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 2'
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 2 True.pickle

System read from file @ system_model/System Model 2 MPL.pickle

System read from file @ system_model/System Model 2 Nominal.pickle
```

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

1.2 Output

System Models

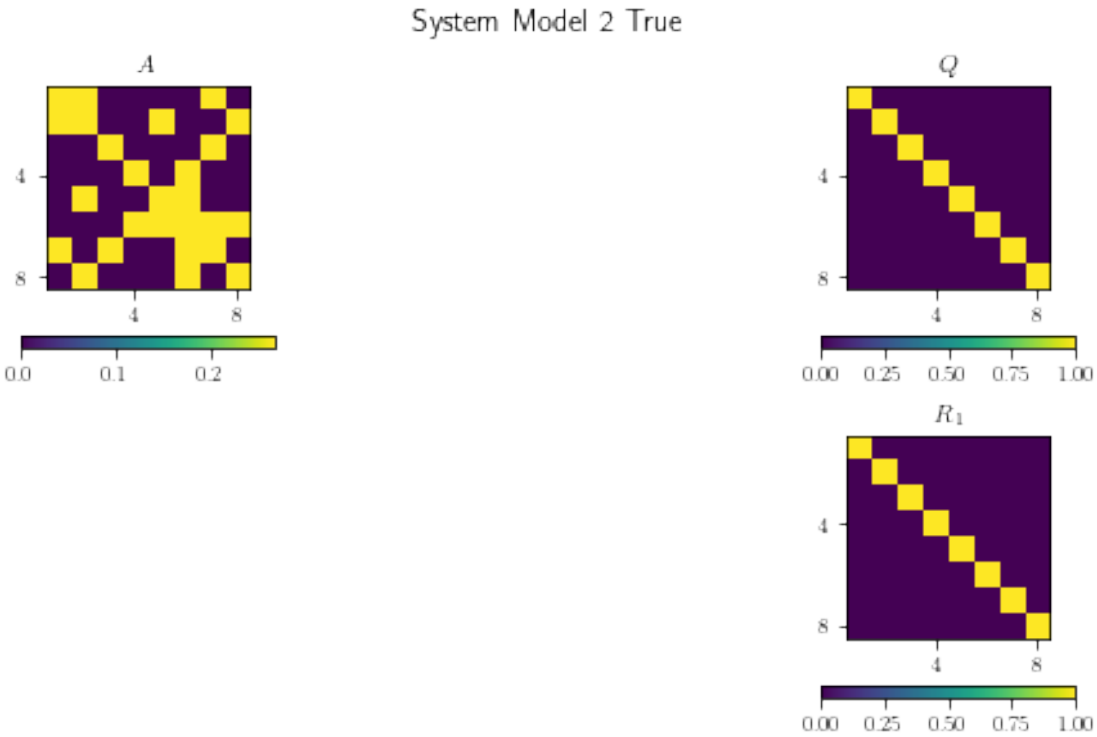
True System

[4]:
```

```

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

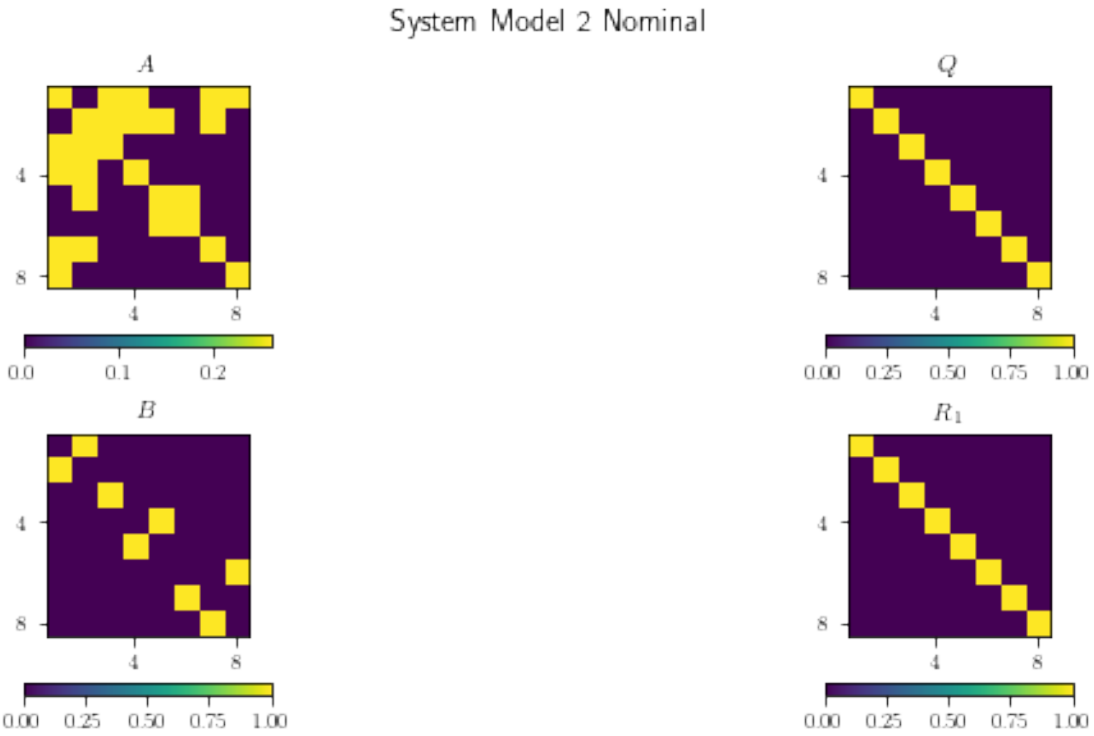
```
max(abs(eigvals(A)))= 0.9500
```



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'])
```

