

108-1 (Fall 2019)

自動控制實驗

Laboratory for Automatic Control

Week 02

# Control Toolbox

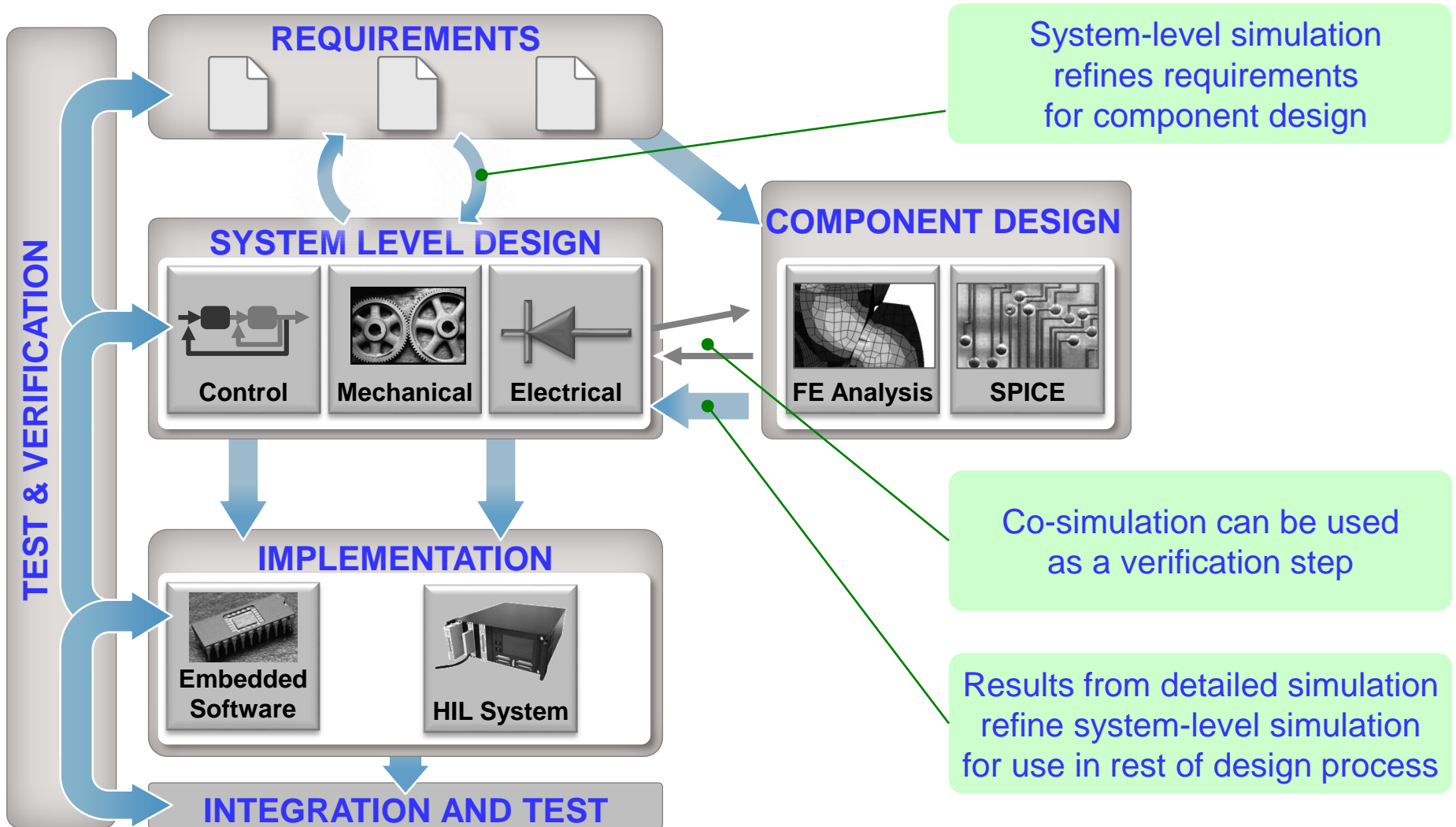
Prof. Feng-Li Lian

NTU-EE

Sep 2019 – Jan 2020

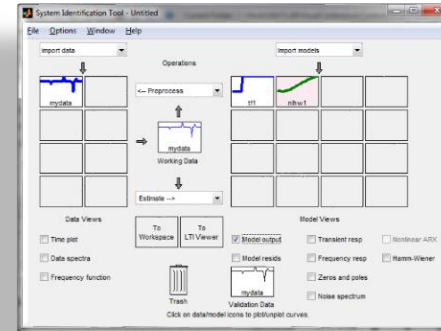


# Model-Based Design Process: System and Component Level Design



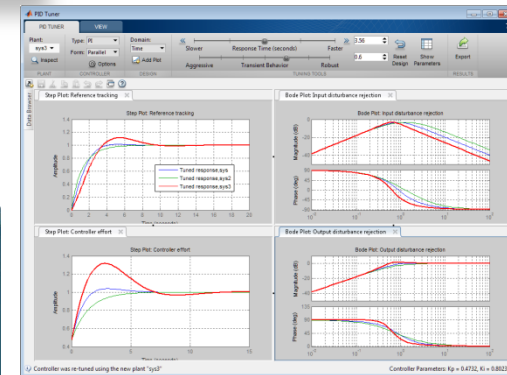
