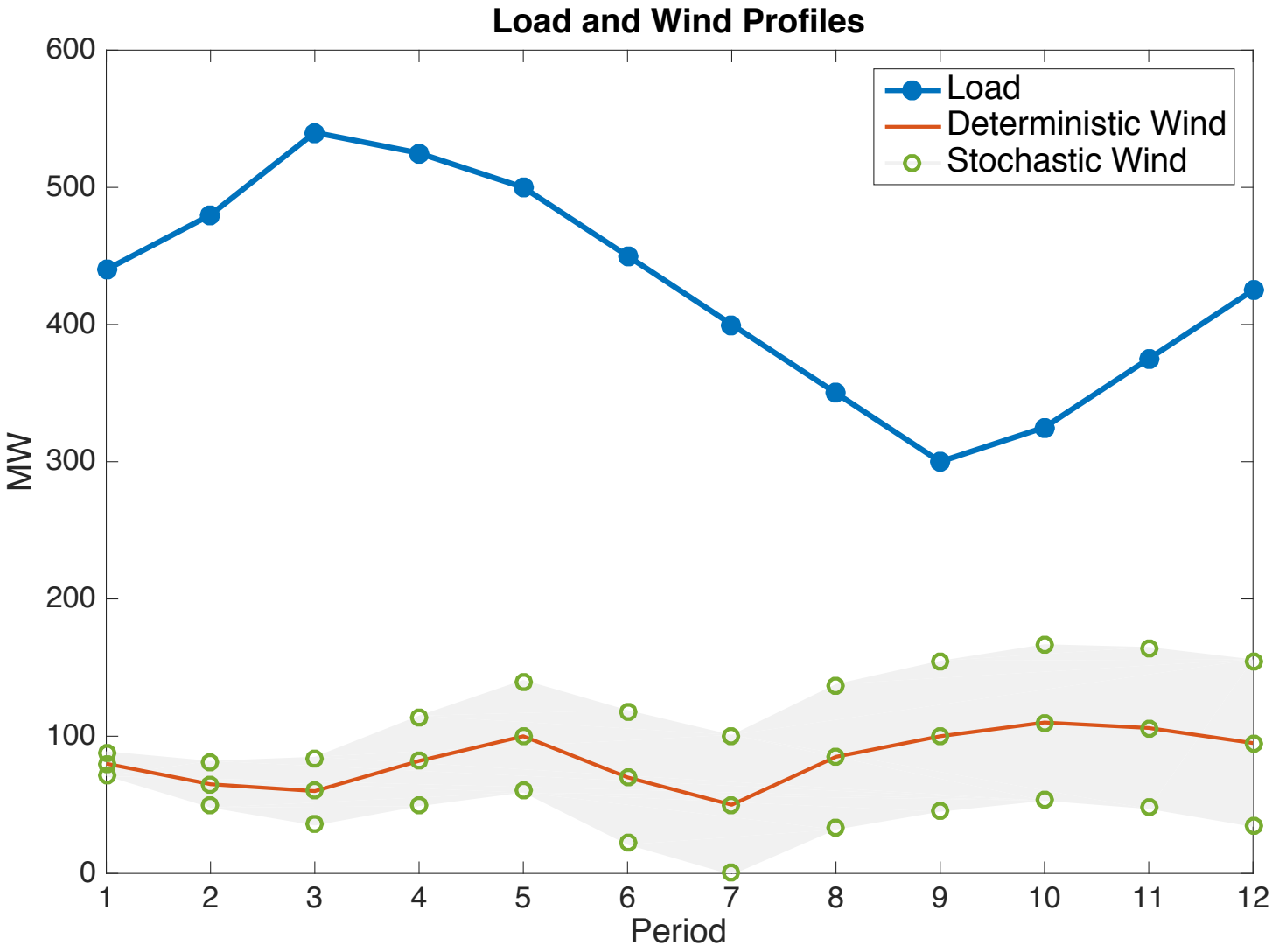
- 3-bus triangle network
- generators
  - 2 identical 200 MW gens at bus 1, diff reserve cost
  - 500 MW gen at bus 2
  - gen costs marginal startup shutdown

1	$25	$ 0	$ 0
2	$30	$ 200	$200
3	$40	$3000	$600

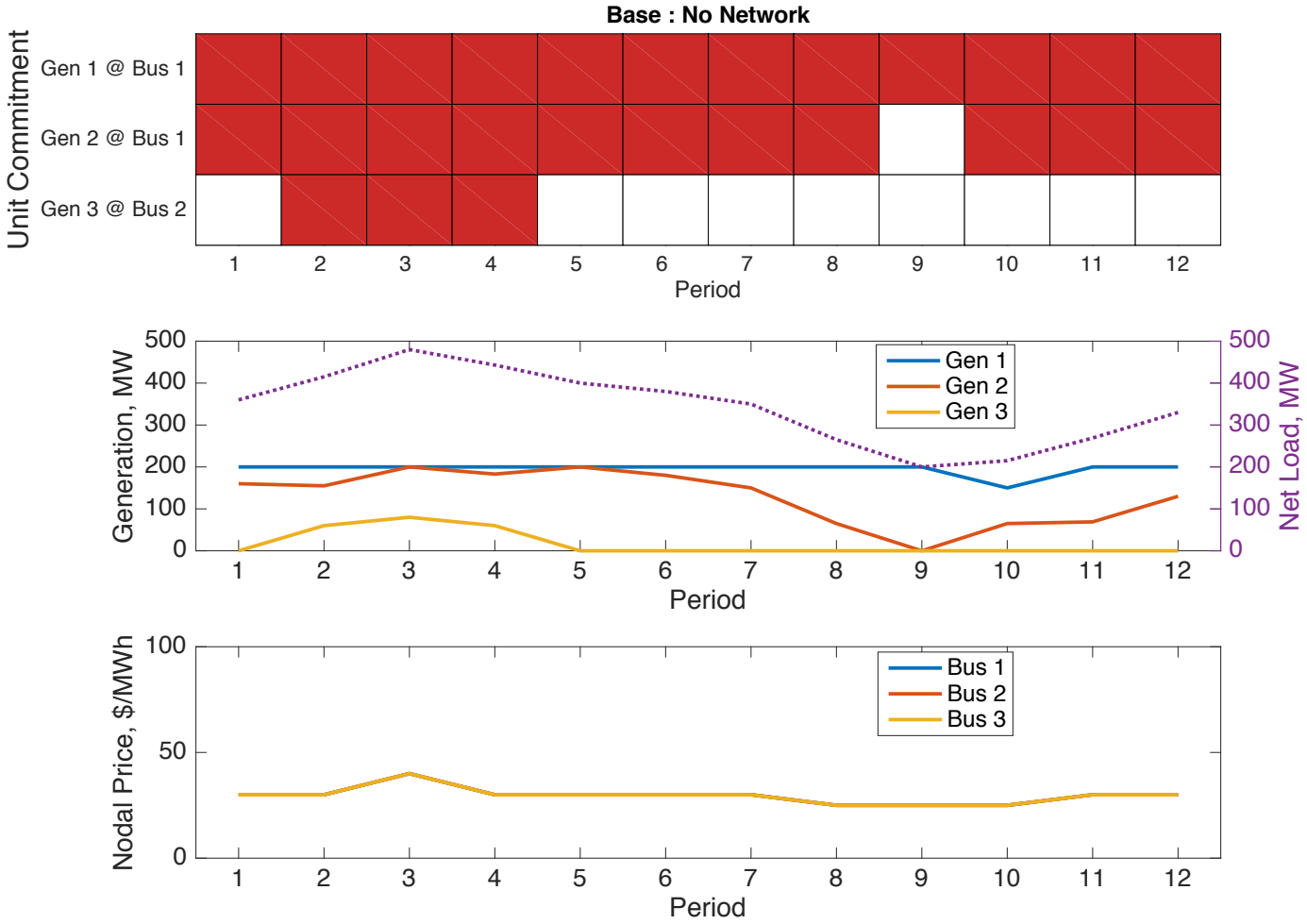
- load 300-540 MW at bus 3, curtailable @ $1000
- branches
  - 300 MW limit, line 1–2
  - 240 MW limit, line 1–3
  - 300 MW limit, line 2–3
- adequacy requirement
  - reserve requirement
    - 150 MW limit
  - contingencies
    - generator 2 at bus 1
    - line 1–3
- wind
  - 0-166 MW unit at bus 2
  - 3 samples of normal distribution around varying mean
- storage at bus 3
  - 200 MWh energy capacity, ±80 MW power capacity