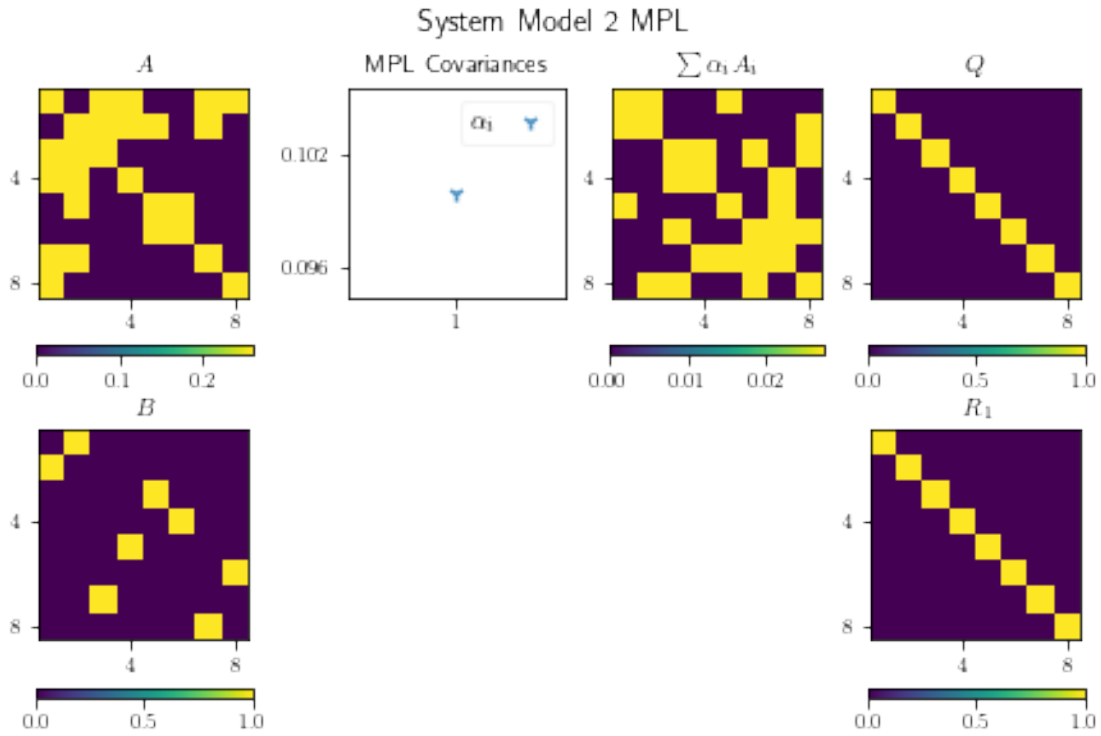
```
max(abs(eigvals(A)))= 0.9500
```



Multiplicative Noise Model

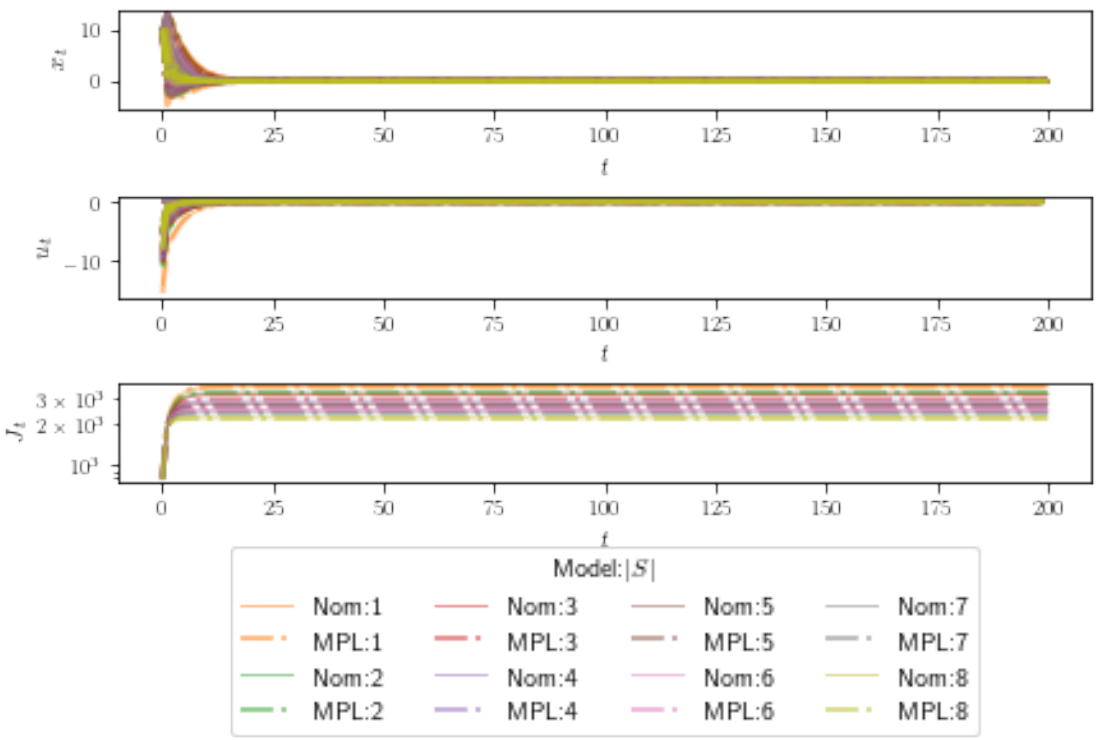
[6]:
```
print('max(abs(eigvals(A)))= %.4f' % (np.max(np.abs(np.linalg.
 ↪eigvals(ret_sim['system_mpl']['A'])))))
