智慧型控制及應用實驗室
ICALAB

An Improved Neural Network Algorithm to Efficiently Track Various Trajectories of Robot Manipulator Arms

MAHMOUD ELSISI1,2, KARAR MAHMOUD 3,4, MATTI LEHTONEN 3 , AND MOHAMED M. F. DARWISH 2,3
Received December 25, 2020, accepted January 11, 2021, date of publication January 14, 2021, date of current version January 22, 2021. Digital Object Identifier 10.1109/ACCESS.2021.3051807

Advisor : Ching-Hung Lee

Presenter : Shi-Xian Yang

Date : 2023/1/3

Outlines

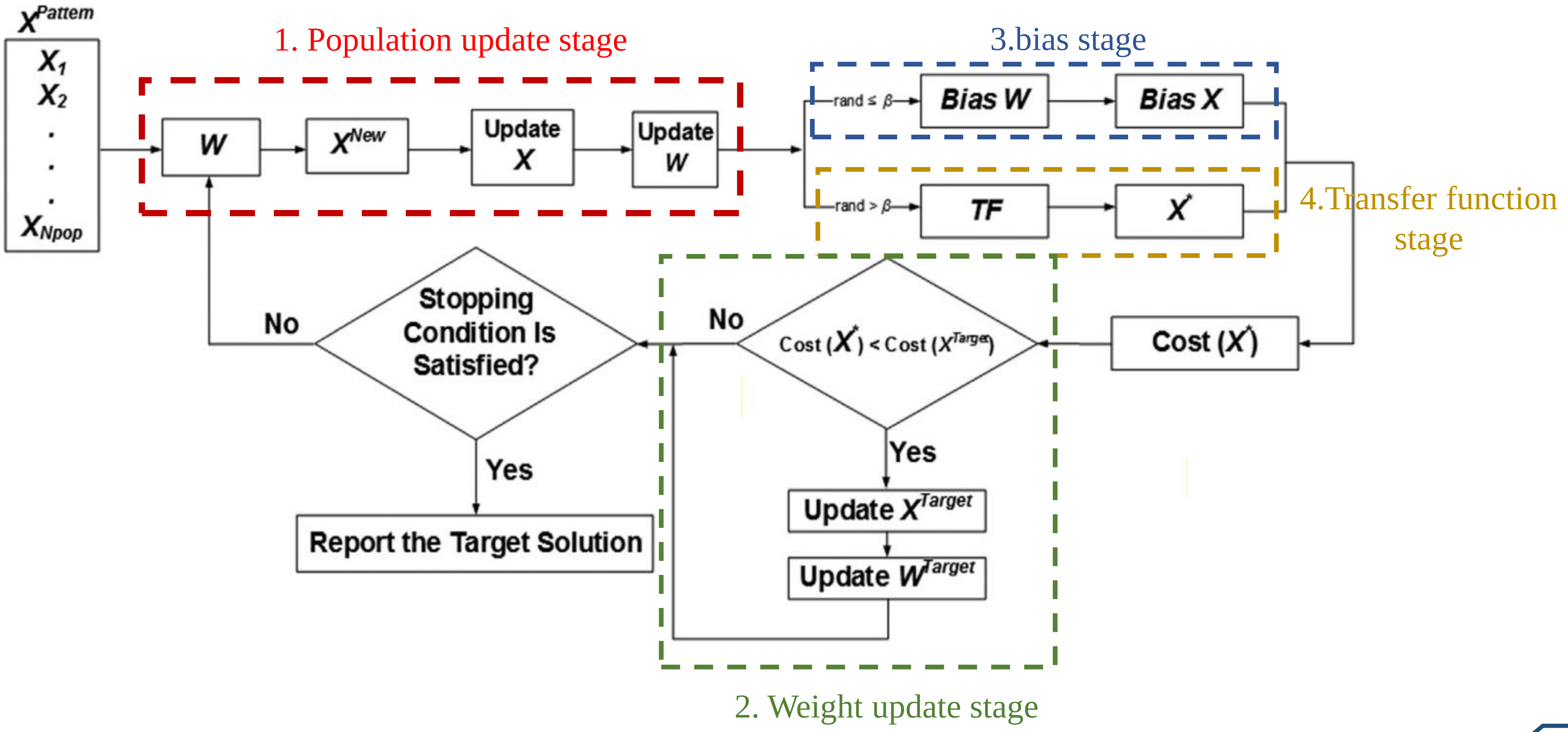
- Recap
- NNA Algorithm
- System modeling
- Experiment and Result
  - Comparison
  - Performance index
  - Robustness Experiment

