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

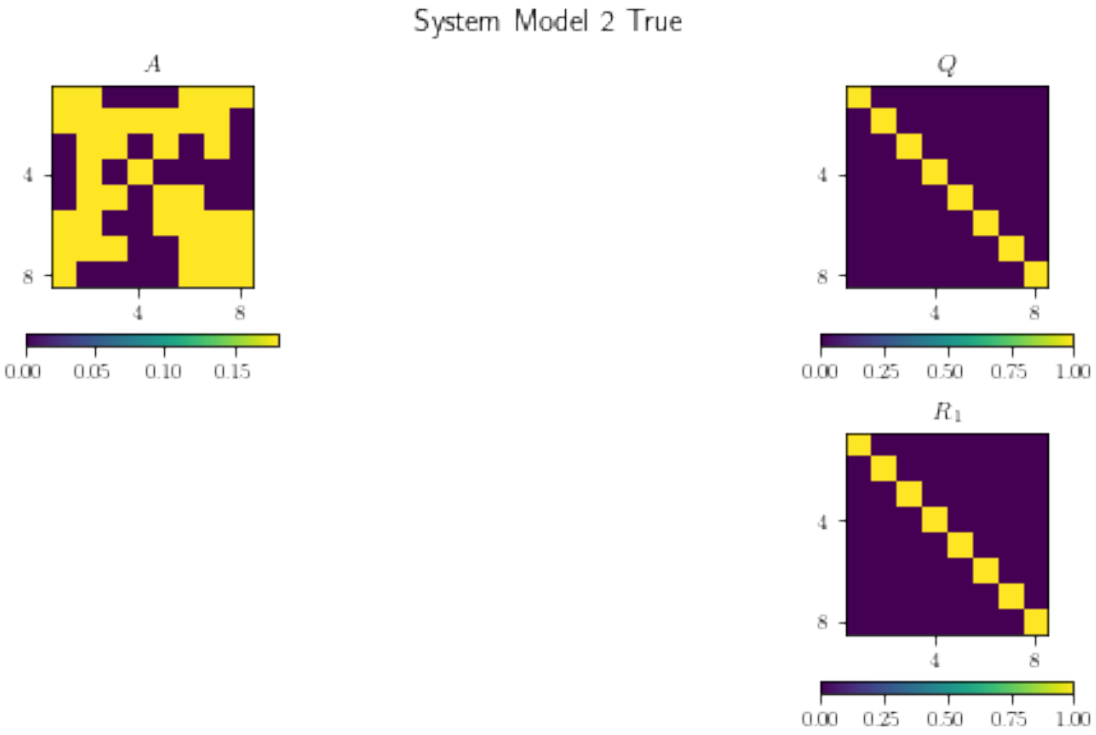
True System

[4]:
```

```

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

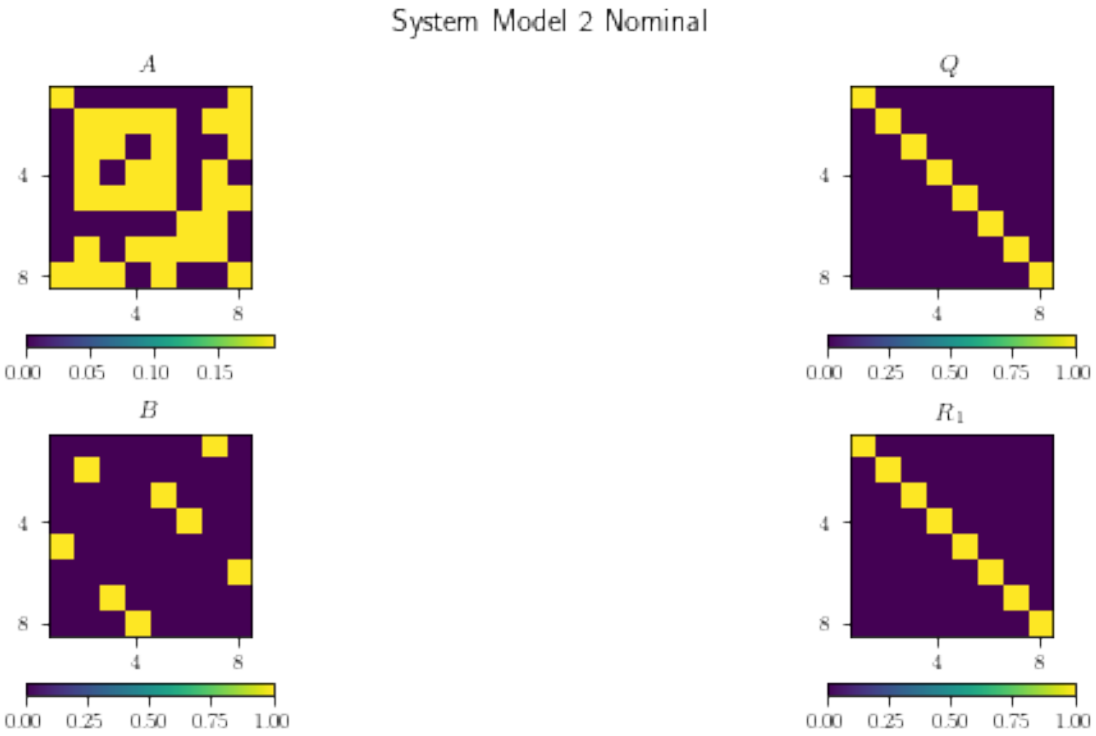
```
max(abs(eigvals(A)))= 0.9500
```



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'])
