ActuatorSelection_Test3_2

October 27, 2021

1 Benefit of Multiplicative (MPL) Models over Nominal (Nom) Models of Systems

Testing actuator selection and feedback of Nominal and MPL models on simulations of True system

Py Packages

[1]:
```
import numpy as np
from copy import deepcopy as dc
# %matplotlib widget

from functionfile_system_definition import sys_from_file, system_display_matrix
from functionfile_system_mplcost import simulation_nom_vs_mpl,␣
↪plot_simulation_nom_vs_mpl_1, plot_simulation_nom_vs_mpl_2,␣
↪actuator_comparison
```

1.1 Code

[2]:
```
test_set = 'System Model 3'
S_True = sys_from_file(test_set + ' True')
S_MPL = sys_from_file(test_set + ' MPL')
S_Nom = sys_from_file(test_set + ' Nominal')
```

```
System read from file @ system_model/System Model 3 True.pickle

System read from file @ system_model/System Model 3 MPL.pickle

System read from file @ system_model/System Model 3 Nominal.pickle
```

[3]:
```
ret_sim = simulation_nom_vs_mpl(S_Nom, S_MPL, S_True)
```

1.2 Output

System Models

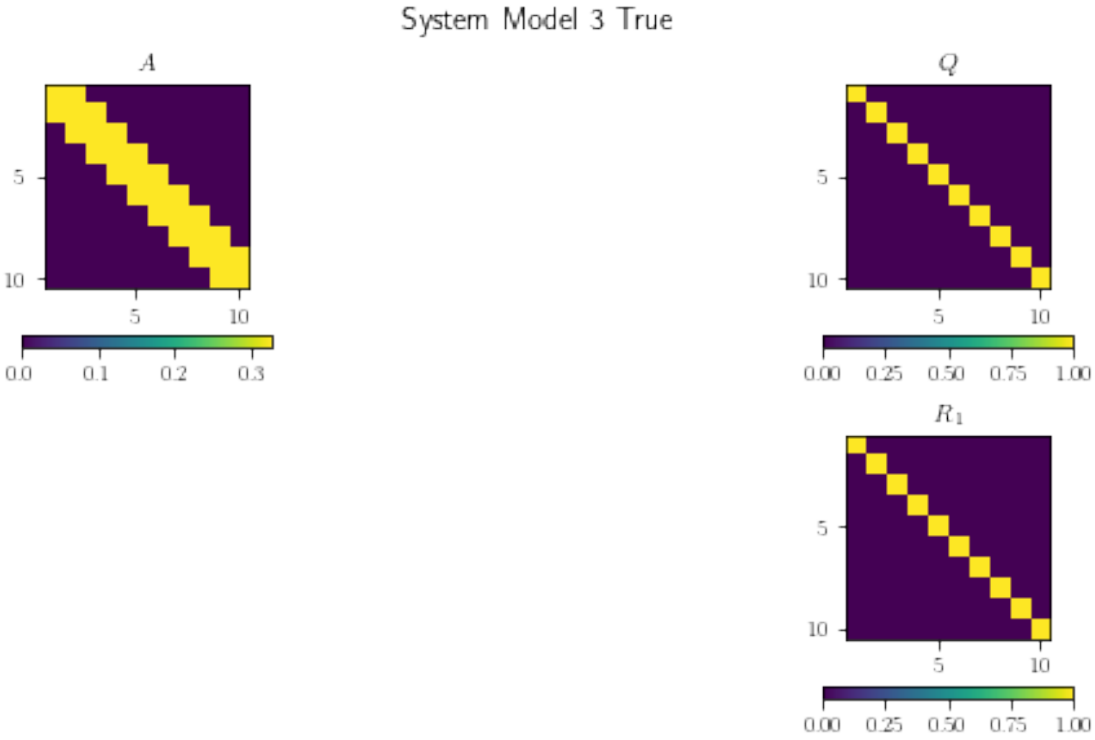
True System

[4]:
```

```

```
print('max(abs(eigvals(A)))= %.4f' % (np.max(np.abs(np.linalg.
 ↪eigvals(S_True['A'])))))
system_display_matrix(S_True)
```