Recap

- A new polynomial mutation is applied to promote the exploration manner of the original NNA without initial parameters.
- The new algorithm is introduced to obtain the optimal gains of the robot manipulator controller instead of conventional procedures of designer expertise.
- A new performance index is created to guarantee the decreasing of the settling time and the overshoot at the same time.
- The progress of the inspired procedure is confirmed against various trajectories and system parameter variations.

Recap

- **Neural Network algorithm**



ICALAB 智慧型控制及應用實驗室

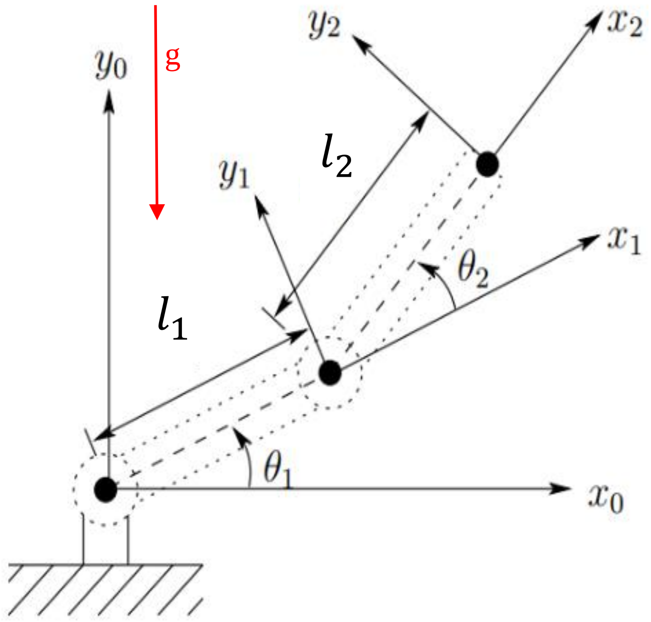
NNA Algorithm

- **Pseudocode of whole process**

Start MNNA
Simulate the manipulator including the chosen controller
Determine the performance function (J)
Select the best solution and the best weights
While t<Max iteration
    Carry out the steps of MNNA
    Simulate the manipulator including the chosen controller
    Obtain the performance function J
    Select the best fitness
END While
Stop

System modeling

- **System modeling**



Lagrange mechanics:

$$\tau = M(q)\ddot{q} + C(q,\dot{q}) + G(q)$$

Where

τ : *Torque vector*
$M(q)$: Inertia matrix
$C(q,\dot{q})$: Centrifugal and Coriolis effect
$G(q)$: Gravity

- Dynamic formulation:

$$\tau_1 = m_2 {l_2}^2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2(2\ddot{\theta}_1 + \ddot{\theta}_2)$$
$$+(m_1 + m_2){l_2}^2\ddot{\theta}_1 - m_2 l_1 l_2 s_2 {\dot{\theta}_2}^2$$
$$-2m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12}$$
$$+(m_1 + m_2) l_1 g c_1$$
$$\tau_2 = m_2 {l_2}^2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2(\ddot{\theta}_1)$$
$$+m_2 l_1 l_2 c_2 {\dot{\theta}_1}^2 + m_2 l_1 g c_{12}$$

