108-1 (Fall 2019)

自動控制實驗

Laboratory for Automatic Control

Week 02

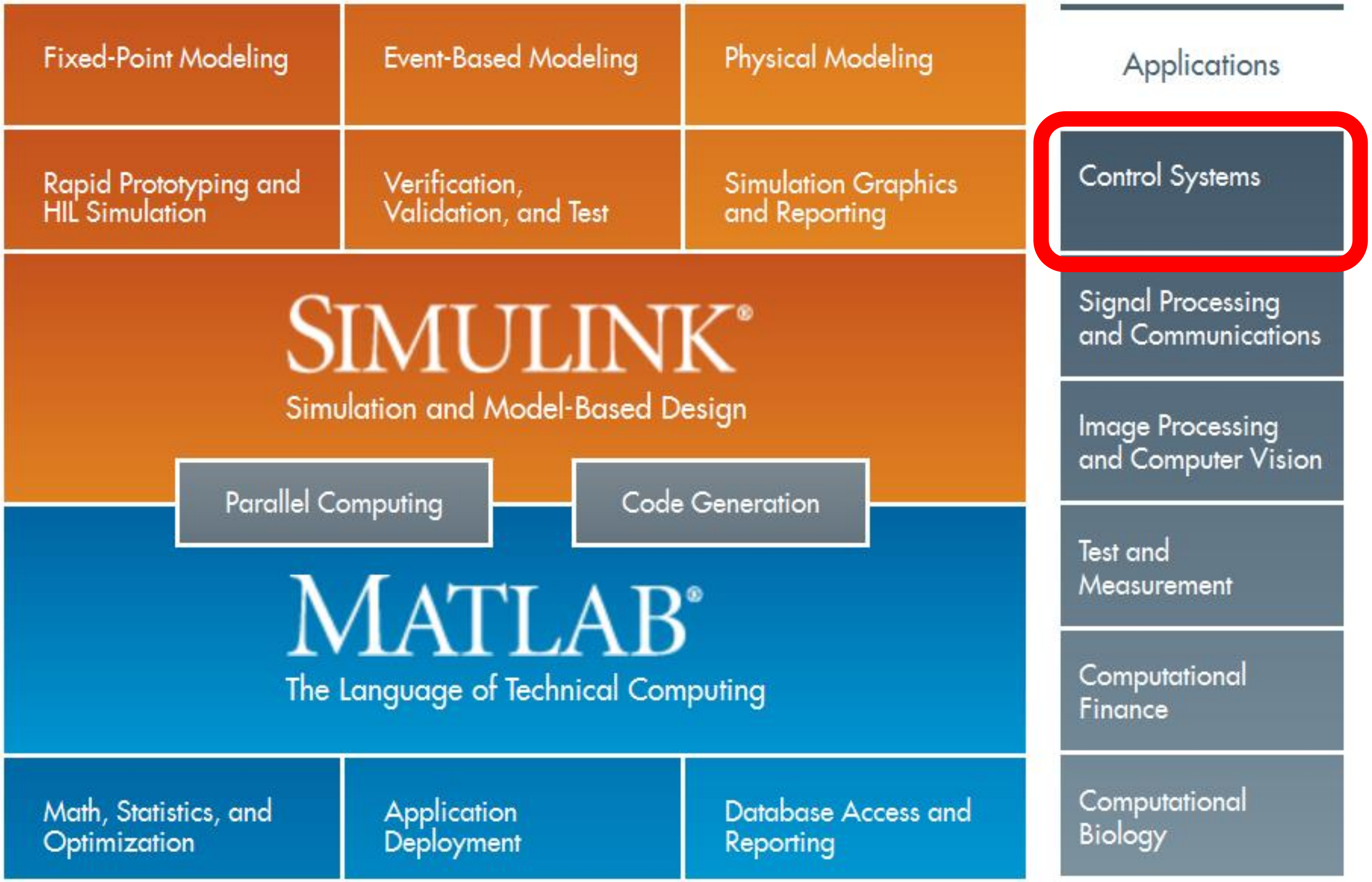
Control Toolbox

Prof. Feng-Li Lian

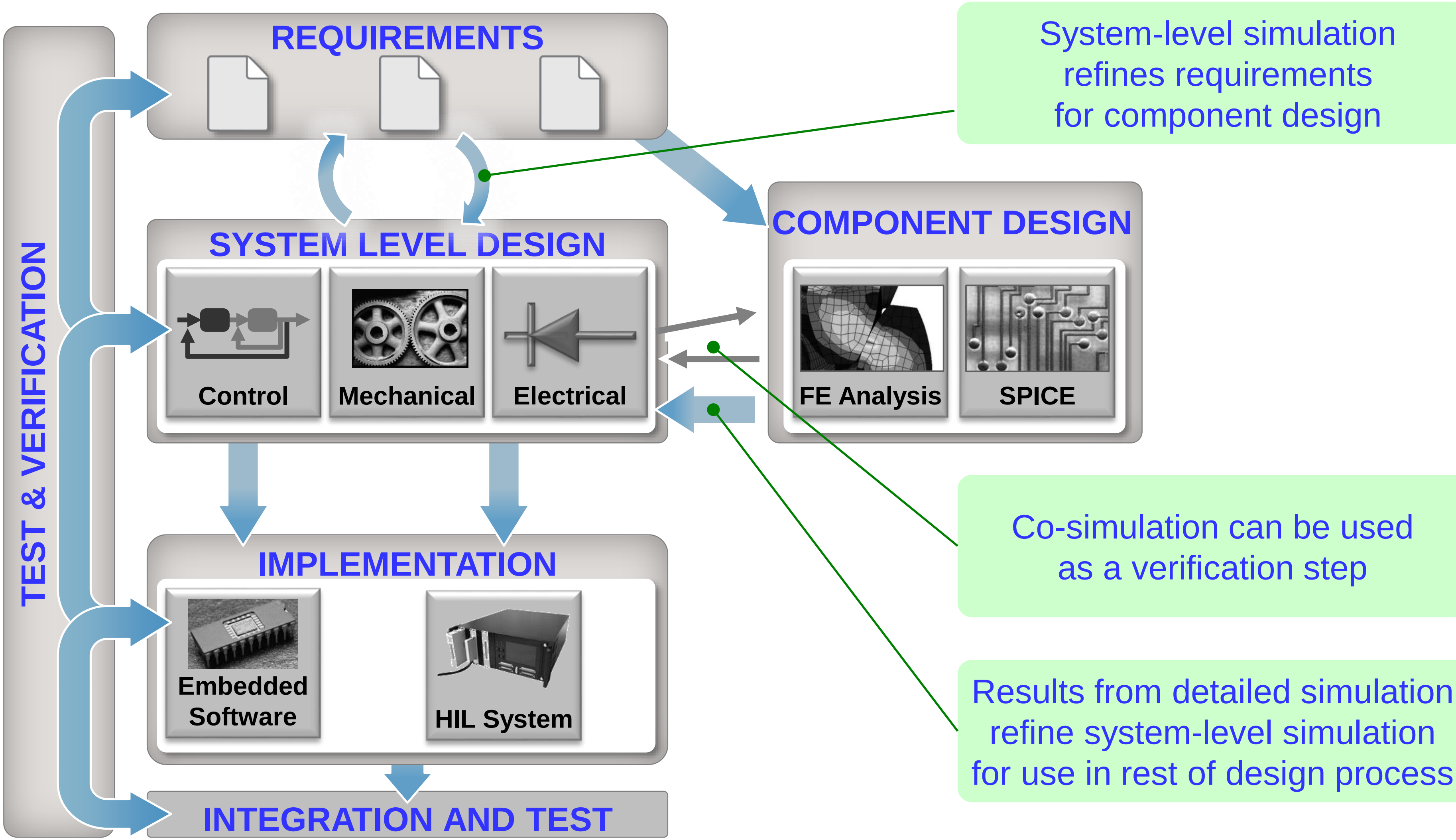
NTU-EE

Sep 2019 – Jan 2020

MathWorks Product Overview

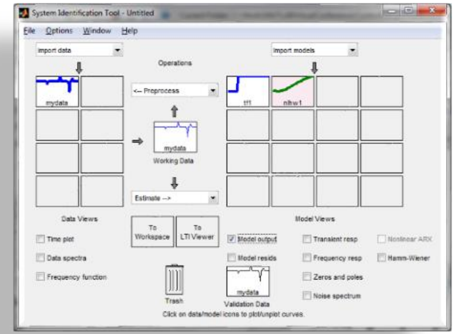