system_display_matrix(ret_sim['system_mpl'])
```

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



Simulation - Trajectory, Control Input and Costs

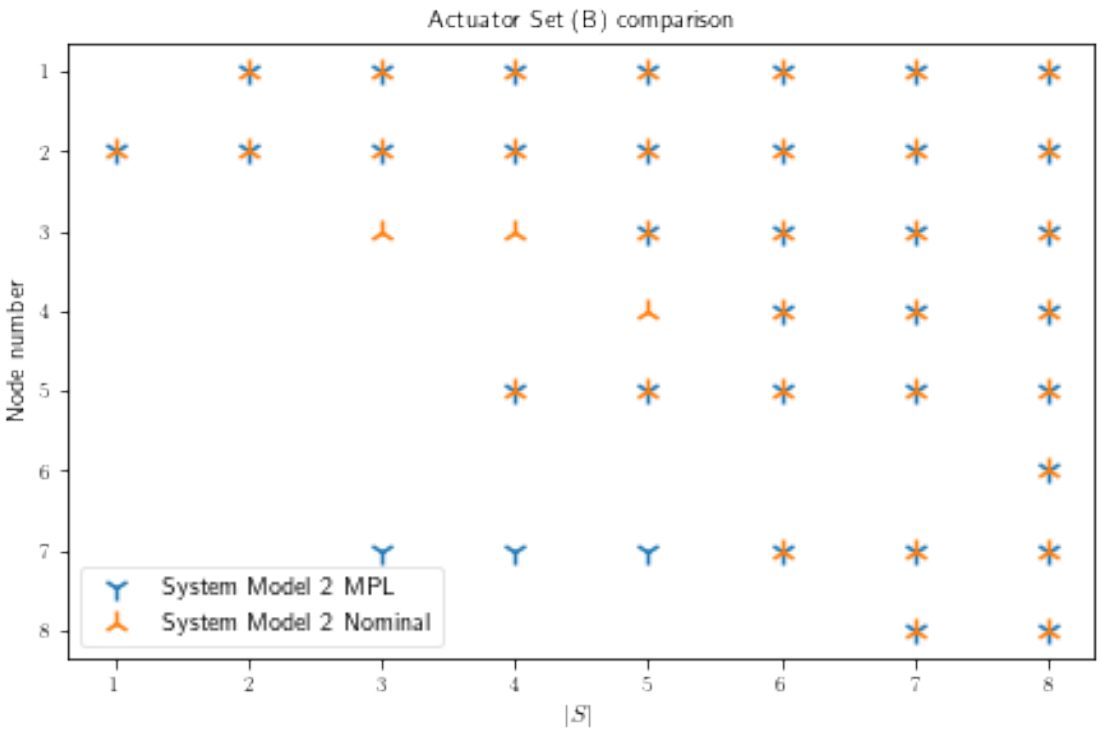
[7]:
```
plot_simulation_nom_vs_mpl_1(ret_sim)
```