System modeling

Lagrange mechanics:

$$\boldsymbol{\tau_i = M(q_i)\ddot{q}_i + C(q_i,\dot{q}_i) + G(q_i)}$$

- Challenge of tuning controller for robotic manipulator: →Joints coupling

$$M(q_i) = \begin{bmatrix} m_2{l_2}^2 + (m_1+m_2){l_2}^2 + 2m_2l_1l_2c_2 & m_2{l_2}^2 + m_2l_1l_2c_2 \\ m_2{l_2}^2 + m_2l_1l_2c_2 & m_2{l_2}^2 \end{bmatrix}$$

$$C(q_i,\dot{q}_i) = \begin{bmatrix} -2m_2l_1l_2s_2\dot{q}_1\dot{q}_2 - m_2l_1l_2s_2{\dot{q}_2}^2 \\ m_2l_1l_2c_2{\dot{q}_1}^2 \end{bmatrix}$$

$$G(q_i) = \begin{bmatrix} m_2l_2gc_{12} + (m_1+m_2)l_1gc_1 \\ m_2l_1gc_{12} \end{bmatrix}$$

- **Control signal of the PID control**

$$\tau_i = K_{p,i}e_i + K_{I,i}\int e_i \cdot dt + K_{D,i}\frac{de_i}{dt}, \qquad i = 1,2$$

$$e_i = q_{d,i} - q_i$$

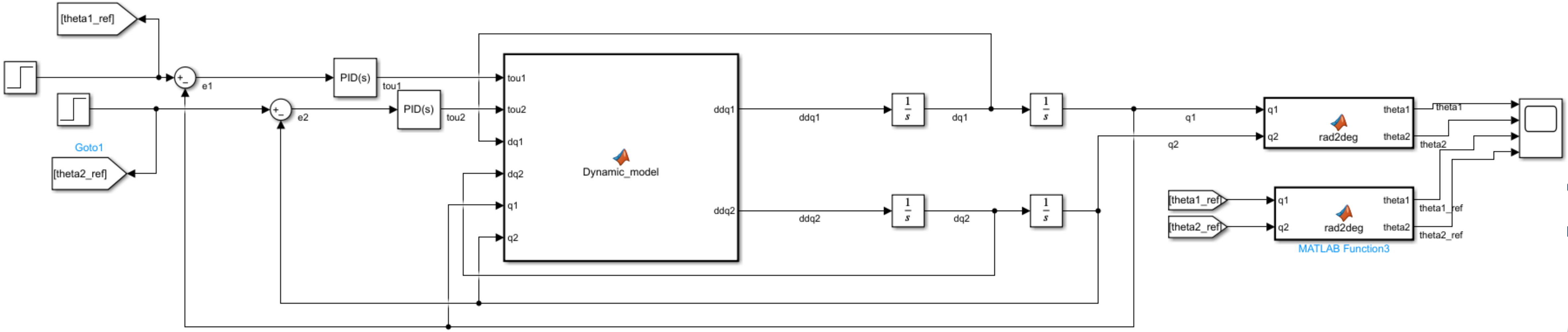
System modeling

$$\textbf{step1}:\ \tau[k] = K_p e[k] + K_d\left(\frac{e[k]-e[k-1]}{dt}\right) + K_i\sum_{j=0}^{k} e[j]\cdot dt$$

$$\textbf{step2}:\ \ddot{q}[k] = M^{-1}(q[k-1])(\tau[k] - C(q[k-1],\dot{q}[k-1]) - G(q[k-1]))$$

$$\textbf{step3}:\ \dot{q}[k] = \dot{q}[k-1] + \ddot{q}[k]\cdot dt$$

$$\textbf{step4}:\ q[k] = q[k-1] + \dot{q}[k]\cdot dt$$



Experiment and Result

- Robotic dynamic Parameter:

$$m_1 = m_2 = 0.1kg$$
$$l_1 = 0.8m \;\; l_2 = 0.4m$$
$$g = 9.81m/s^2$$

- Max iteration: 100

- Population size: 50

- Controller gain limit:

	Link1		Link2	
	min	max	min	max
K_p	0	250	0	250
K_I	0	1	0	1
K_D	0	20	0	10

- Compare subjects
  - GA-PID
  - PSO-PID
  - MNNA-PID

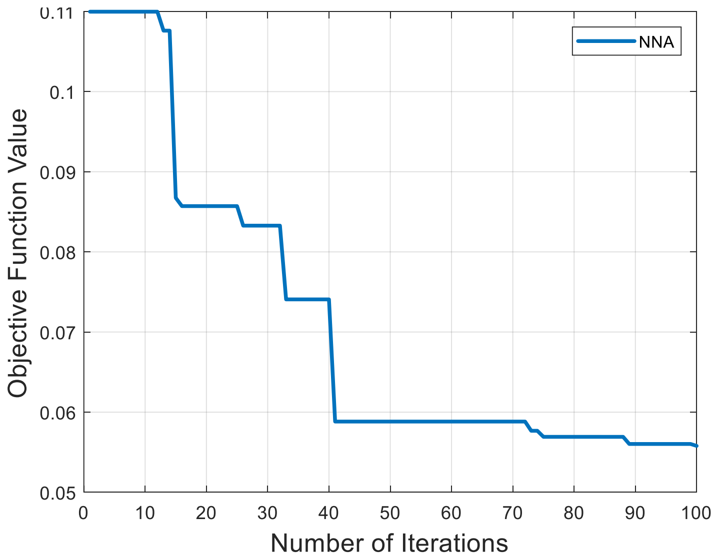
	Factors number	Tuning factor
GA-PID	4	Population size, iteration Crossover, mutation
PSO-PID	5	Population size, iteration C1,c2, w
MNNA-PID	2	Population size, iteration

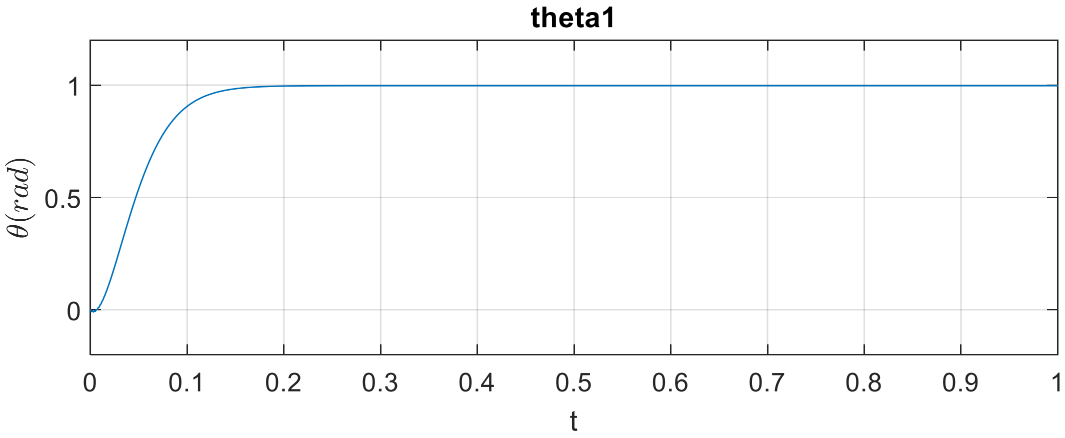
NNA result

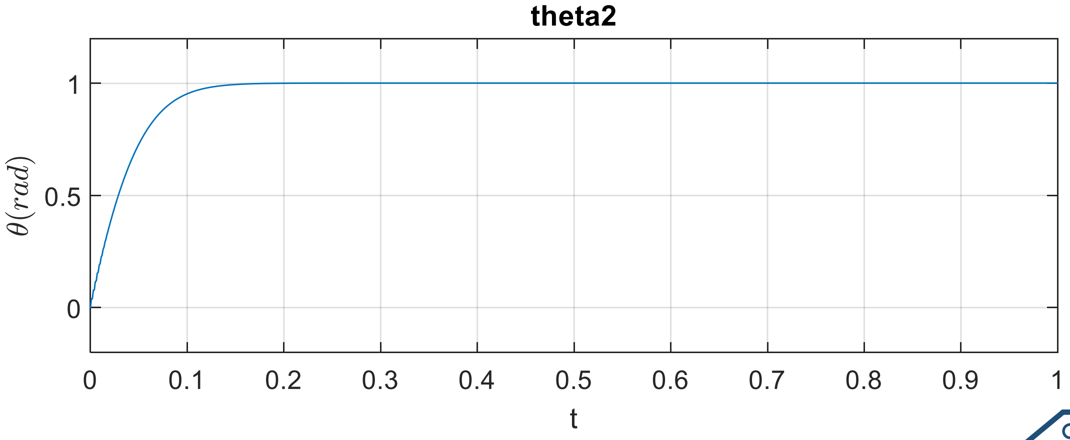
Tunning factor setting:
Max iteration: 100
Population size:50