## System Identification Toolbox

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



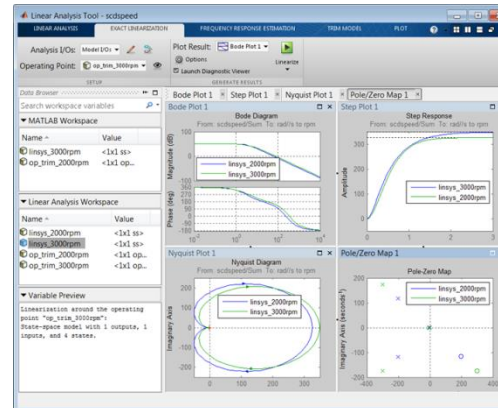
## PID Tuner

- Interactive tuning of PID controllers



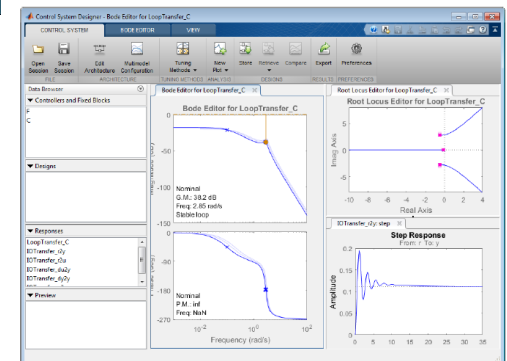
## Linear System Analyzer

- Improved linear analysis workflow

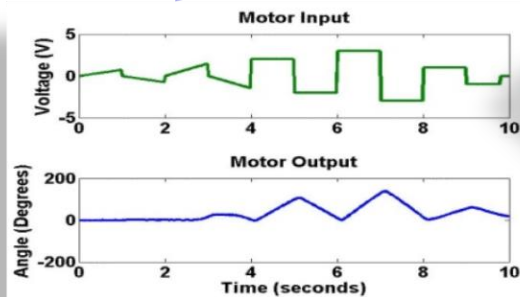
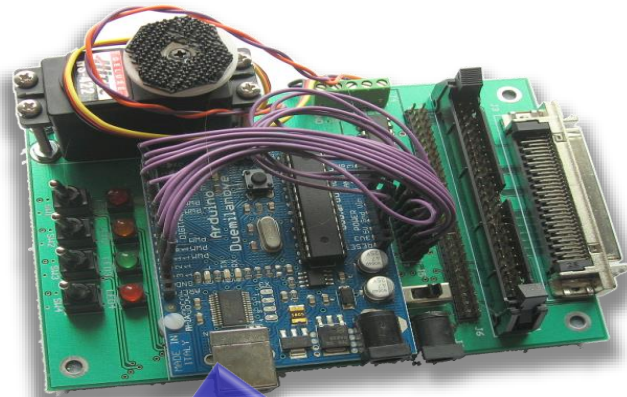