Model-Based Design Process: System and Component Level Design



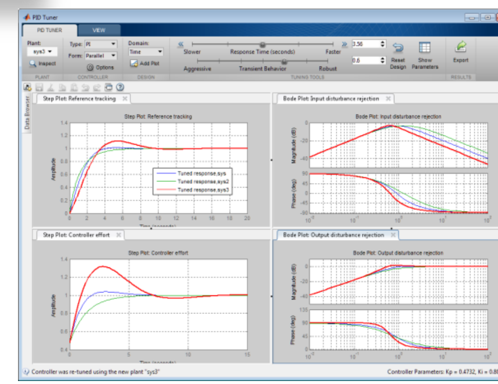
Control Toolboxes

System Identification Toolbox



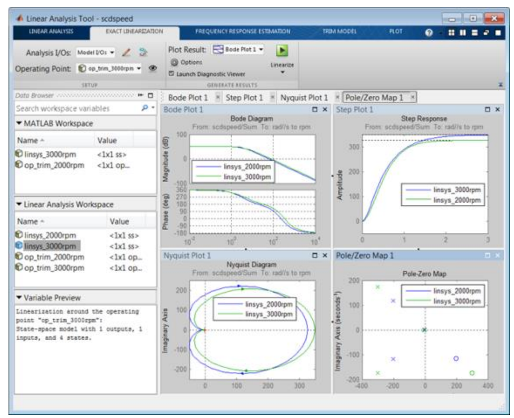
- To create data driven black-box or gray-box dynamic models