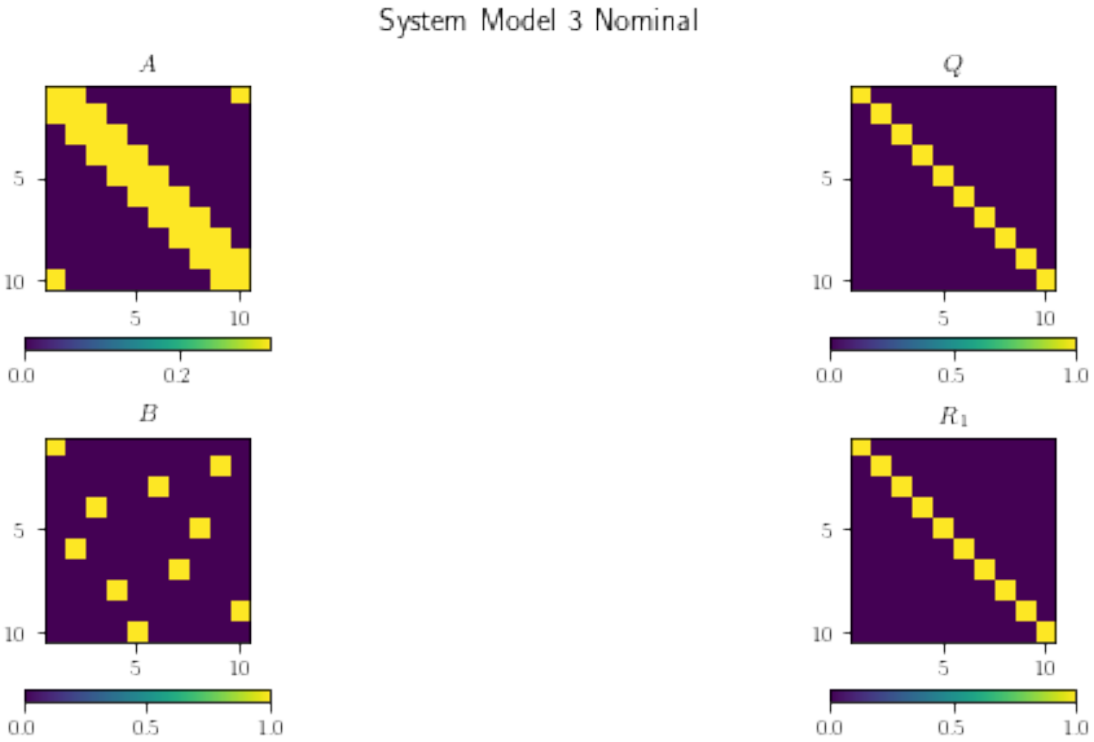
```
max(abs(eigvals(A)))= 0.9500
```



Nominal Model

[5]:
```
print('max(abs(eigvals(A)))= %.4f' % (np.max(np.abs(np.linalg.
 ↪eigvals(ret_sim['system_nom']['A'])))))
system_display_matrix(ret_sim['system_nom'])
```