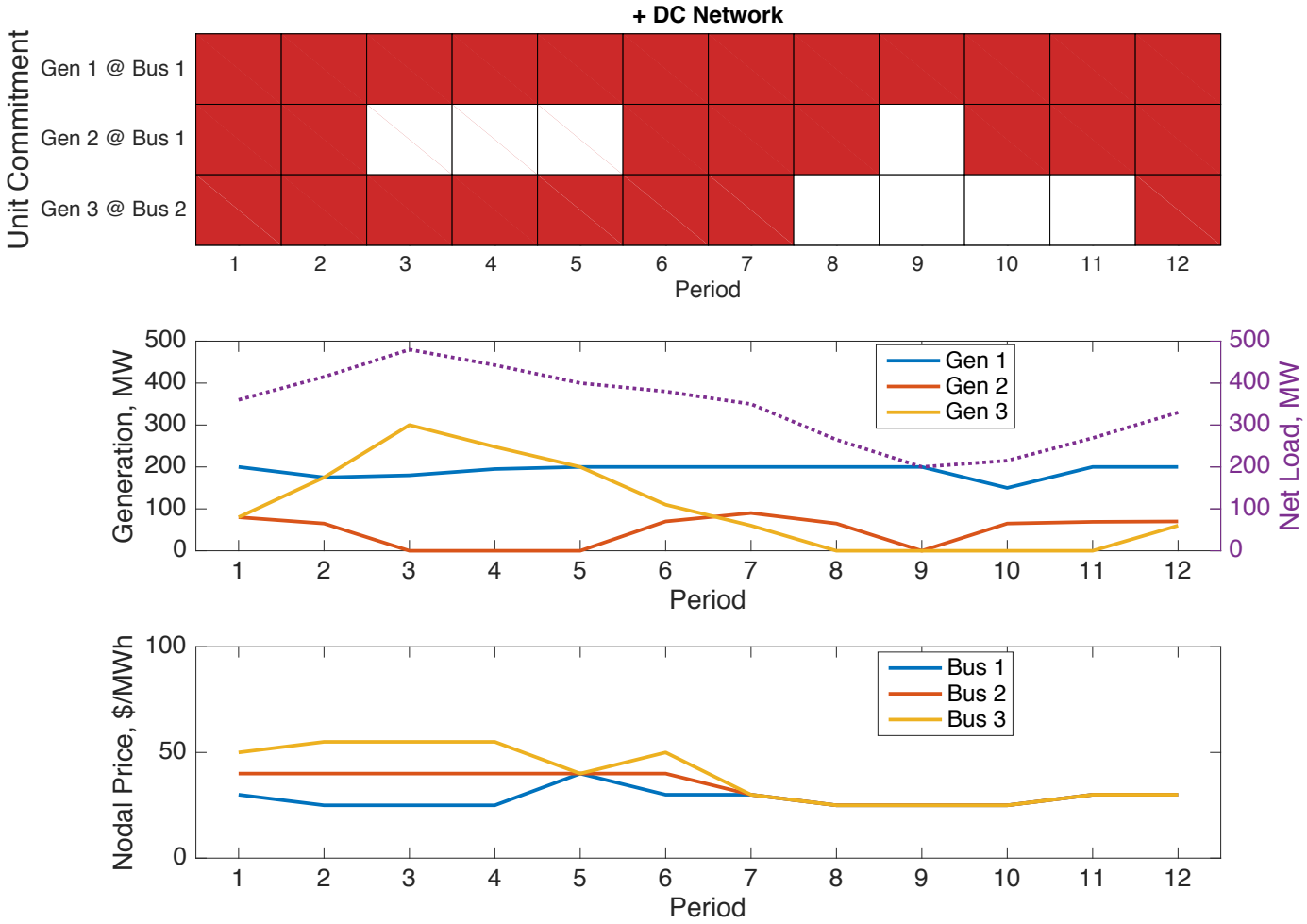
Mult-Interval (UC) Examples



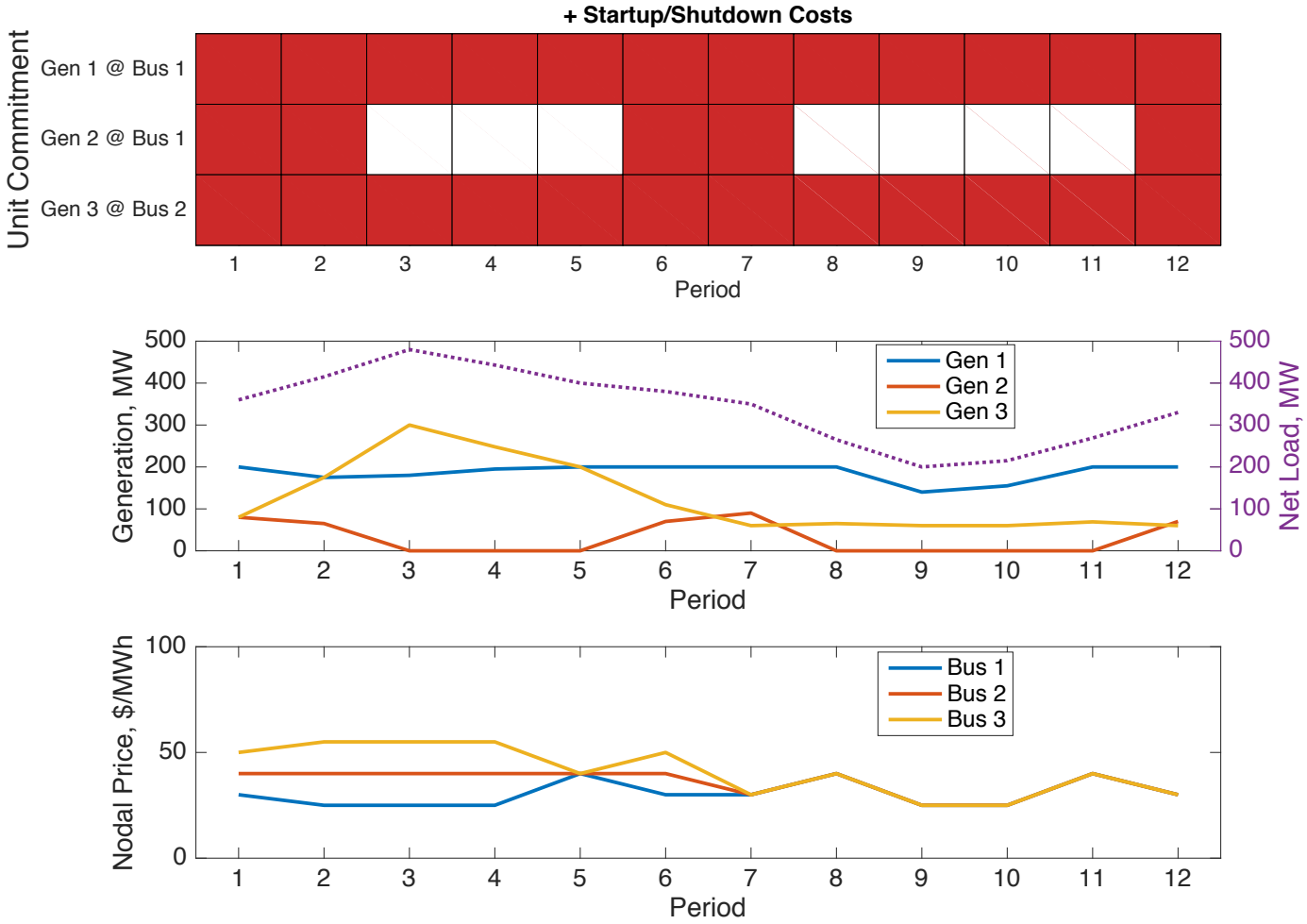
Deterministic UC Examples



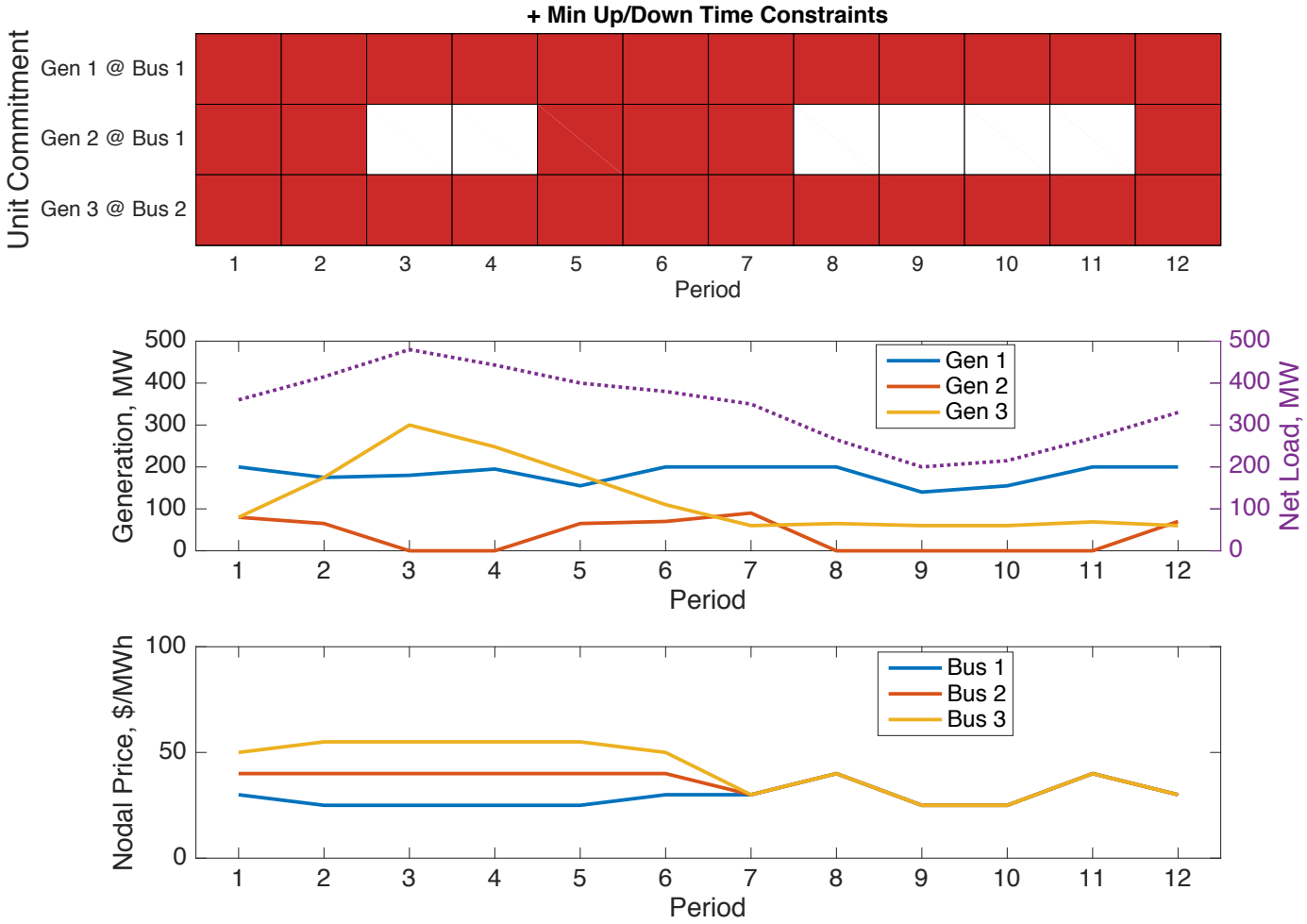
Deterministic UC Examples



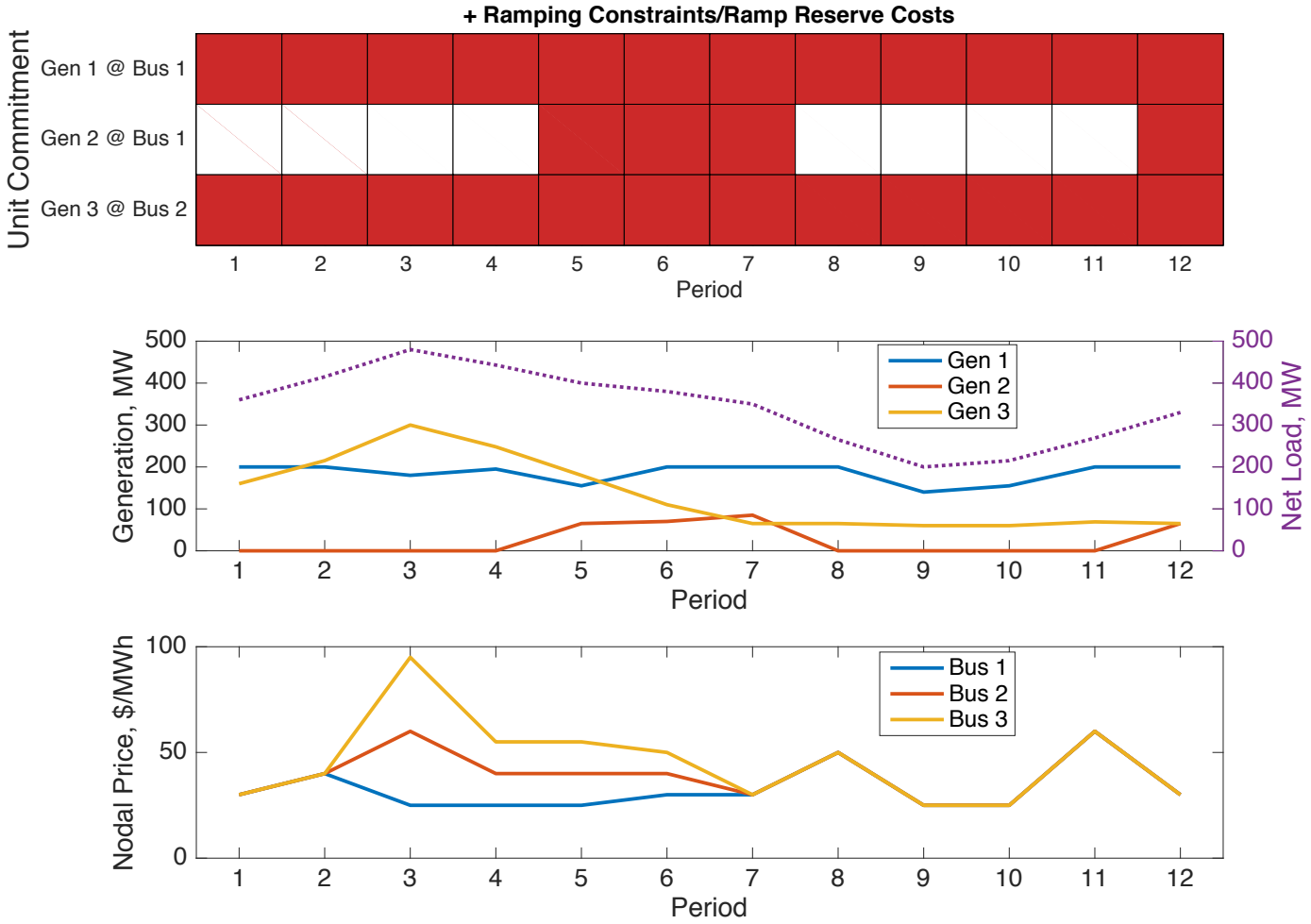