## Control System Designer

- Tune compensators using various graphical and automated tuning methods



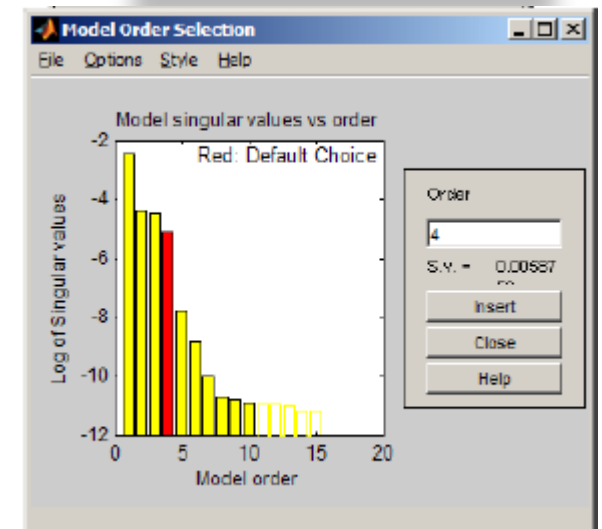
## System Identification Toolbox

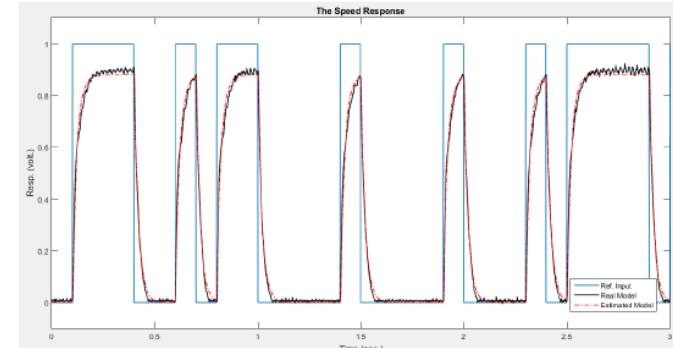
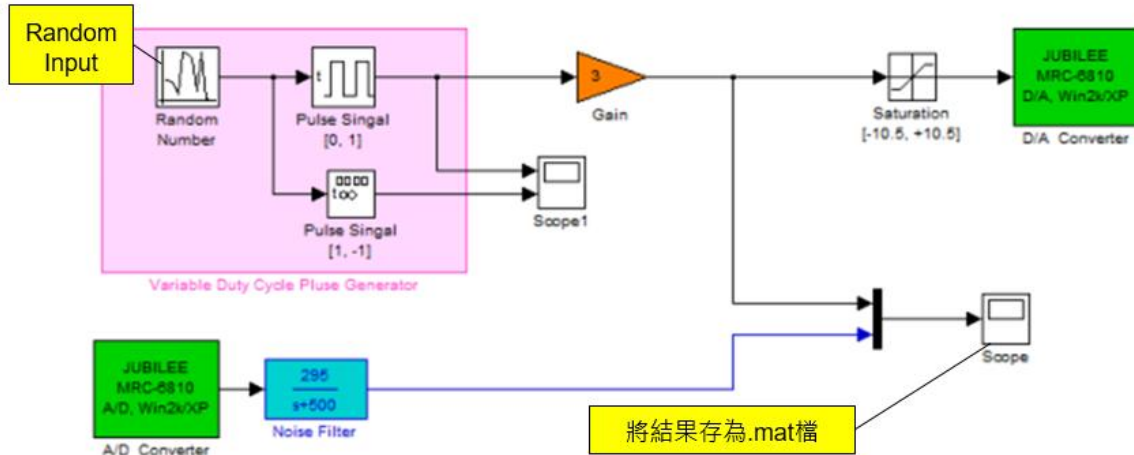


$$G(s) = \frac{1.4e9}{1 + 2.8e7s} e^{-0.1s}$$

Use [System Identification Toolbox](#)

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





## 實驗原理

統計建模工具

MATLAB system Identification Toolbox

1. import data  
(time domain signal)

2. 從 .mat 檔讀入 TF  
的 input 和 output

5. Preprocessing  
-> remove means

3. Sampling time  
改為 0.001

4. Import

6. Estimate TF

統計建模:

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

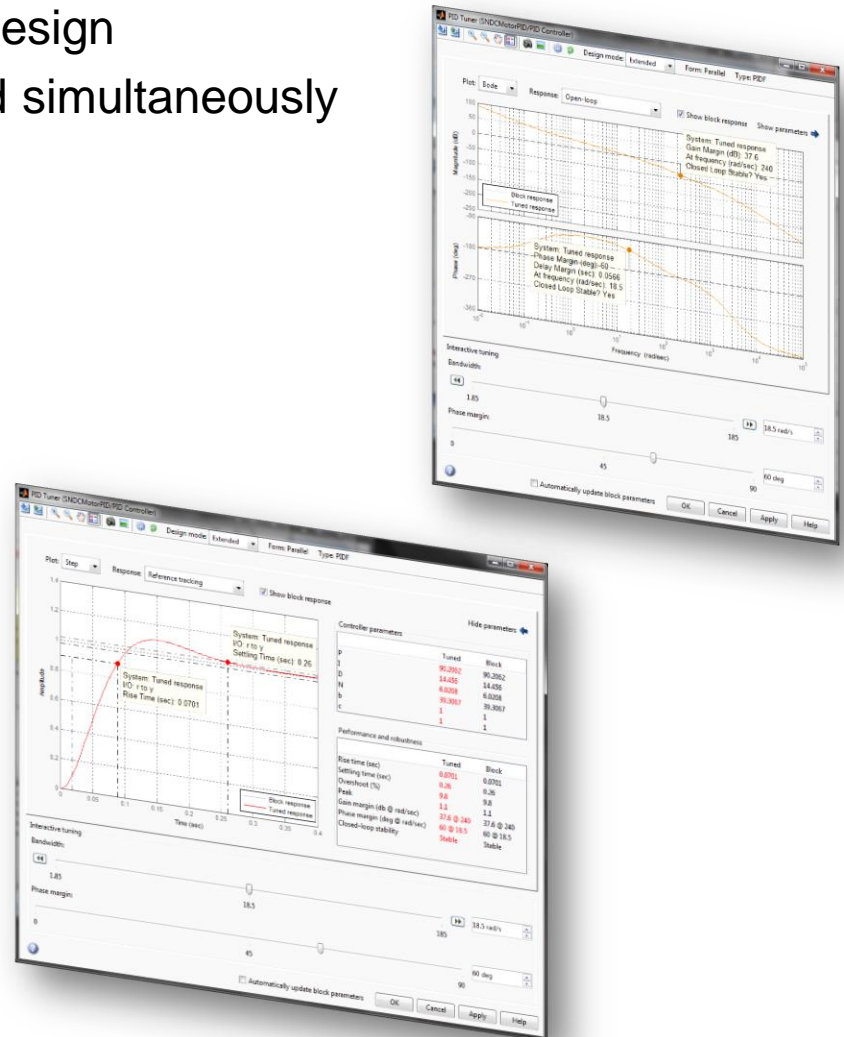
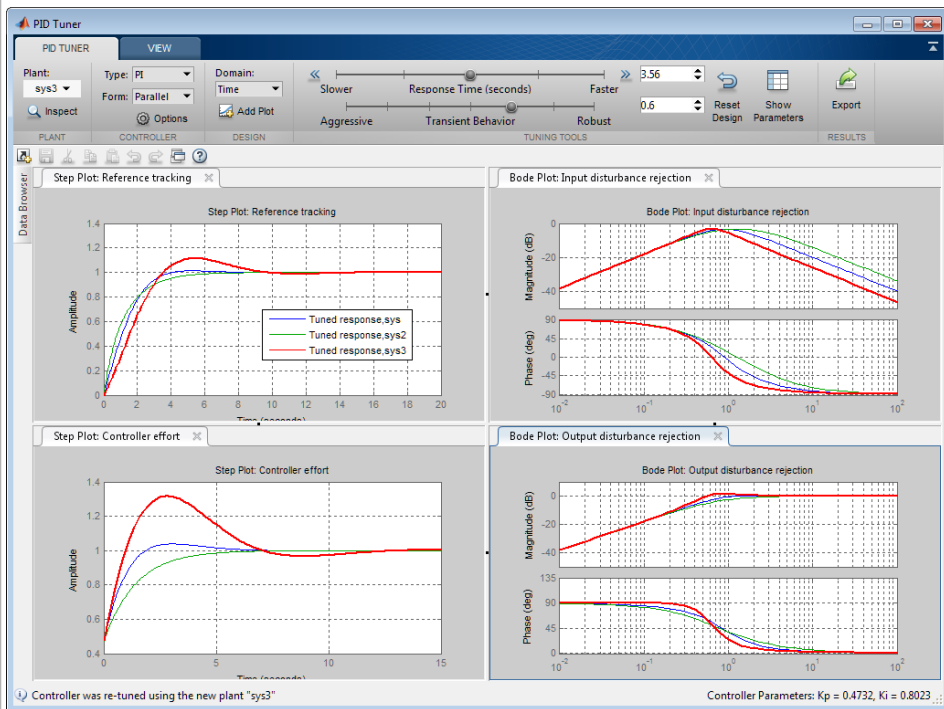
(Fit to estimation data: 92.5%)