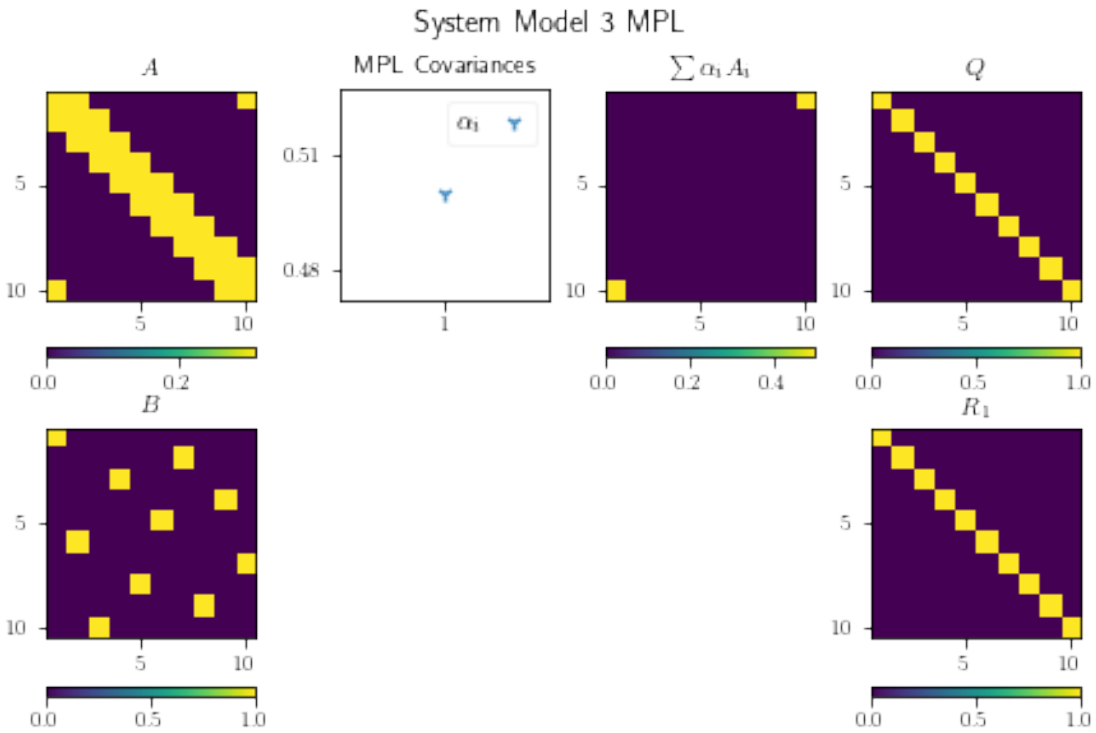
```
max(abs(eigvals(A)))= 0.9500
```



Multiplicative Noise Model

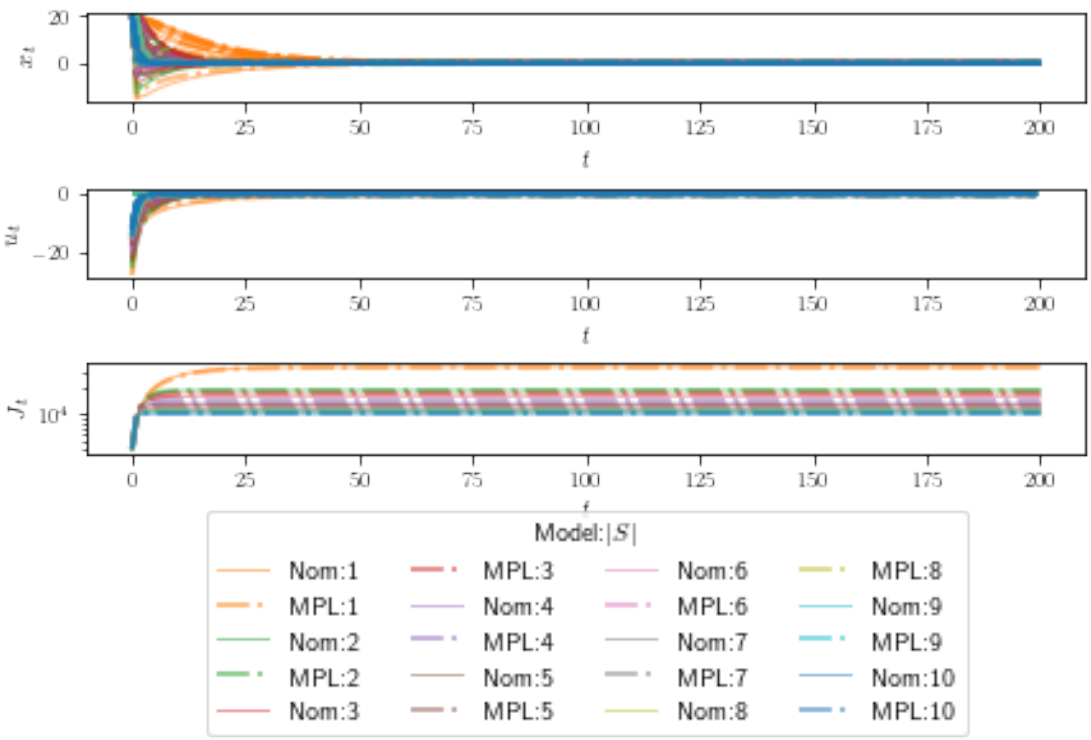
[6]:
```
print('max(abs(eigvals(A)))= %.4f' % (np.max(np.abs(np.linalg.
 ↪eigvals(ret_sim['system_mpl']['A'])))))
system_display_matrix(ret_sim['system_mpl'])
```

```
max(abs(eigvals(A)))= 0.9500
```