```

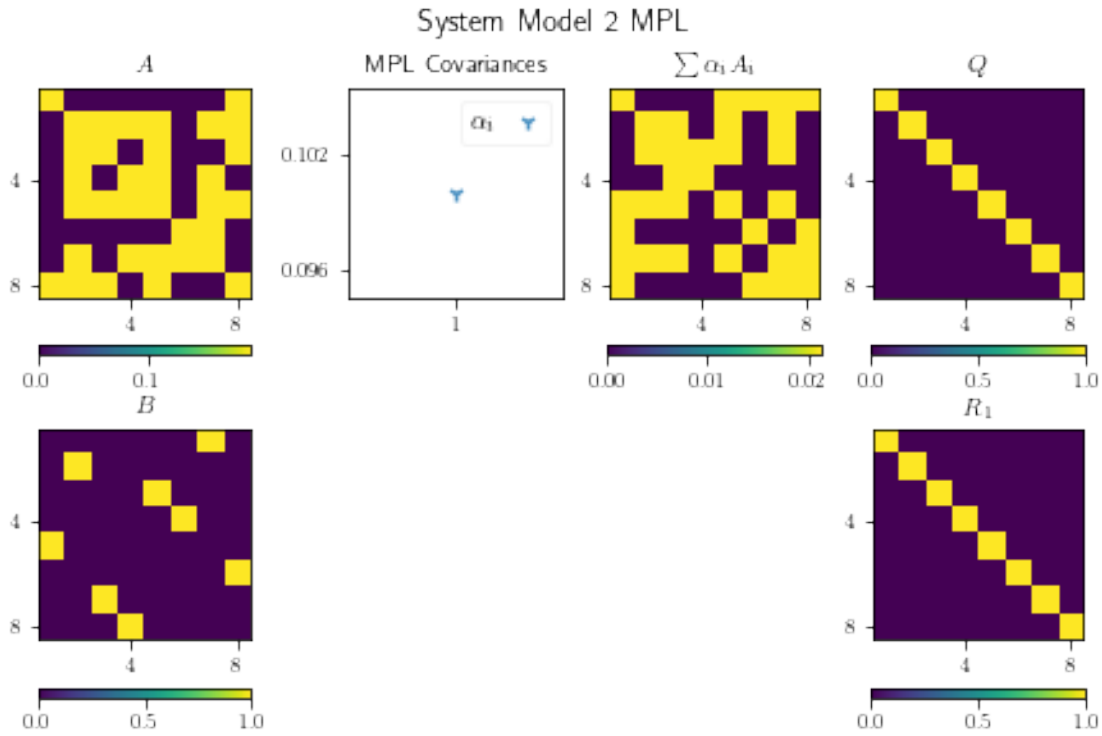
```
max(abs(eigvals(A)))= 0.9500
```



Multiplicative Noise Model

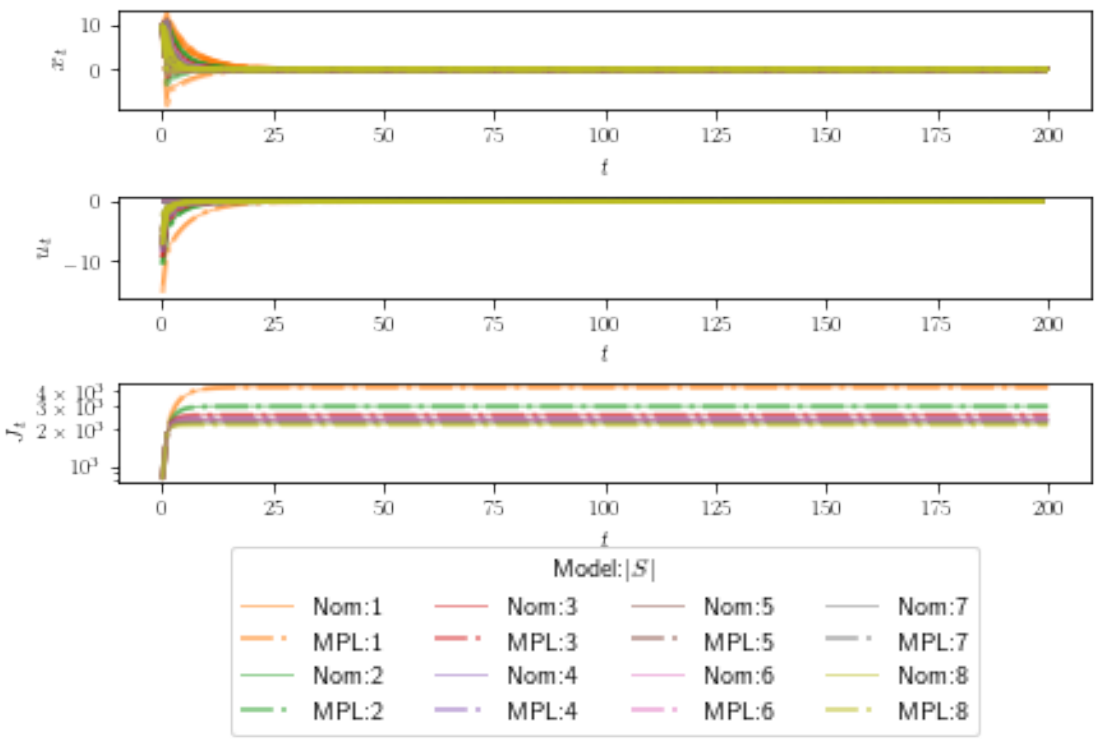
[6]:
```
print('max(abs(eigvals(A)))= %.4f' % (np.max(np.abs(np.linalg.
 ↪eigvals(ret_sim['system_mpl']['A'])))))
system_display_matrix(ret_sim['system_mpl'])