PID Tuner



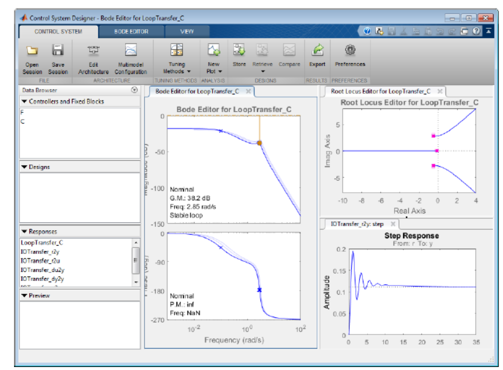
- Interactive tuning of PID controllers

Linear System Analyzer



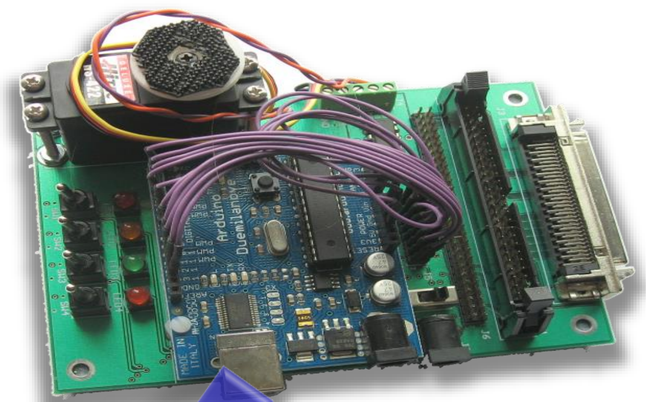
- Improved linear analysis workflow

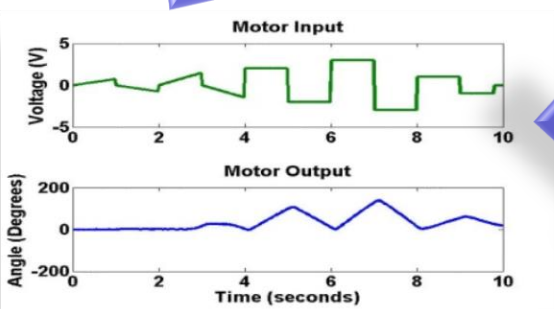
Control System Designer



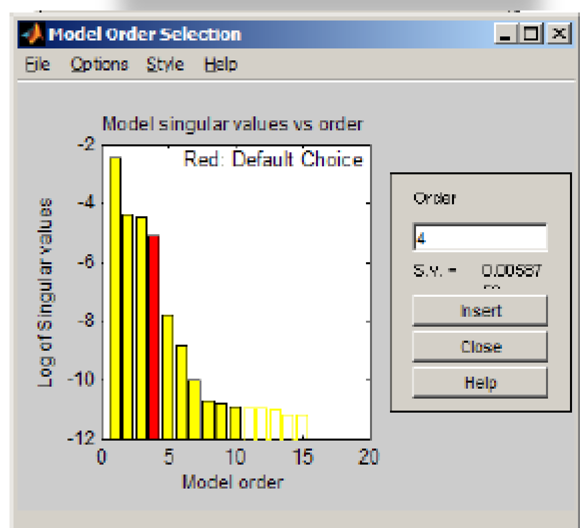
- Tune compensators using various graphical and automated tuning methods

▪ System Identification Toolbox





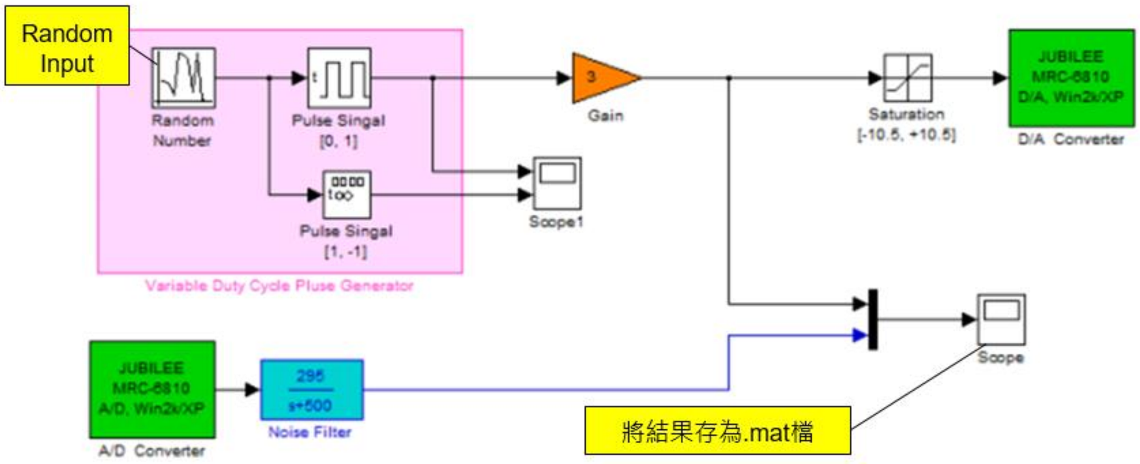
$$G(s) = \frac{1.4e9}{1 + 2.8e7s} e^{-0.1s}$$

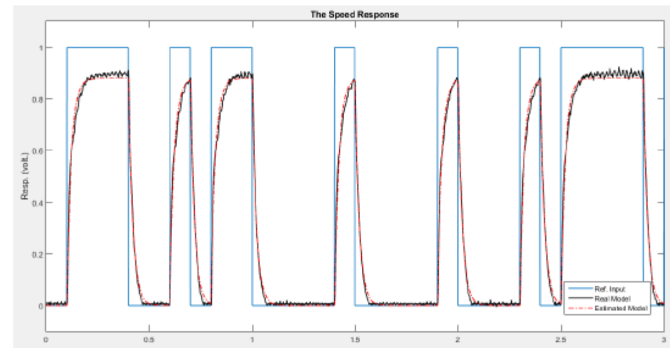


Use System Identification Toolbox

to create data driven black-box or gray-box dynamic models

Example: 105-1, G5Exp1

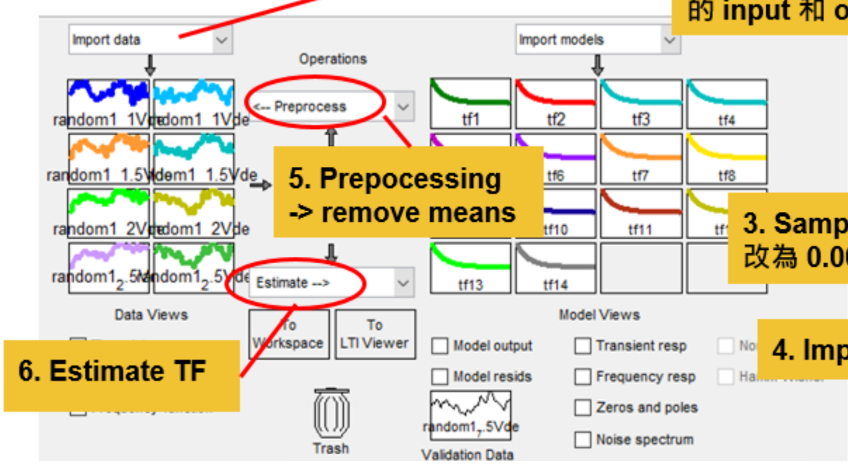


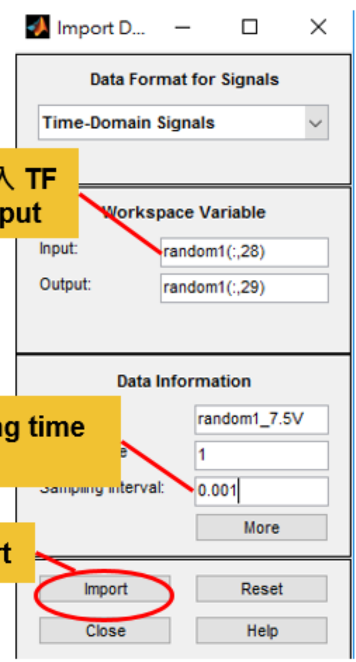


實驗原理

統計建模工具

MATLAB system Identfication Toolbox





統計建模:

$$TF_{7.5V} = \frac{3911}{s^2+95.71s+4727}$$

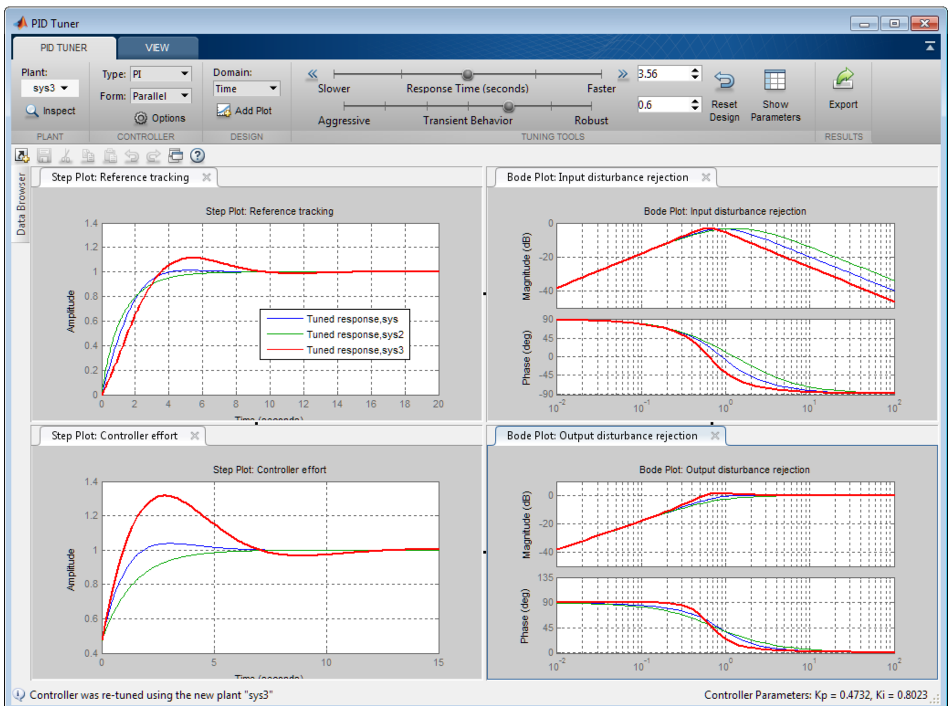
(Fit to estimation data: 92.5%)

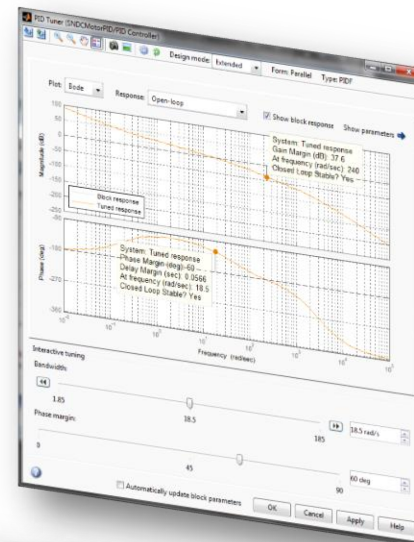
暫態響應:

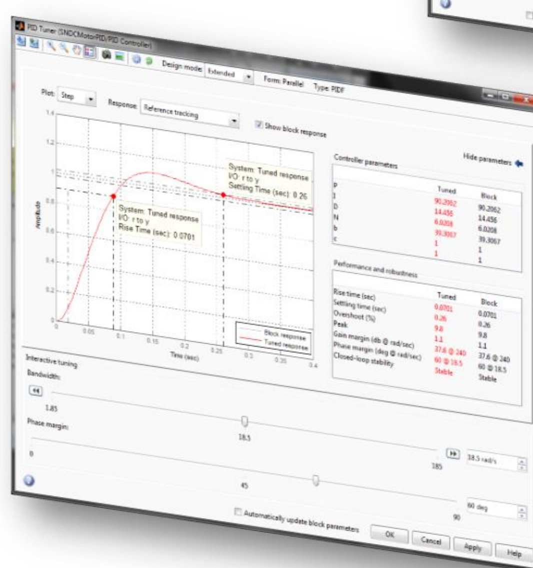
$$TF_{7.5V} = \frac{3042}{s^2+58.1s+3042}$$

Interactive tuning of PID controllers

- Automatically find the design that balances performance and robustness
- Let you easily try different controller structures
- Provides two sliders for fine-tuning the design
- Several response plots can be displayed simultaneously







Example: 105-1, G1Exp2

PIDTOOL: 統計模型

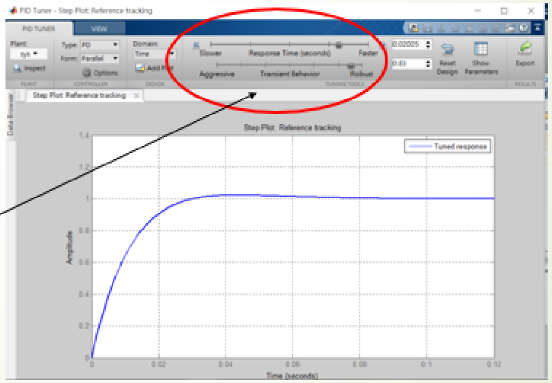
- **1. matlab code:**

```
sys = tf(217.4, [1 46.65 0])
pidtool(sys)
```