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



## PIDTOOL: 統計模型

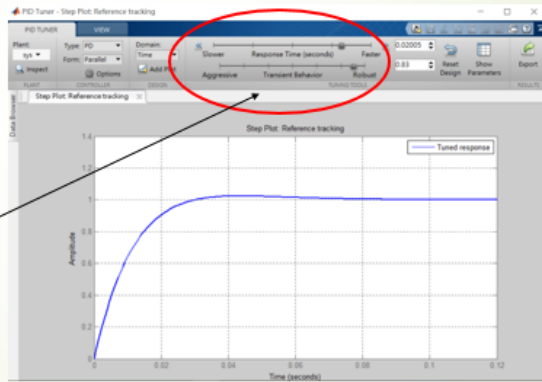
1. matlab code:

```
1 - sys = tf(217.4, [1 46.65 0])
2 - 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

➤  $K_p = 26.8276$

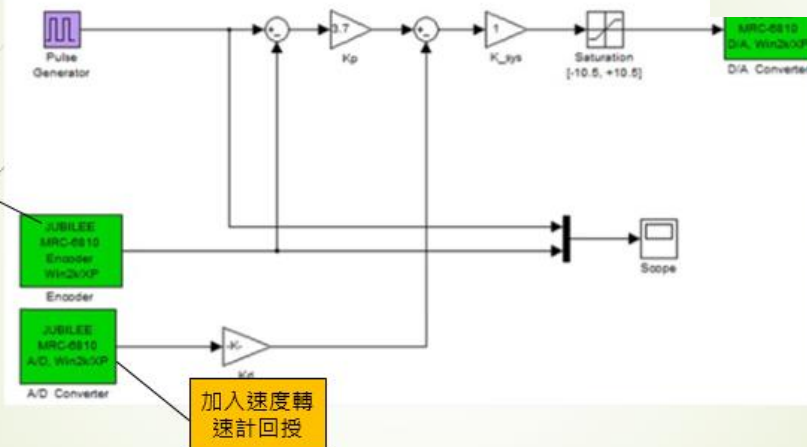
➤  $K_D = 0.42933$

➤ Settling time = 0.0528s

➤ Overshoot = 2.39%

Controller Parameters	
Kp	Tuned 26.8276
Ki	
Kd	0.42933
Tf	
Performance and Robustness	
Rise time	Tuned 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

Encoder測角位移回授

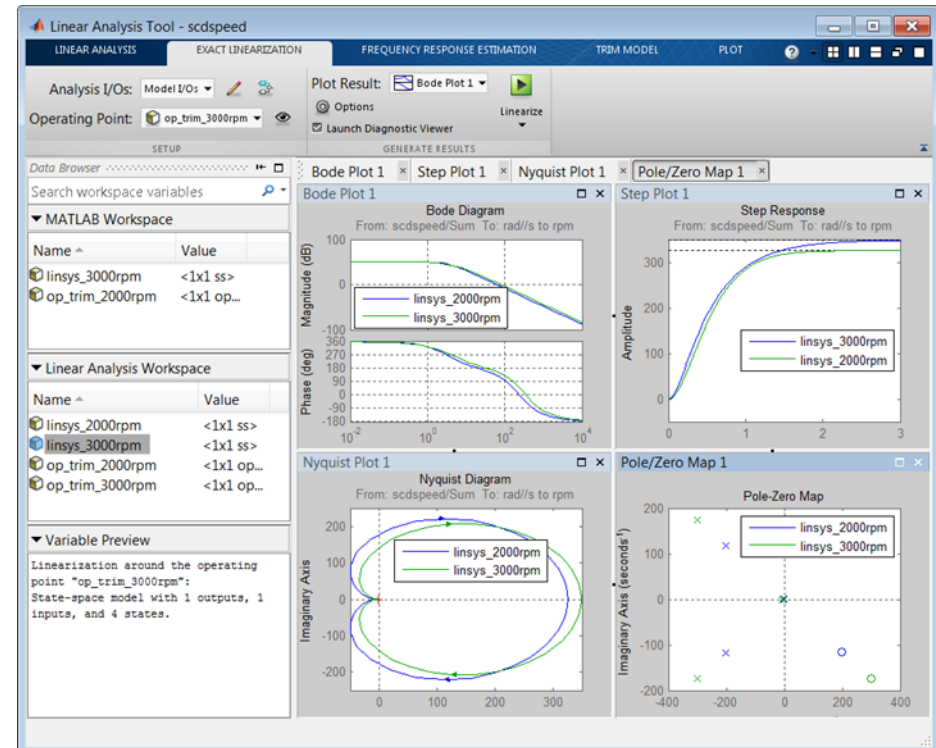
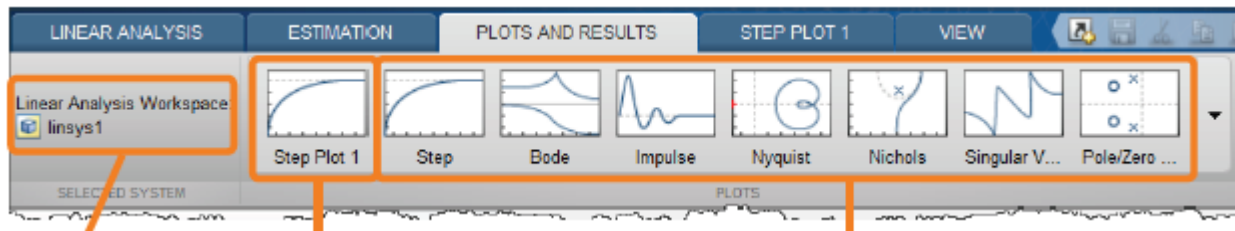


加入速度轉速計回授

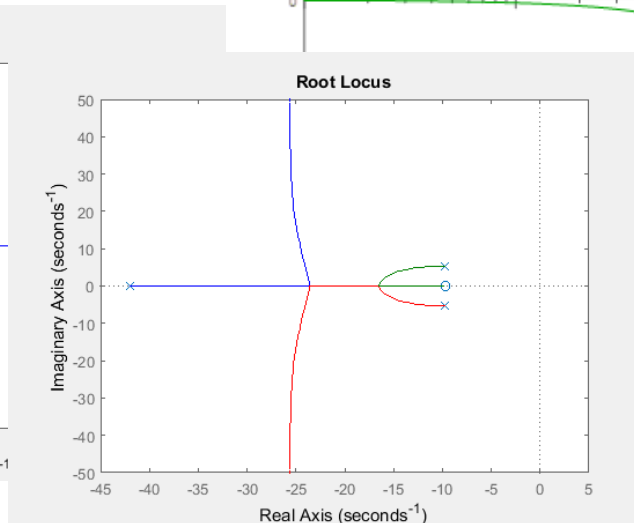
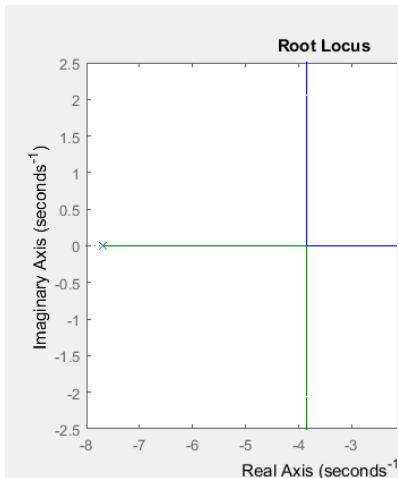
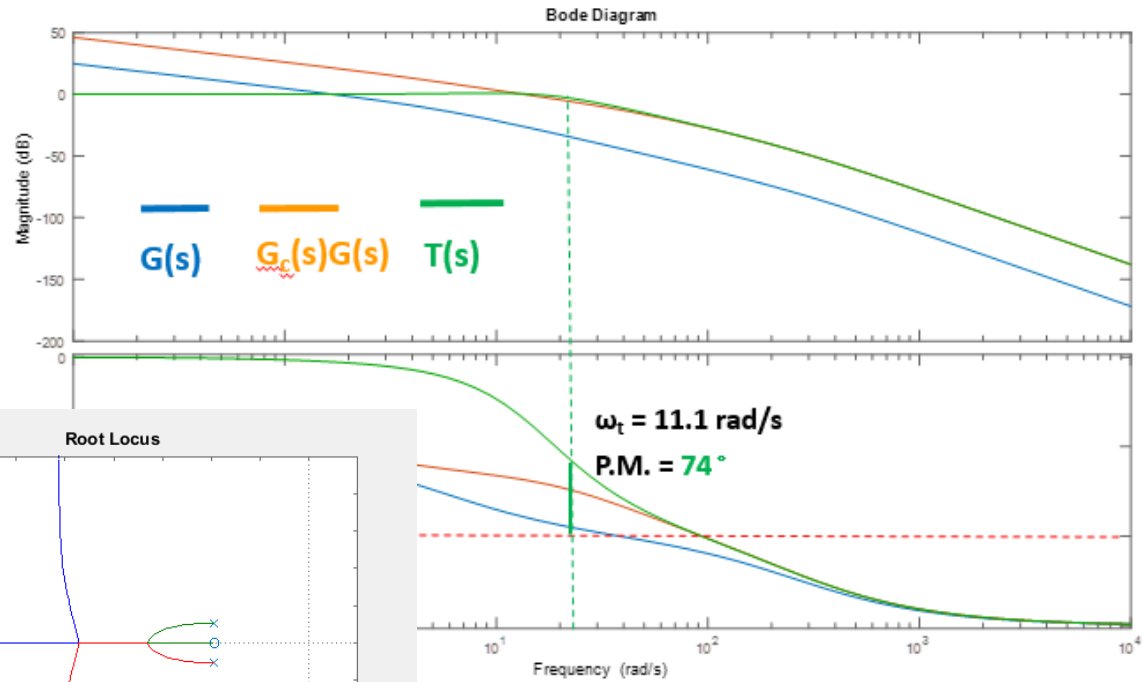