Deterministic UC Examples



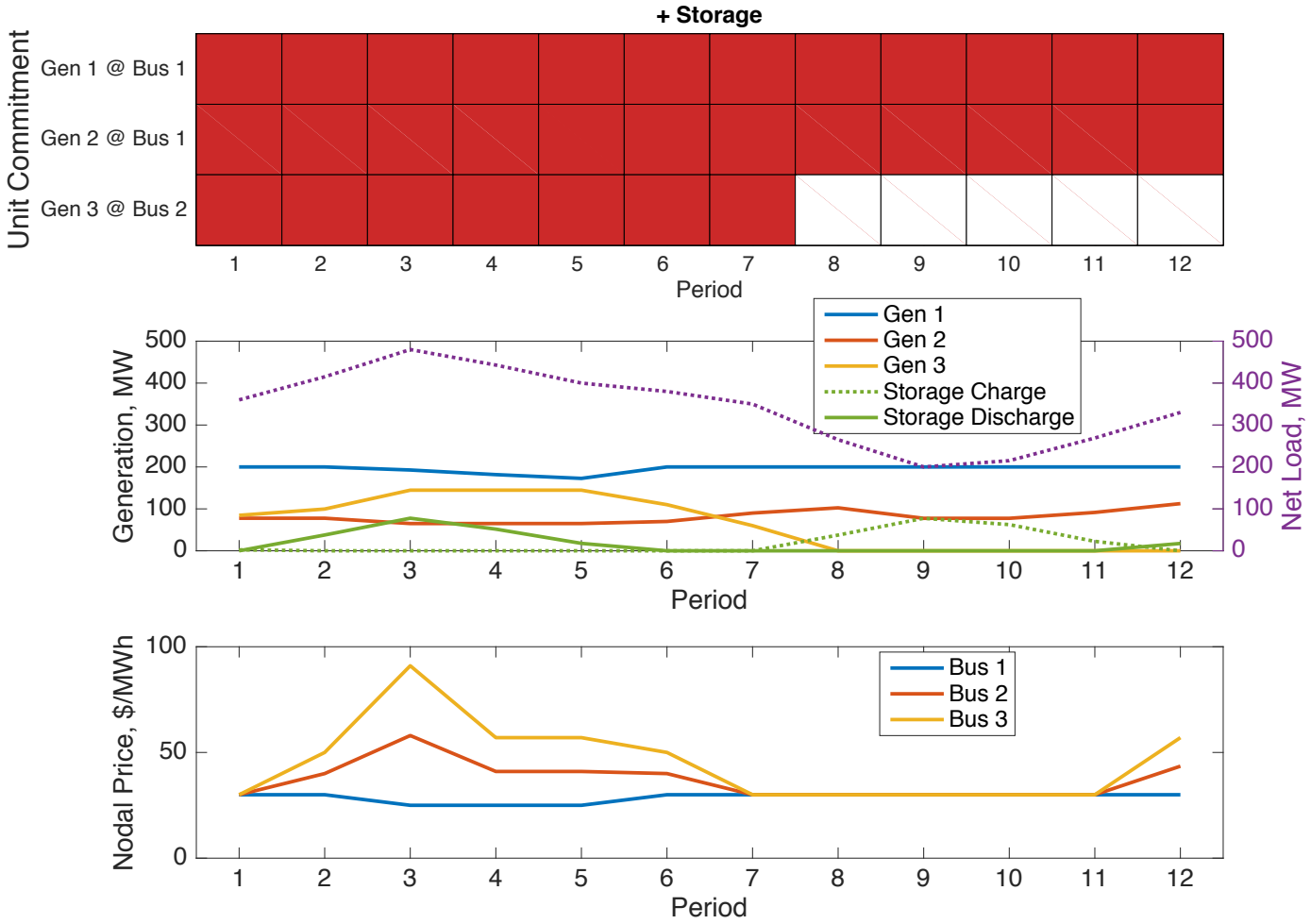
Deterministic UC Examples



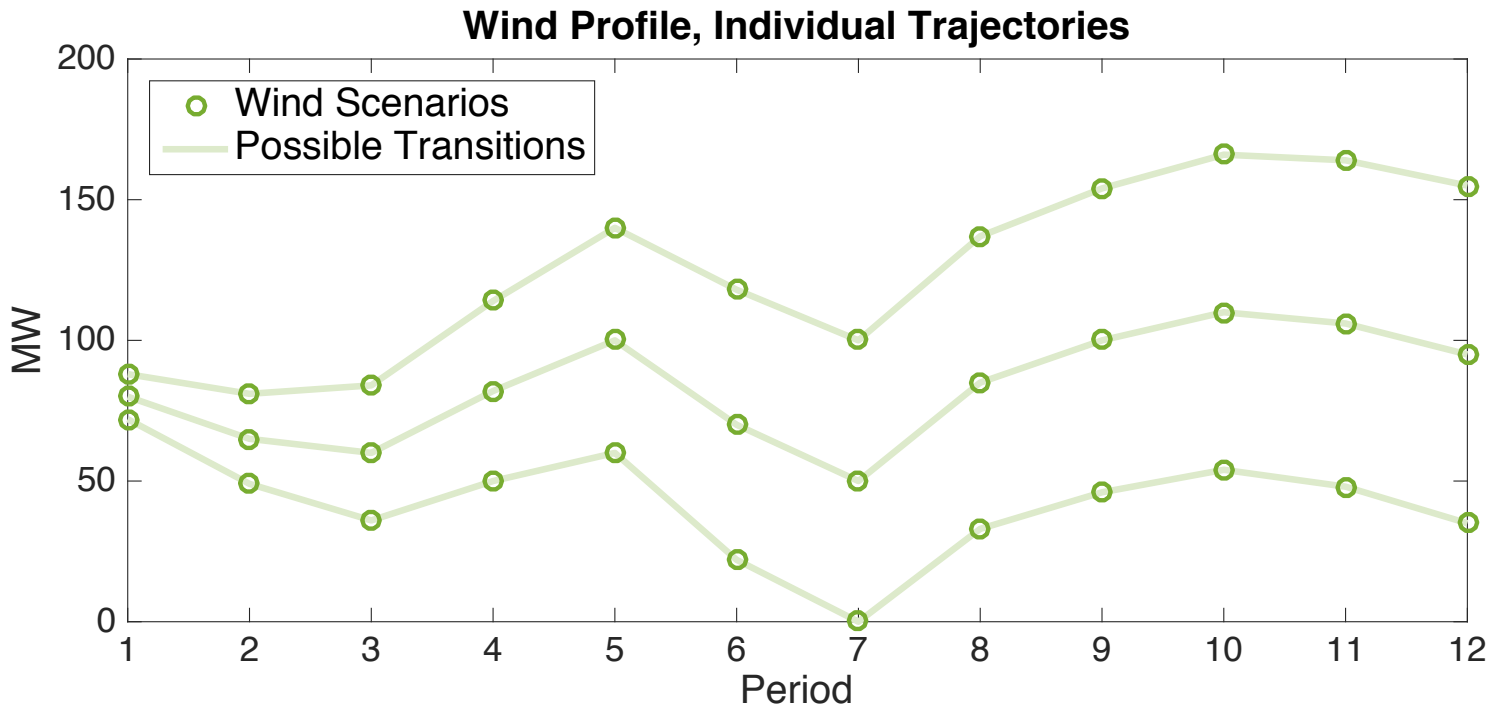
Deterministic UC Examples



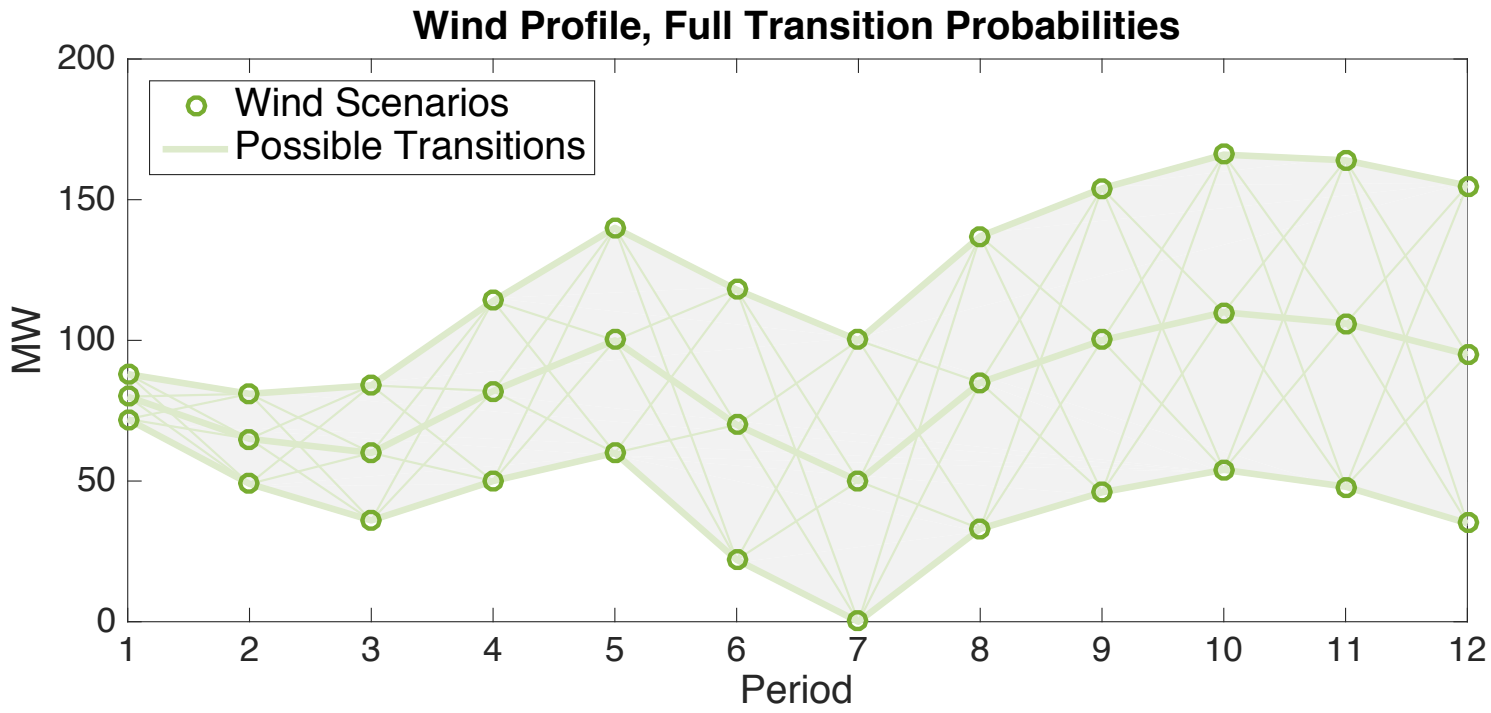