1.建立參數模型tf

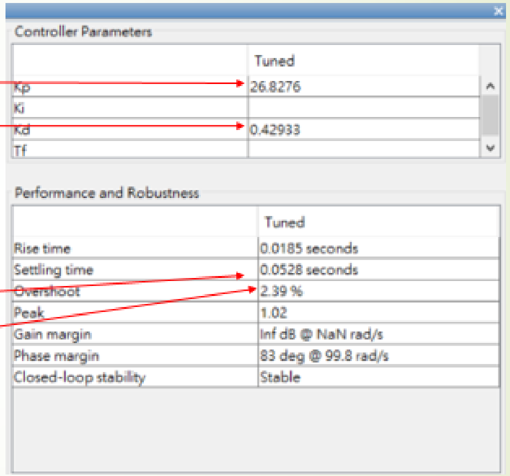
2.啟動PIDTOOL

- **2. 調整訊號的視窗:**



3.滑動調整模型規格

- $K_P = 26.8276$
- $K_D = 0.42933$
- Settling time = 0.0528s
- Overshoot = 2.39%

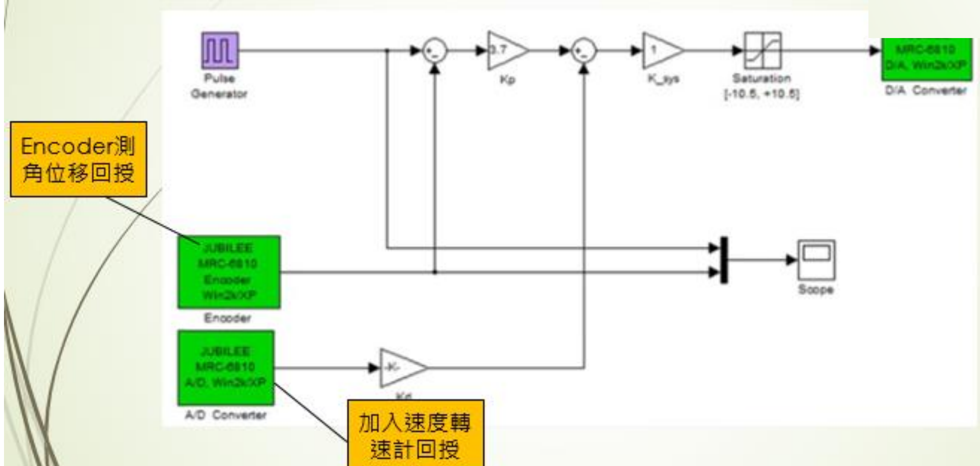


Controller Parameters

	Tuned
Kp	26.8276
Ki	
Kd	0.42933
Tf	

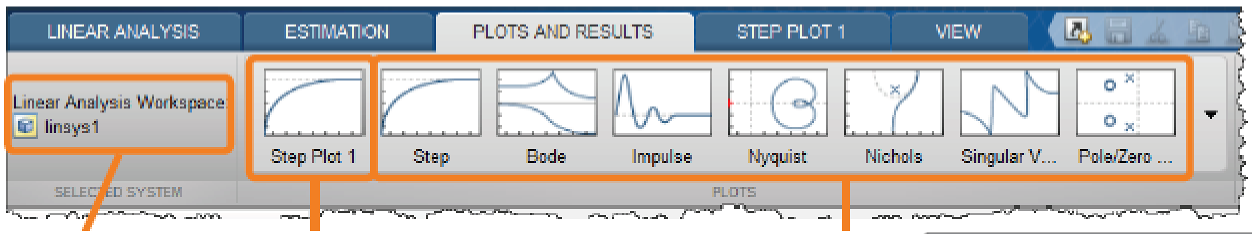
Performance and Robustness

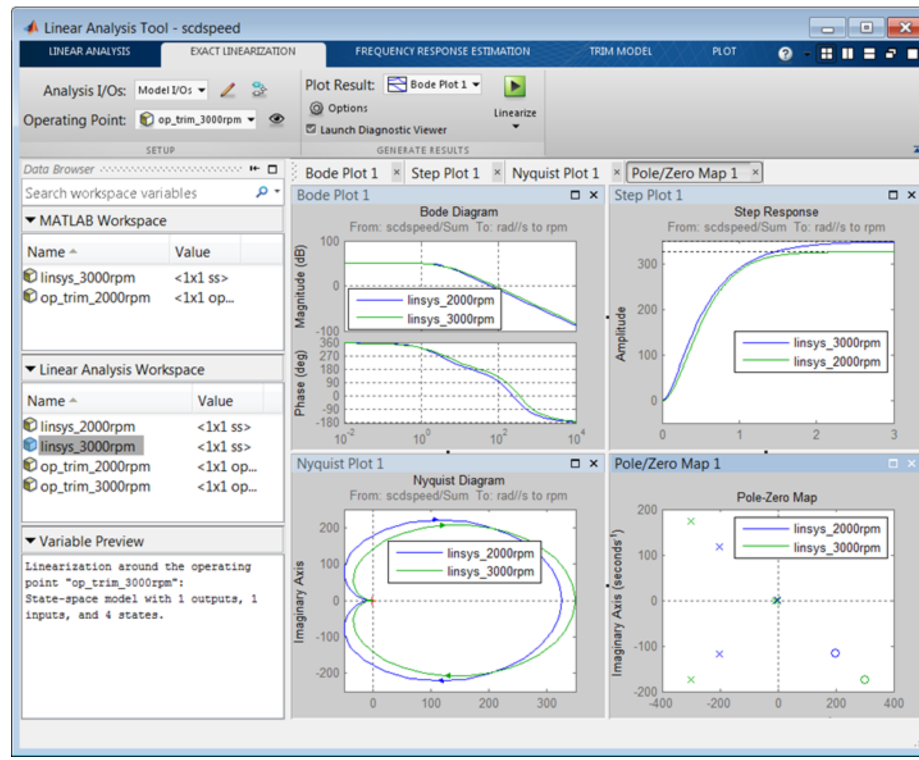
	Tuned
Rise time	0.0185 seconds
Settling time	0.0528 seconds
Overshoot	2.39 %
Peak	1.02
Gain margin	Inf dB @ NaN rad/s
Phase margin	83 deg @ 99.8 rad/s
Closed-loop stability	Stable