Simulation - Trajectory, Control Input and Costs

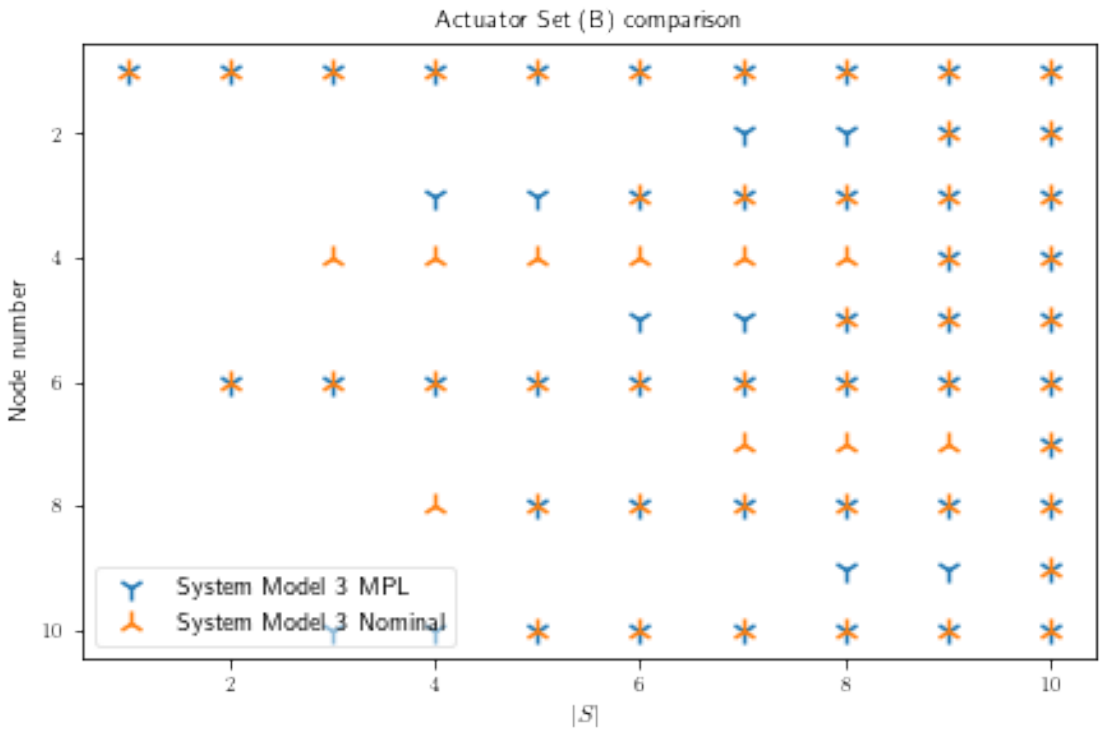
[7]:
```
plot_simulation_nom_vs_mpl_1(ret_sim)
```



Actuator set comparison

[8]:
```
actuator_comparison(ret_sim['system_mpl'], ret_sim['system_nom']);
```

```
Control sets are different
```