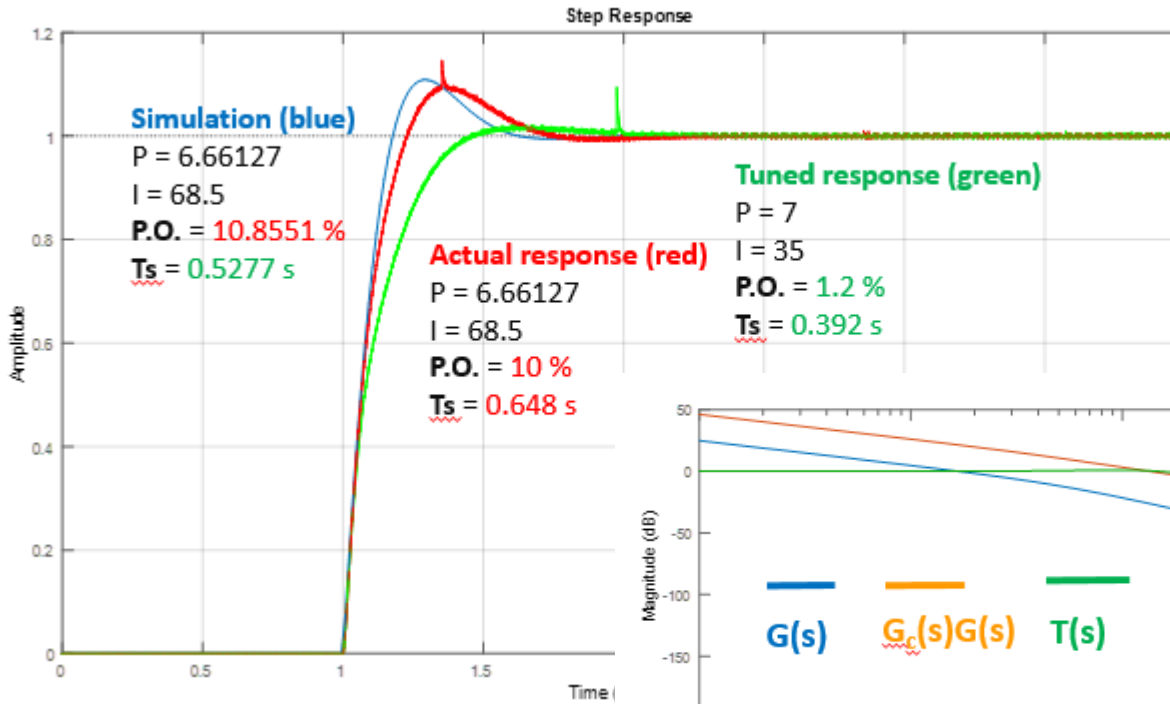


## ■ Improve linear analysis workflow

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



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



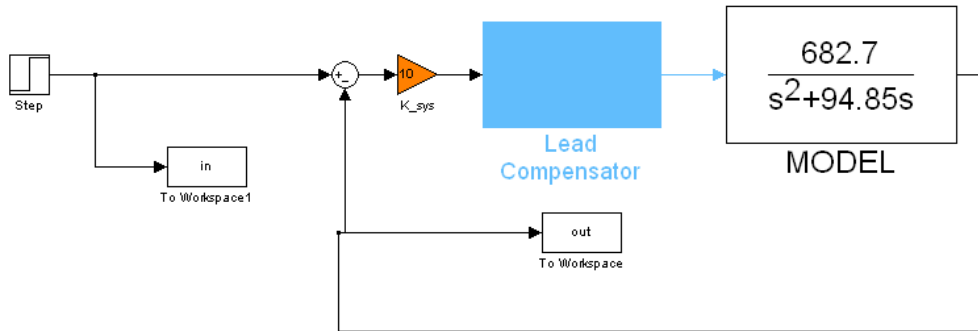
- Using **Control System Designer**, you can tune compensators using **various graphical** and **automated tuning** methods.

The screenshot displays the Control System Designer software interface with several key components:

- Bode Editor for LoopTransfer\_C:** Shows two Bode plots. The top plot displays the magnitude response (G.M.: 38.2 dB, Freq: 2.85 rad/s, Stable loop). The bottom plot displays the phase response (Nominal P.M.: inf, Freq: NaN).