Actuator set comparison

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

```
Control sets are different
```



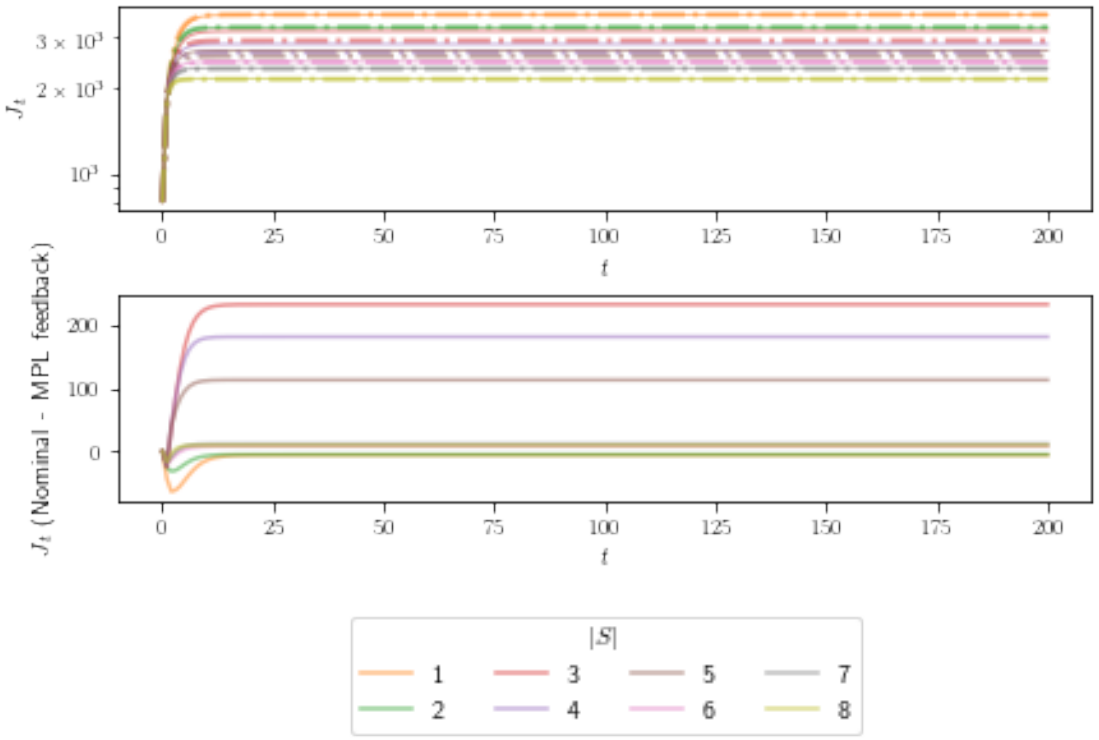
Simulated costs

[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: 3606.9008 | MPL: 3612.4029 | Diff (Nom-MPL) -5.5022 (-0.1525 % of
Nom)
|S|: 2 | Nom: 3256.9132 | MPL: 3261.6416 | Diff (Nom-MPL) -4.7284 (-0.1452 % of
Nom)
|S|: 3 | Nom: 3159.8338 | MPL: 2926.6825 | Diff (Nom-MPL) 233.1513 (7.3786 % of
Nom)
|S|: 4 | Nom: 2845.9666 | MPL: 2664.0970 | Diff (Nom-MPL) 181.8696 (6.3904 % of
Nom)
|S|: 5 | Nom: 2702.6602 | MPL: 2589.1462 | Diff (Nom-MPL) 113.5139 (4.2001 % of
Nom)
|S|: 6 | Nom: 2471.7069 | MPL: 2463.4772 | Diff (Nom-MPL) 8.2297 (0.3330 % of
Nom)
```

```
|S|: 7 | Nom: 2347.6342 | MPL: 2335.5871 | Diff (Nom-MPL) 12.0471 (0.5132 % of
Nom)
|S|: 8 | Nom: 2153.9735 | MPL: 2144.2142 | Diff (Nom-MPL) 9.7593 (0.4531 % of
Nom)
```

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



1.3 Run Complete

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

```
Run Complete
```