```

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



Simulation - Trajectory, Control Input and Costs

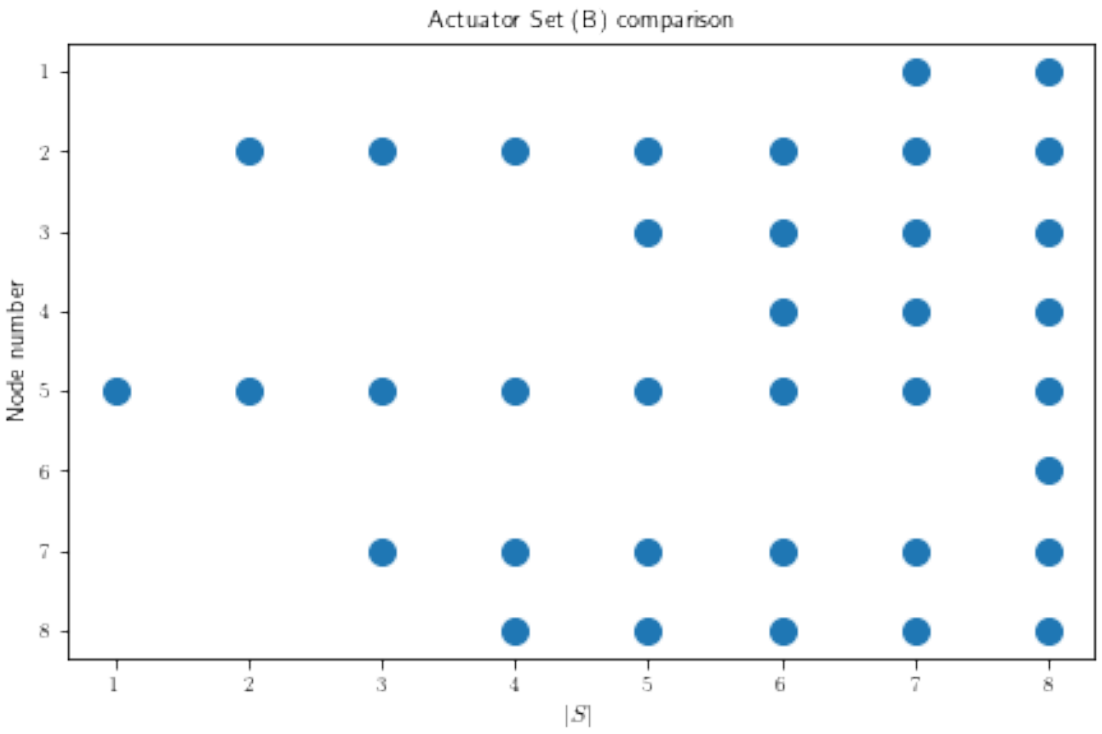
[7]:
```
plot_simulation_nom_vs_mpl_1(ret_sim)
```



Actuator set comparison

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