- Root Locus Editor for LoopTransfer\_C1:** Shows the root locus plot in the complex plane, with poles marked by 'x' and zeros by 'o'.
- Step Response:** A plot showing the system's response to a step input, with Amplitude on the y-axis and Time (seconds) on the x-axis.
- Edit Architecture - Configuration 1:** A block diagram showing the control architecture with blocks F, C, G, and H. The diagram includes feedback loops and signal paths labeled with 'r', 'e', 'u', 'y', and 'n'.

The Edit Architecture dialog box includes a table for block configuration:

Identifier	Block Name	Value
C	C	<1x1 zpk>
F	F	<1x1 zpk>
G	G	<1x1x3x1 tf>
H	H	<1x1x3x1 tf>

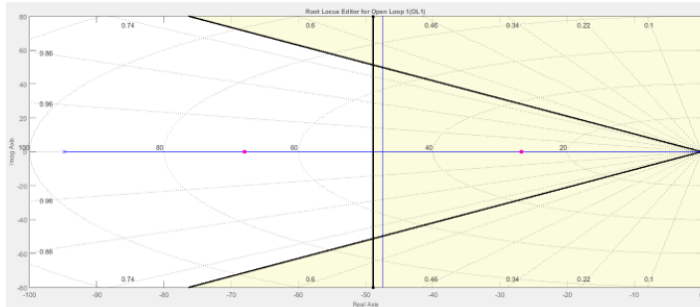


	Estimated Model Resp.		Real Motor Response		Pass?	
	Ts	OS(%)	Ts	OS(%)	Ts < 0.12	OS < 5%