Deterministic UC Examples



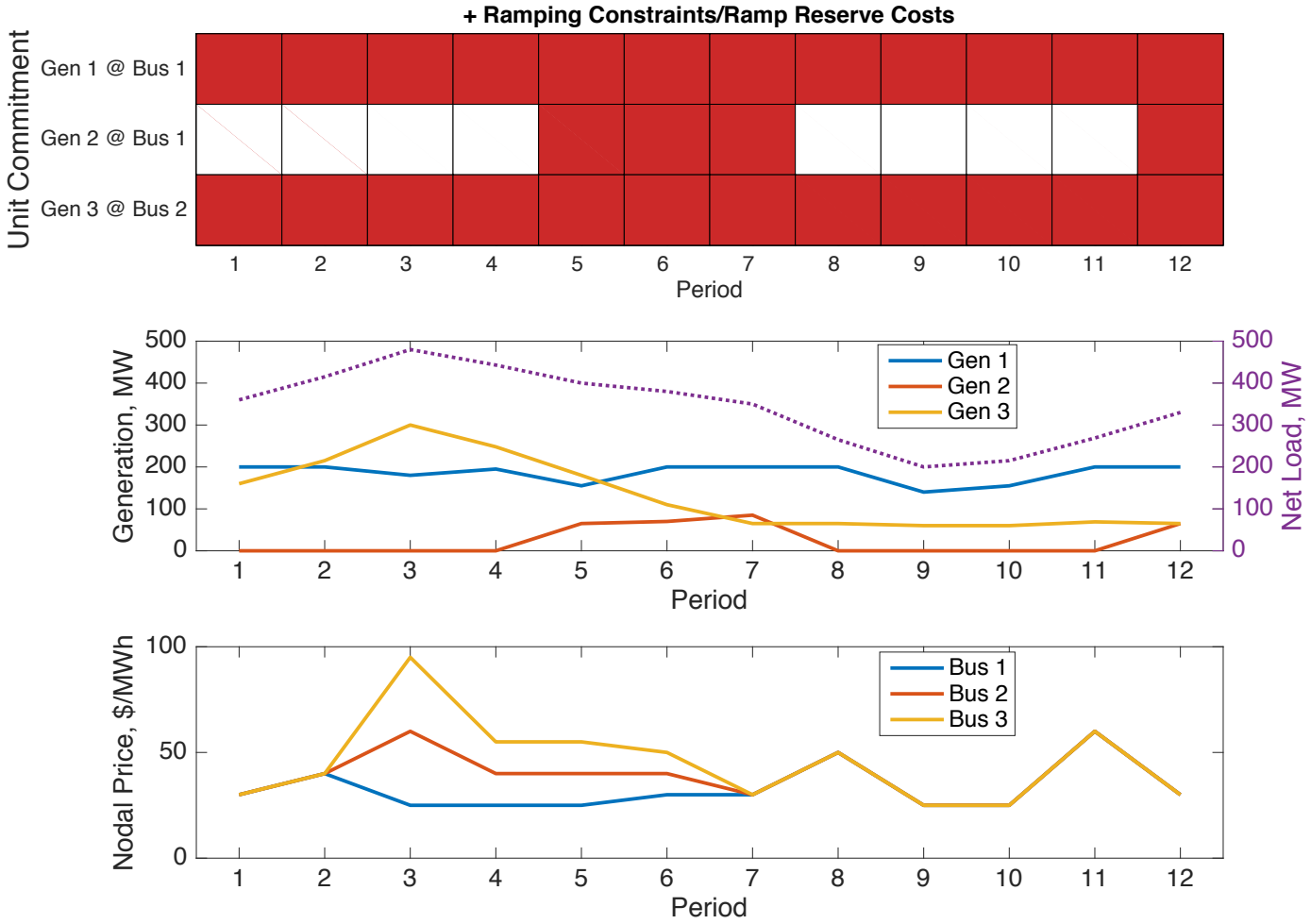
Stochastic Wind



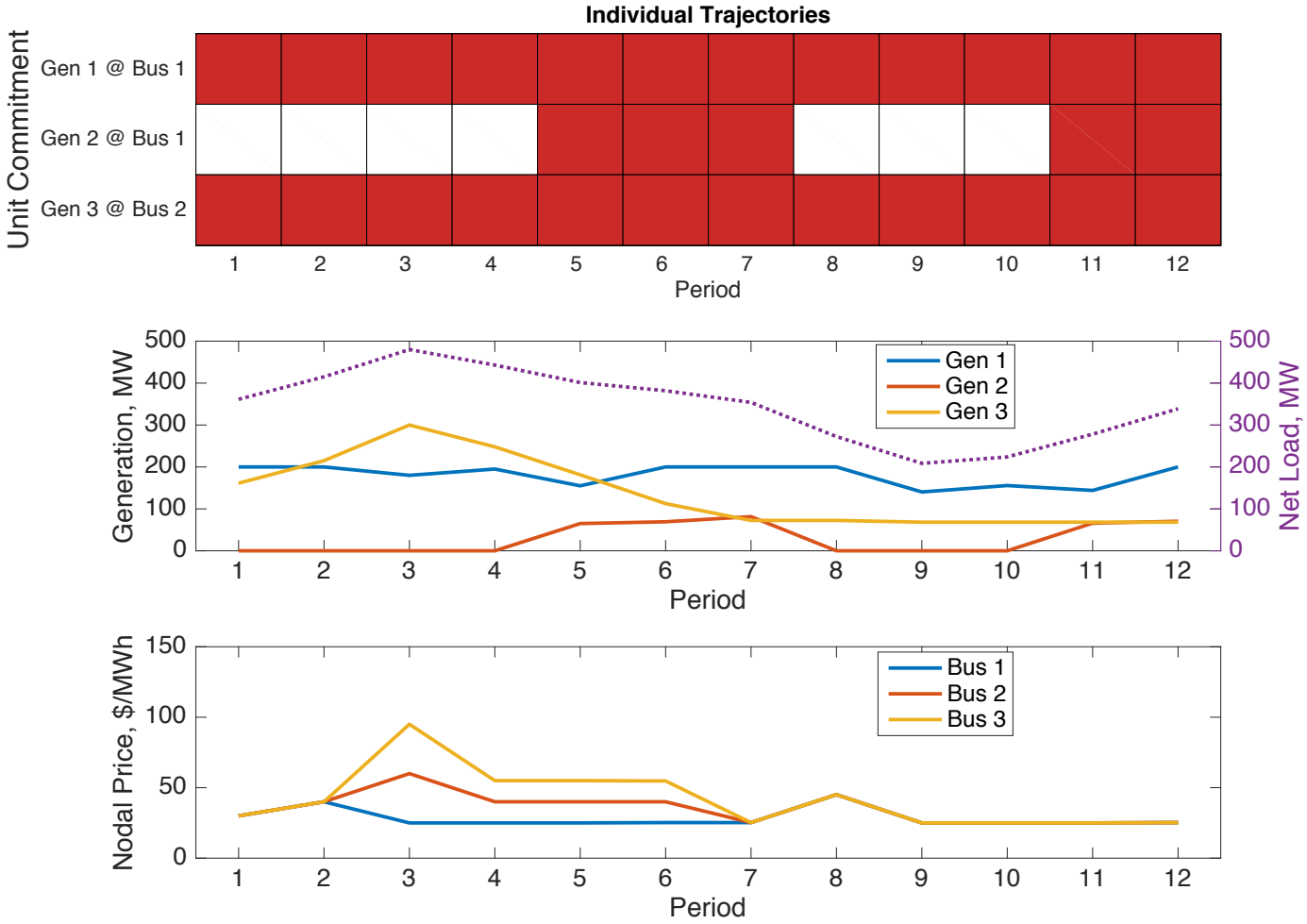
Stochastic Wind



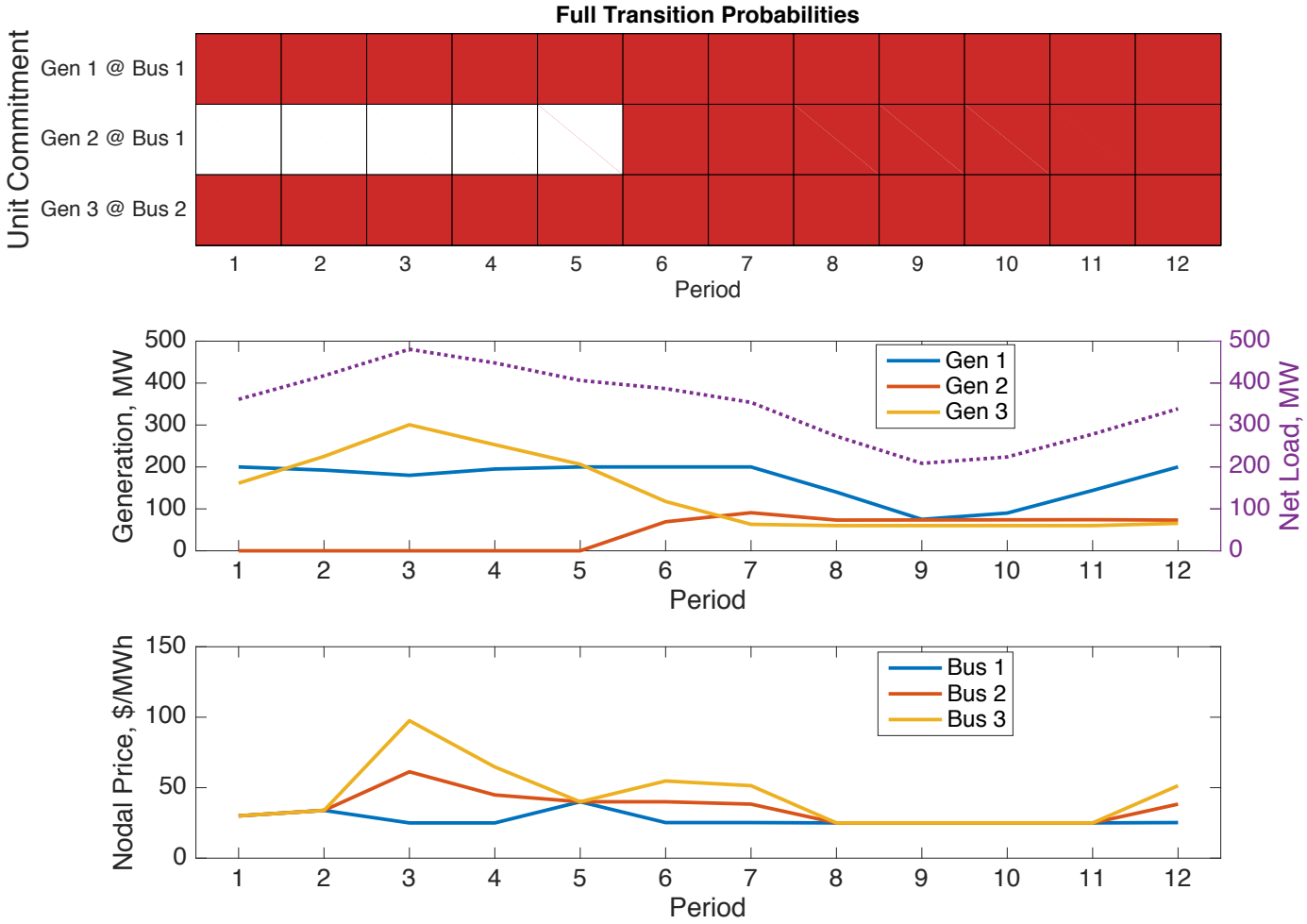
Deterministic UC w/o Storage



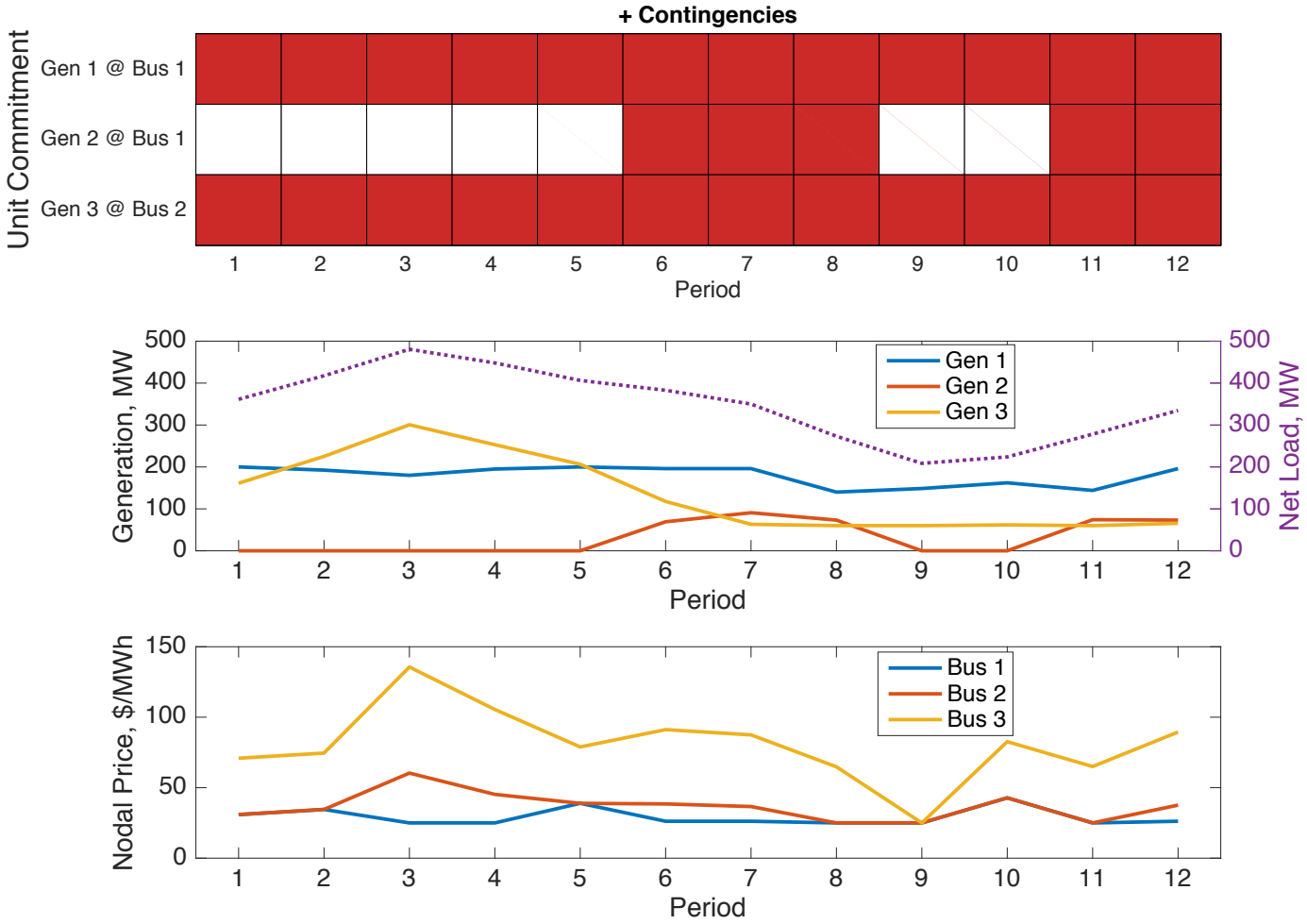