```
Both control sets are close/equal
```



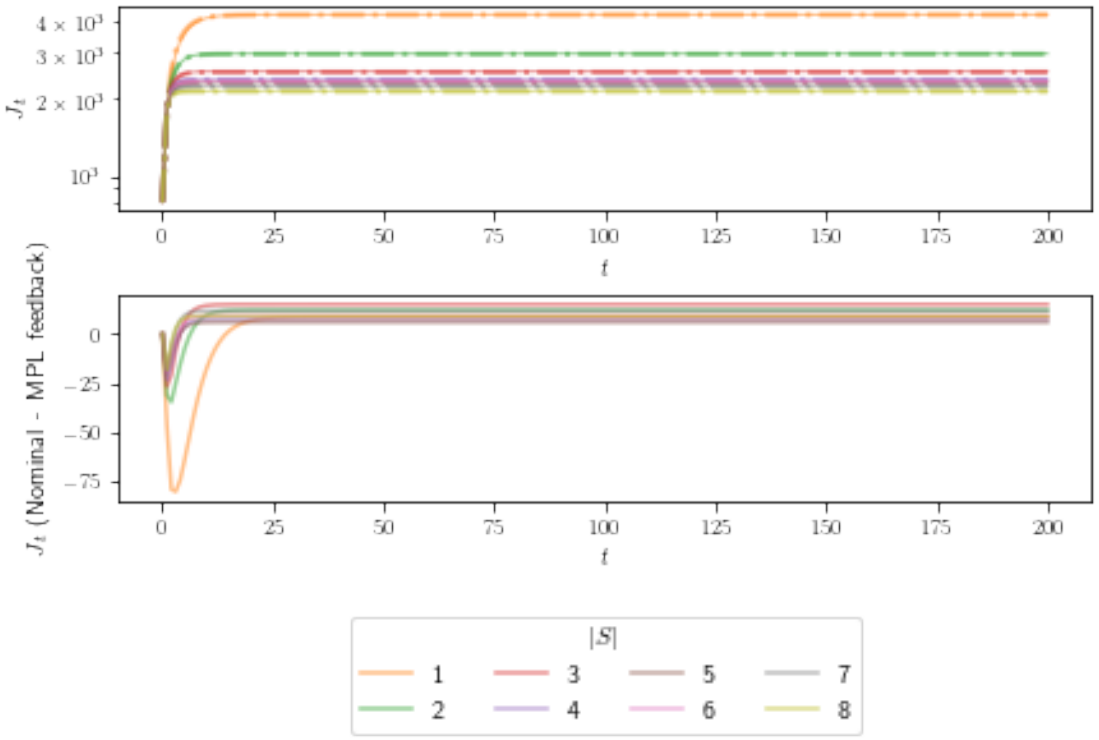
Simulated costs

[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: 4220.6831 | MPL: 4212.0645 | Diff (Nom-MPL) 8.6186 (0.2042 % of
Nom)
|S|: 2 | Nom: 2982.8254 | MPL: 2970.6105 | Diff (Nom-MPL) 12.2149 (0.4095 % of
Nom)
|S|: 3 | Nom: 2545.2506 | MPL: 2530.2153 | Diff (Nom-MPL) 15.0353 (0.5907 % of
Nom)
|S|: 4 | Nom: 2373.6717 | MPL: 2366.1708 | Diff (Nom-MPL) 7.5009 (0.3160 % of
Nom)
|S|: 5 | Nom: 2308.8786 | MPL: 2302.8577 | Diff (Nom-MPL) 6.0209 (0.2608 % of
Nom)
|S|: 6 | Nom: 2331.4944 | MPL: 2322.0881 | Diff (Nom-MPL) 9.4063 (0.4034 % of
Nom)
```

```
|S|: 7 | Nom: 2237.6855 | MPL: 2225.8219 | Diff (Nom-MPL) 11.8636 (0.5302 % of
Nom)
|S|: 8 | Nom: 2140.8183 | MPL: 2131.6711 | Diff (Nom-MPL) 9.1472 (0.4273 % of
Nom)
```

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



1.3 Run Complete

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

```
Run Complete
```