Without Ctrl	0.088	12.01	0.071	11.05		X
1st	0.767	24.01	0.466	12.62		X
2nd	0.078	41.77	0.07	16.94		X
3rd	0.08	37.49	0.075	16.83		X
4th	0.071	35.98	0.078	17.55		X
5th	0.072	35.71	0.081	17.32		X
6th	0.242	32.14	0.895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

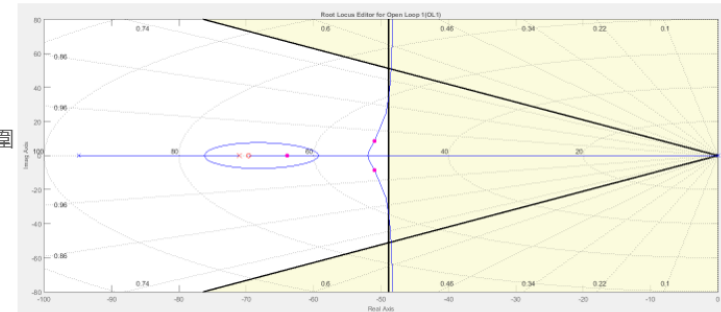


加入 Lead補償

使 Roots 們落在合格範圍

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

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



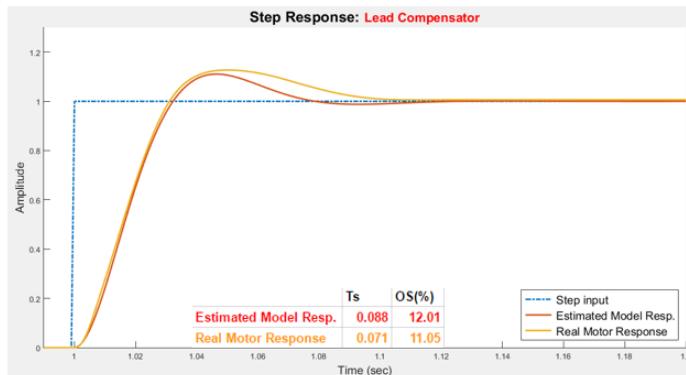
Open-Loop Model:

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

Ts 尚未控制就已符合標準

=> 在此改變標準:

**Ts < 0.08 sec**  
**OS < 5%**

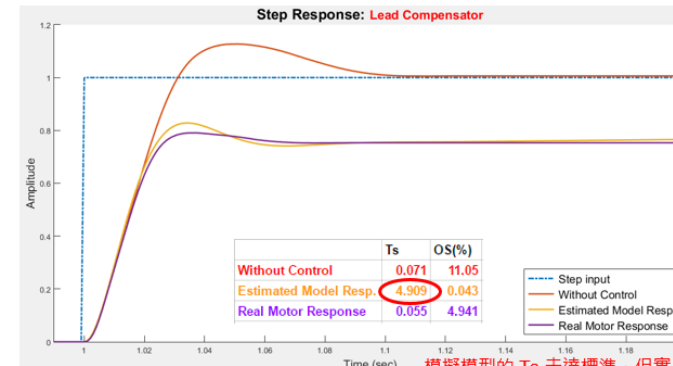


加入 Lead補償

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