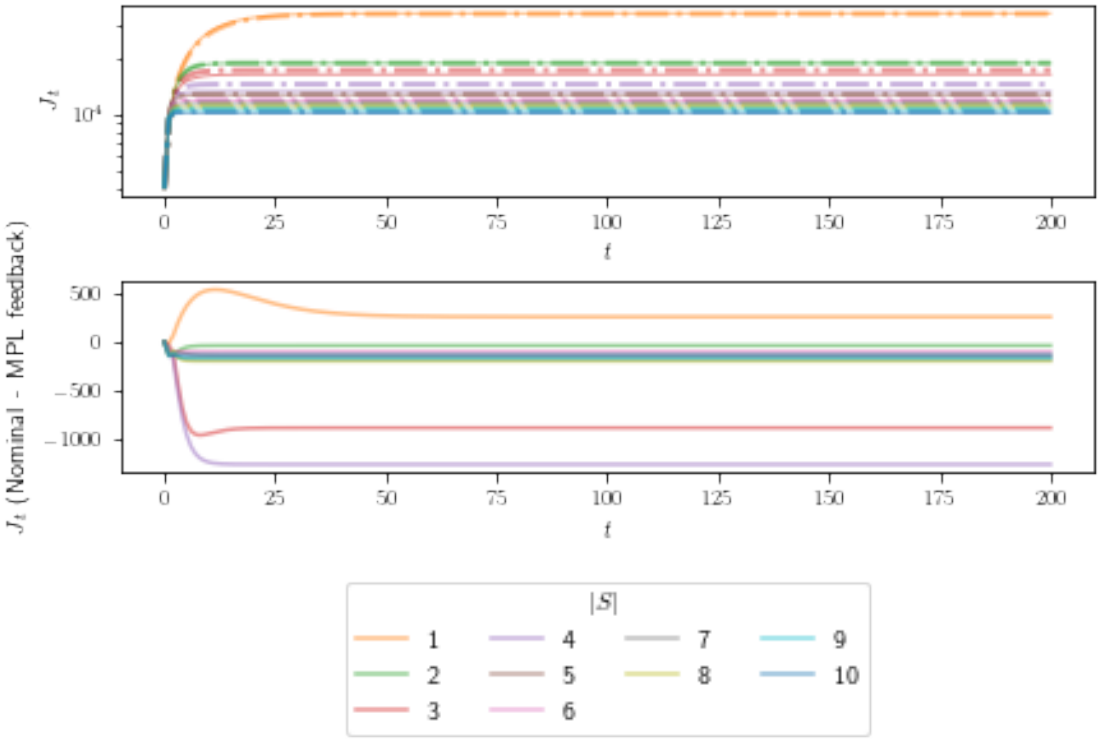
Simulated costs

[9]:
```
print('True simulation cost with <___> feedback (4decimal approx)')
for key in ret_sim['T_Nom']['costs']:
    print ("|S|: %s | Nom: %.4f | MPL: %.4f | Diff (Nom-MPL) %.4f (%.4f %% of␣
 ↪Nom)" % (key, ret_sim['T_Nom']['costs'][key][-1],␣
 ↪ret_sim['T_MPL']['costs'][key][-1],␣
 ↪ret_sim['T_Nom']['costs'][key][-1]-ret_sim['T_MPL']['costs'][key][-1],␣
 ↪(ret_sim['T_Nom']['costs'][key][-1]-ret_sim['T_MPL']['costs'][key][-1])*100/
 ↪ret_sim['T_Nom']['costs'][key][-1]))
```

```
True simulation cost with <___> feedback (4decimal approx)
|S|: 1 | Nom: 34815.6853 | MPL: 34556.2774 | Diff (Nom-MPL) 259.4079 (0.7451 %
of Nom)
|S|: 2 | Nom: 18776.4275 | MPL: 18812.2290 | Diff (Nom-MPL) -35.8015 (-0.1907 %
of Nom)
|S|: 3 | Nom: 16329.9215 | MPL: 17211.8090 | Diff (Nom-MPL) -881.8874 (-5.4004 %
of Nom)
|S|: 4 | Nom: 13211.8646 | MPL: 14469.4590 | Diff (Nom-MPL) -1257.5944 (-9.5187
% of Nom)
|S|: 5 | Nom: 12676.1704 | MPL: 12811.6062 | Diff (Nom-MPL) -135.4358 (-1.0684 %
of Nom)
|S|: 6 | Nom: 11825.3233 | MPL: 11922.5220 | Diff (Nom-MPL) -97.1987 (-0.8220 %
of Nom)
```

```
|S|: 7 | Nom: 11341.9881 | MPL: 11518.6139 | Diff (Nom-MPL) -176.6259 (-1.5573 %
of Nom)
|S|: 8 | Nom: 10917.0678 | MPL: 11111.9140 | Diff (Nom-MPL) -194.8462 (-1.7848 %
of Nom)
|S|: 9 | Nom: 10518.8577 | MPL: 10685.4212 | Diff (Nom-MPL) -166.5634 (-1.5835 %
of Nom)
|S|: 10 | Nom: 10119.9392 | MPL: 10253.0080 | Diff (Nom-MPL) -133.0687 (-1.3149
% of Nom)
```

[10]:
```
plot_simulation_nom_vs_mpl_2(ret_sim)
```



1.3 Run Complete

[11]:
```
print('Run Complete')
```

```
Run Complete
```