Stochastic UC Examples



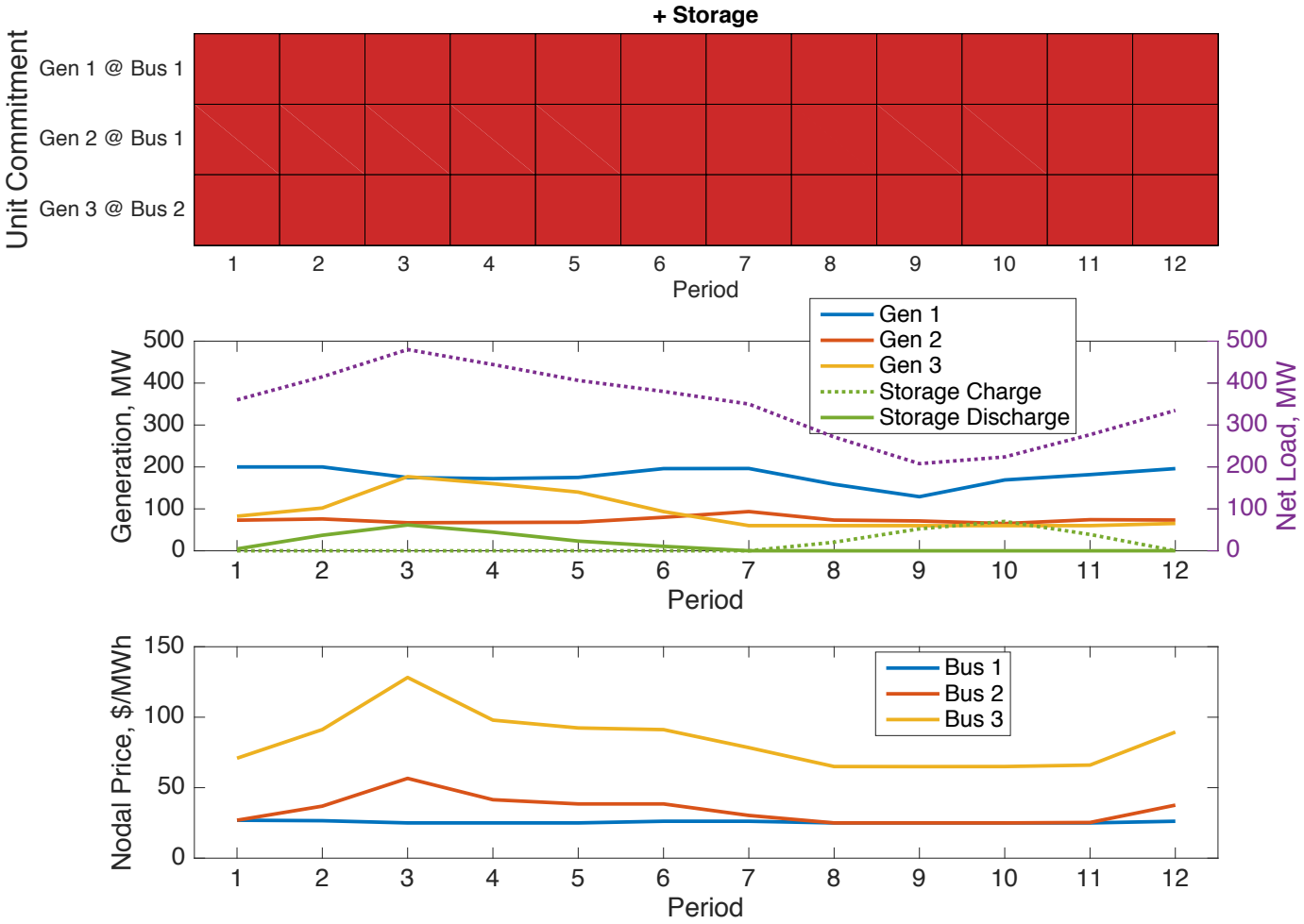
Stochastic UC Examples



Stochastic UC Examples



Stochastic UC Examples



Storage Data

```
function storage = ex_storage(mpc)

ecap = 200;             %% energy capacity
pcap = 80;              %% power capacity
scost = 45.17;          %% cost/value of initial/residual stored energy

%   bus Pg  Qg  Qmax Qmin Vg  mBase status Pmax Pmin  ...
storage.gen = [
    3   0   0   0    0    1   100   1      pcap -pcap   ...
];

storage.xgd_table.colnames = {
'CommitKey', 'CommitSched', 'PositiveActiveReservePrice', ... };

storage.xgd_table.data = [
 2   1   0   2*pcap  0   2*pcap  0   0   0   2*pcap  0   2*pcap;
];
```