Improve linear analysis workflow

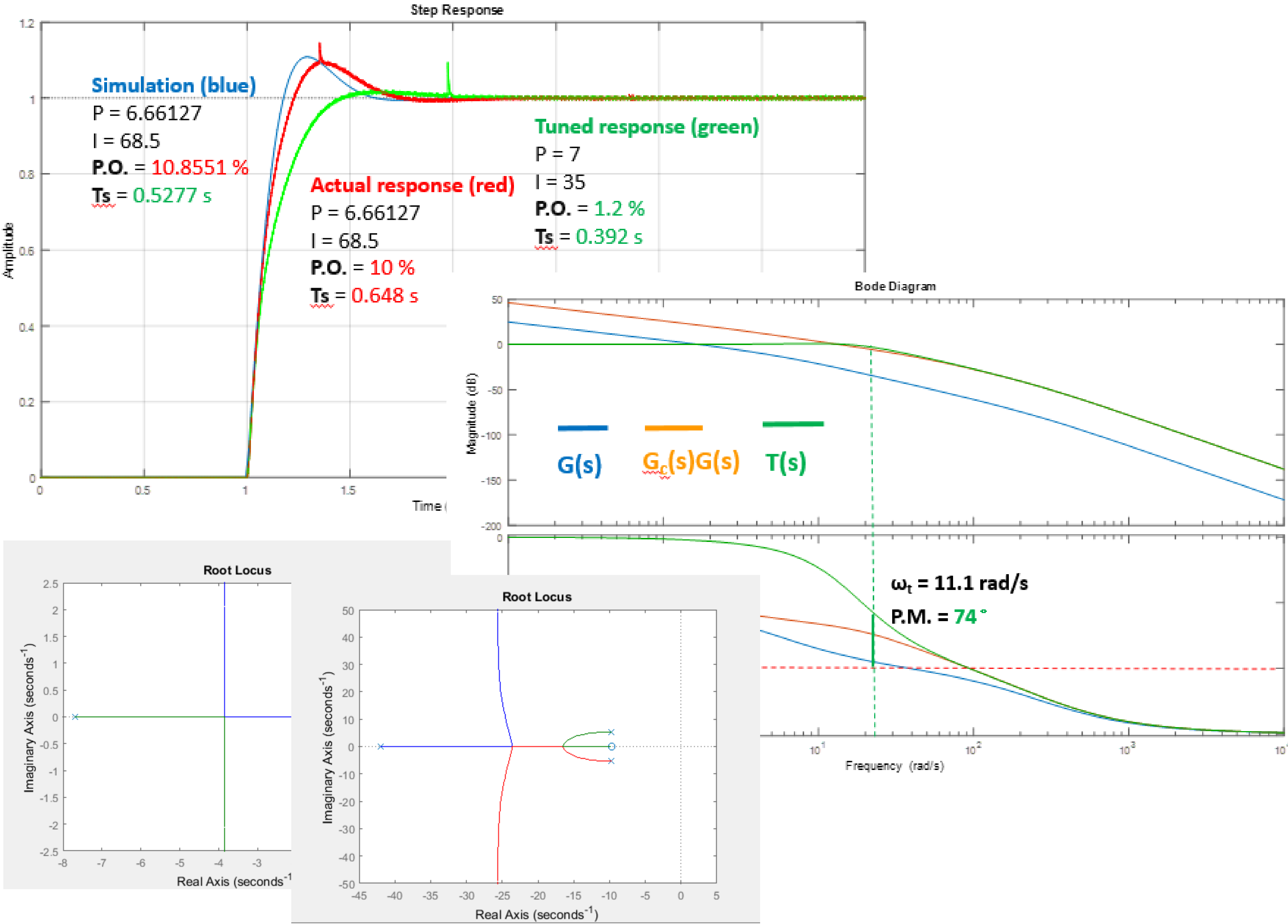
- Task-based, easy to use design
- Data browser to manage and manipulate results
- All plots available in one window