Storage Data (continued)

```
function storage = ex_storage(mpc)
...
storage.sd_table.OutEff      = 1;
storage.sd_table.InEff       = 1;
storage.sd_table.LossFactor  = 0;
storage.sd_table.rho         = 0;
storage.sd_table.colnames  = {
    'InitialStorageLowerBound', ...
        'InitialStorageUpperBound', ...
                'InitialStorageCost', ...
                        'TerminalStoragePrice', ...
                                'MinStorageLevel', ...
                                    'MaxStorageLevel' };
storage.sd_table.data  = [
    0   ecap    scost   scost   0   ecap;
];
```

Load Profile Input

```
function loadprofile = ex_load_profile

loadprofile = struct( ...
    'type', 'mpcData', ...
    'table', CT_TLOAD, ...
    'rows', 0, ...
    'col', CT_LOAD_ALL_PQ, ...
    'chgtype', CT_REP, ...
    'values', [] );

loadprofile.values(:, 1, 1) = [
 440;
 480;
 540
 525;
 500;
 450;
 400;
 350;
 300;
 325;
 375;
 425;
];
```

load data

for all loads

should be replaced

with specified values

number of periods in planning horizon

Wind Profile Input

```
function windprofile = ex_wind_profile

windprofile = struct( ...
    'type', 'mpcData', ...
    'table', CT_TGEN, ...
    'rows', 1, ...
    'col', PMAX, ...
    'chgtype', CT_REL, ...
    'values', [] );

windprofile.values(:, :, 1) = [
 0.72   0.80   0.88;
 0.49   0.65   0.81;
 0.36   0.60   0.84;
 0.50   0.82   1.14;
 0.60   1.00   1.40;
 0.22   0.70   1.18;
 0.00   0.50   1.00;
 0.33   0.85   1.37;
 0.46   1.00   1.54;
 0.54   1.10   1.66;
 0.48   1.06   1.64;
 0.35   0.95   1.55;
];
```

generator parameter PMAX (max generation) of wind generator 1 should be scaled by specified values

number of periods in planning horizon

number of wind scenarios

Transition Probabilities

```
function transmat = ex_transmat(nt)

transmat = cell(1, nt);
transmat{1} = [ 0.16;
                0.68;
                0.16 ];
transmat{2}  = [ 0.16  0.16  0.16;
                 0.68  0.68  0.68;
                 0.16  0.16  0.16 ];
 .
 .
 .
transmat{nt} = [ 0.16  0.16  0.16;
                 0.68  0.68  0.68;
                 0.16  0.16  0.16 ];
```

scenarios in period 2

scenarios in period 1

Example Code

```
mpc = loadcase('ex_case3b');
xgd = loadxgendata('ex_xgd_uc', mpc);
[iwind, mpc, xgd] = addwind('ex_wind_uc', mpc, xgd);
[iess, mpc, xgd, sd] = addstorage('ex_storage', mpc, xgd);
profiles = getprofiles('ex_wind_profile', iwind);
profiles = getprofiles('ex_load_profile', profiles);
nt = size(profiles(1).values, 1);
transmat = ex_transmat(nt);

mdi = loadmd(mpc, transmat, xgd, sd, 'ex_contab', profiles);

mpopt = mpoption(mpopt, 'most.storage.cyclic', 1);

mdo = most(mdi, mpopt);

ms = most_summary(mdo);
plot_uc(mdo)
plot_storage(mdo)
```

load individual input data pieces

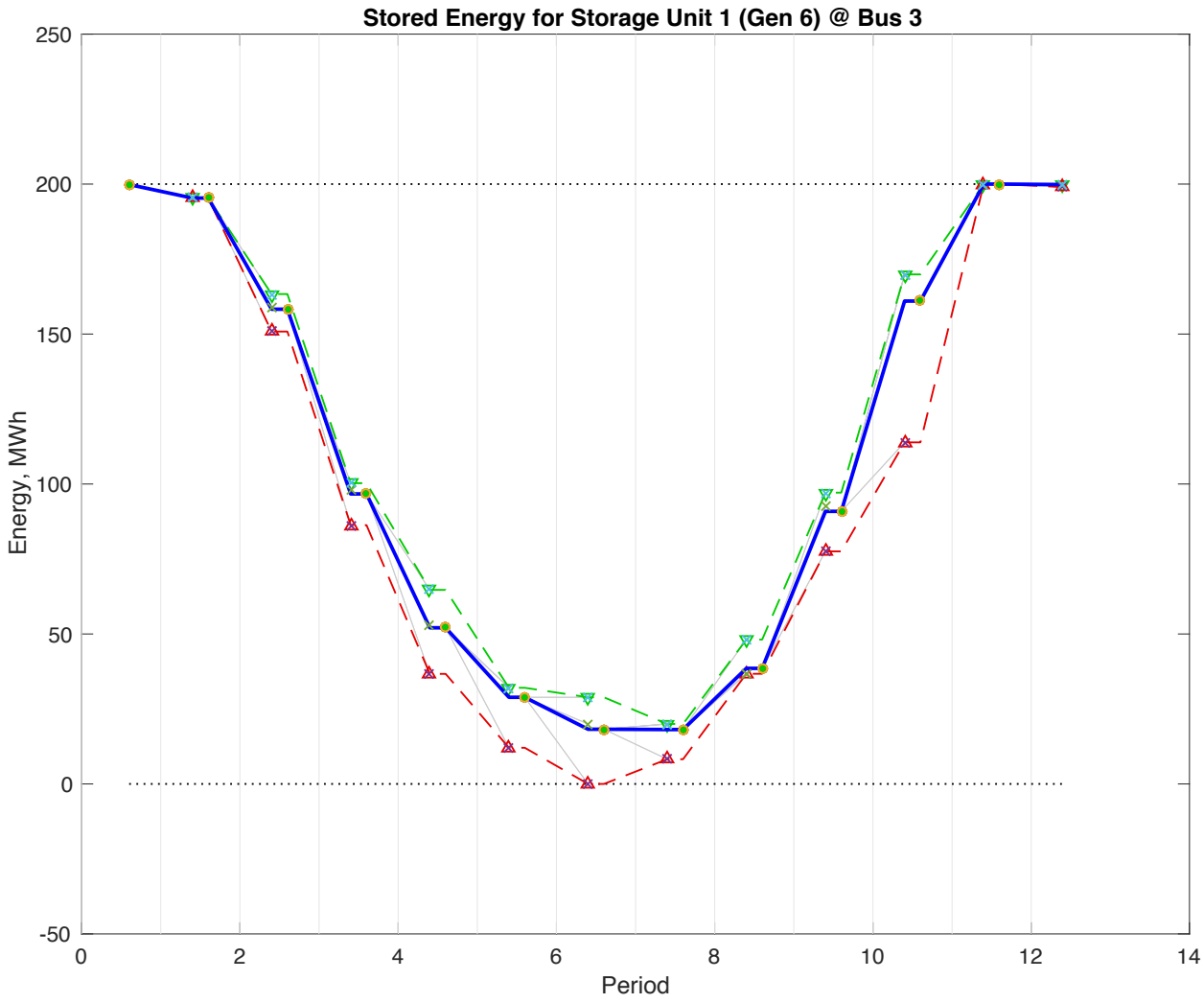
assemble everything

set options

run MOST optimization

examine results

Storage Profile Solution



MOST v1.0b1

- MOST v1.0b1 is available as part of MATPOWER 6.0b1 from the MATPOWER website:
  - http://www.pserc.cornell.edu/matpower/
- Full description of MOST problem formulation, software reference, and tutorial examples found in:
  - R.D. Zimmerman and C.E. Murillo-Sánchez, ***M****ATPOWER* ***O****ptimal* ***S****cheduling* ***T****ool (MOST) User's Manual*, 2016.

http://www.pserc.cornell.edu/matpower/MOST-manual.pdf

Questions?