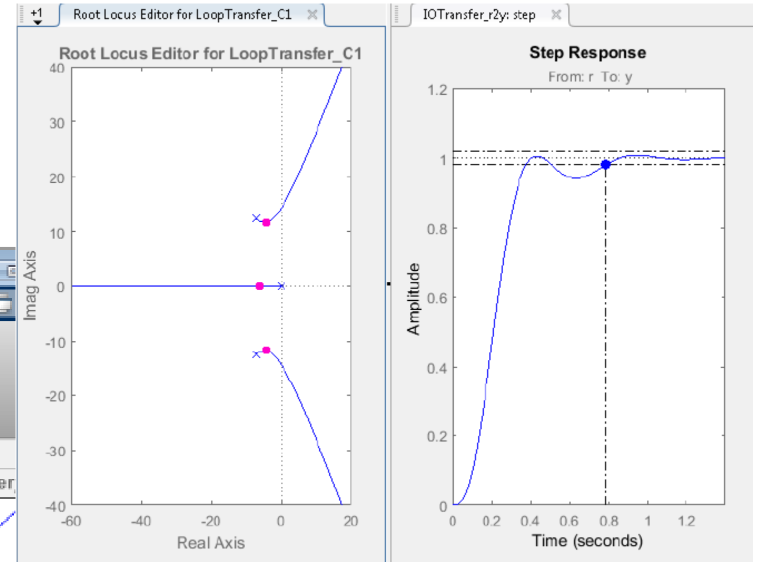


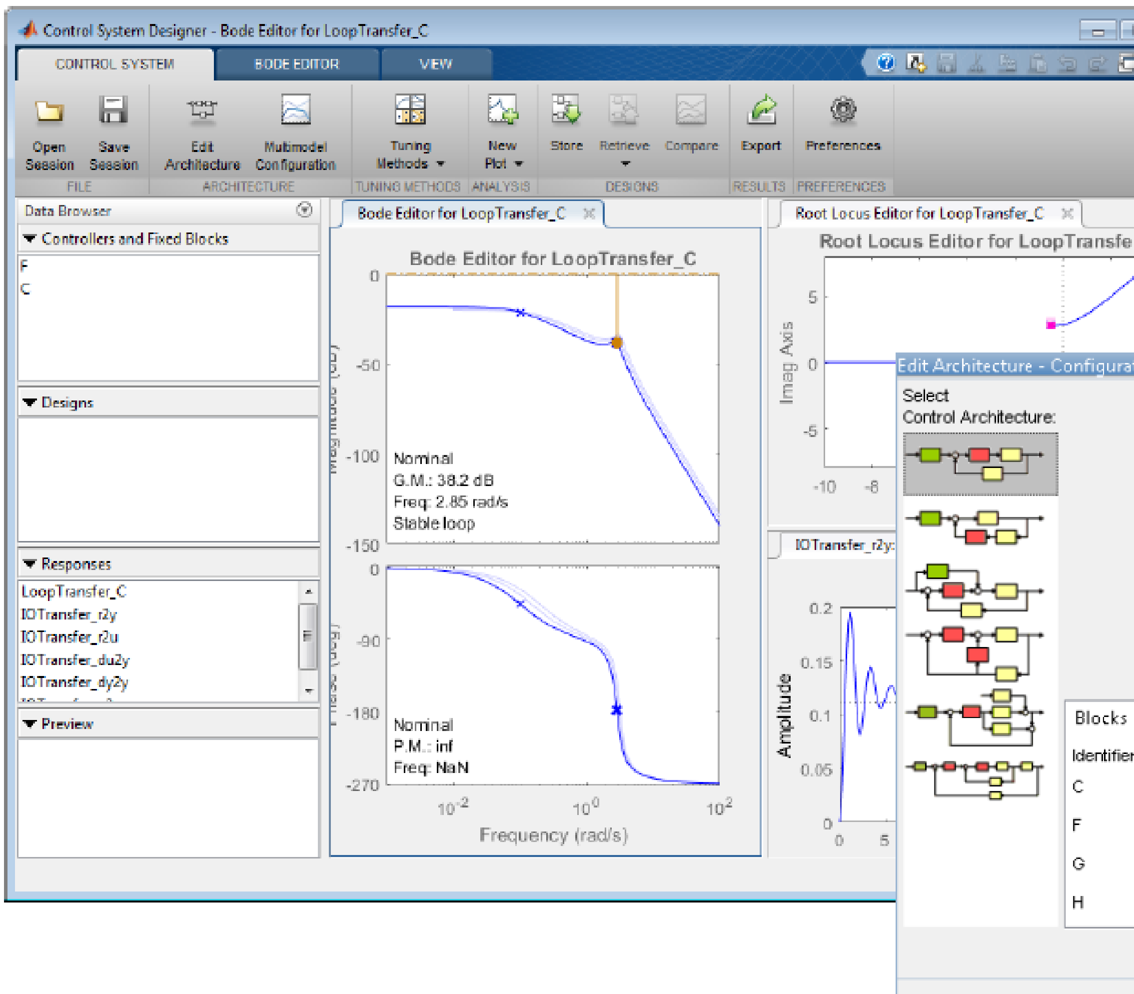
Example: 105-1, G4Exp3

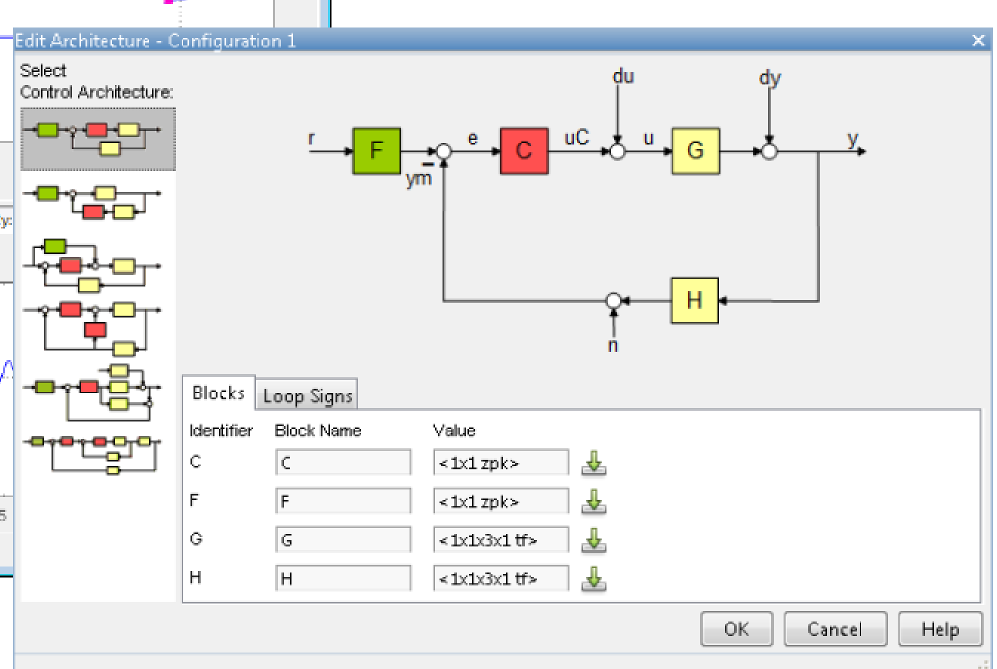
Simulation results and **actual system response** (velocity step : 1800 rpm)



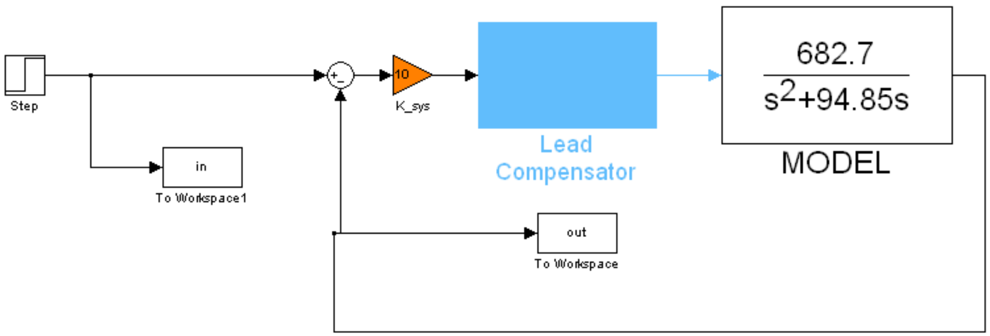
- Using Control System Designer, you can tune compensators using various graphical and automated tuning methods.







Example: 105-1, G4Exp3

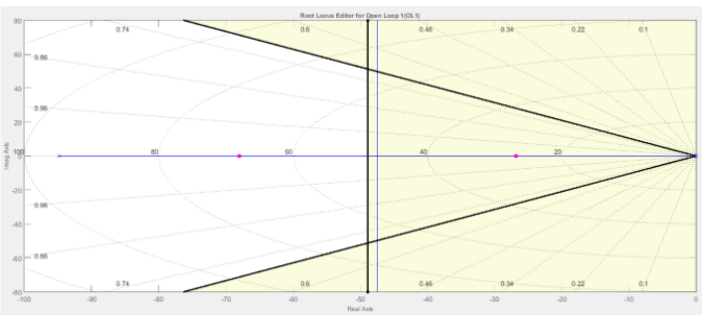


	Estimated Model Resp.		Real Motor Response		Pass?
	Ts	OS(%)	Ts	OS(%)	Ts<0.12 OS<5%
Without Ctrl	0.088	12.01	0.071	11.05	X
1st	0.767	24.01	0.466	12.62	X
2nd	0.078	41.77	0.07	16.94	X
3rd	0.08	37.49	0.075	16.83	X
4th	0.071	35.98	0.078	17.55	X
5th	0.072	35.71	0.081	17.32	X
6th	0.242	32.14	0.895	11.89	X
7th	2.245	0.001	1.633	5.376	X
8th	4.909	0.043	0.055	4.941	V

Open-Loop Model:

$$G(s)=\frac{682.7}{s^2+94.85s}$$

Root 無法同時落在合格範圍

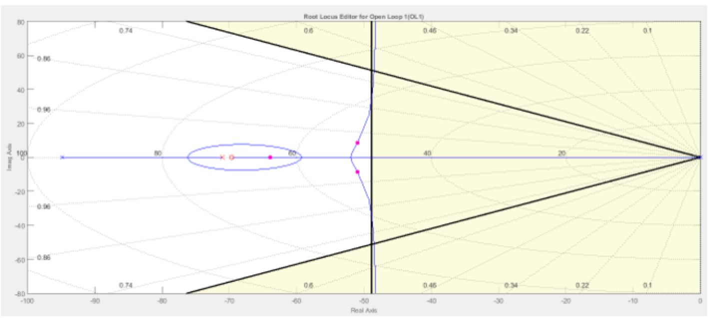


加入 Lead補償

使 Roots 們落在合格範圍

$$G_c(s)=\frac{k(s+z)}{s+p}$$

$$\begin{cases} z=-69.65 \\ p=-71.02 \\ k=3.521 \end{cases}$$



Open-Loop Model:

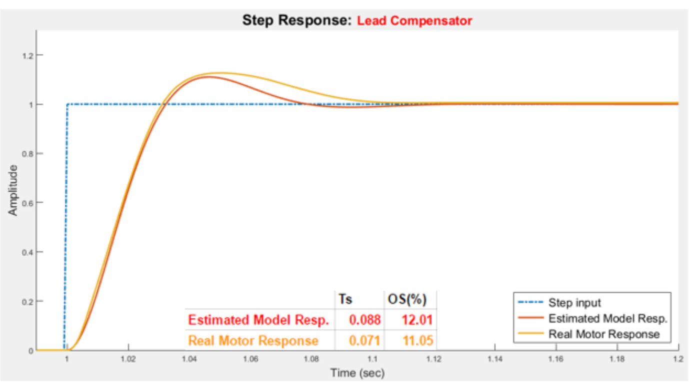
$$G(s)=\frac{682.7}{s^2+94.85s}$$

Ts 尚未控制就已符合標準

=> 在此改變標準：

Ts < 0.08 sec

OS < 5%

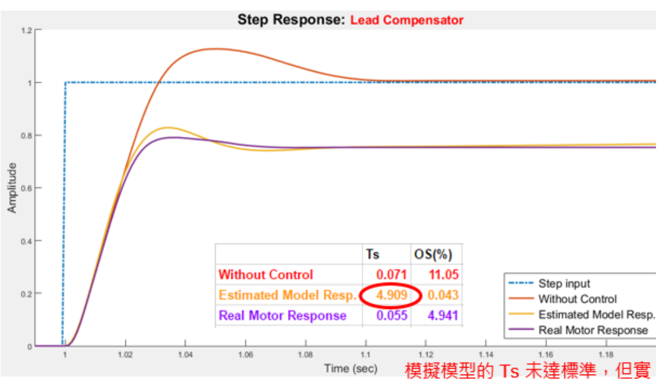


加入 Lead補償

使 Roots 們落在合格範圍

$$G_c(s)=\frac{k(s+z)}{s+p}$$

$$\begin{cases} z=-0.6559 \\ p=-34.99 \\ k=1.381 \end{cases}$$



模擬模型的 Ts 未達標準，但實際模型已達標 => 設計完成