Run: 1 Error: 0.055614 Elapsed_Time: 24.6562
============optimal solution==============

Link1	Link2
Kp1:250	Kp2:250
Ki1:0.41475	Ki2:0.59791
Kd1:13.4205	Kd2:8.8604



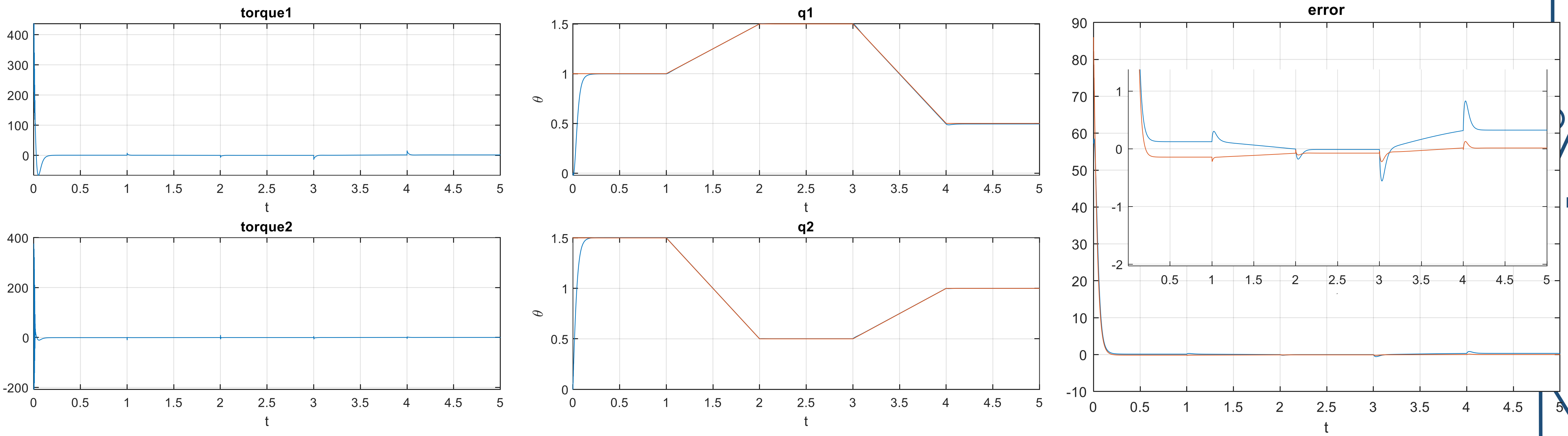




ICALAB
智慧型控制及應用實驗室

NNA result

- VARIOUS TRAJECTORIES

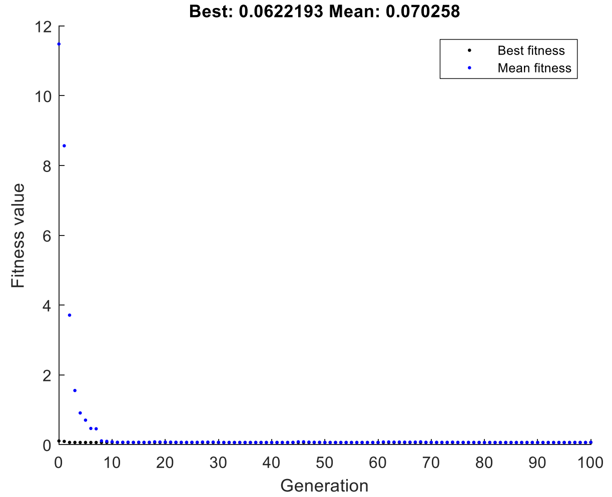


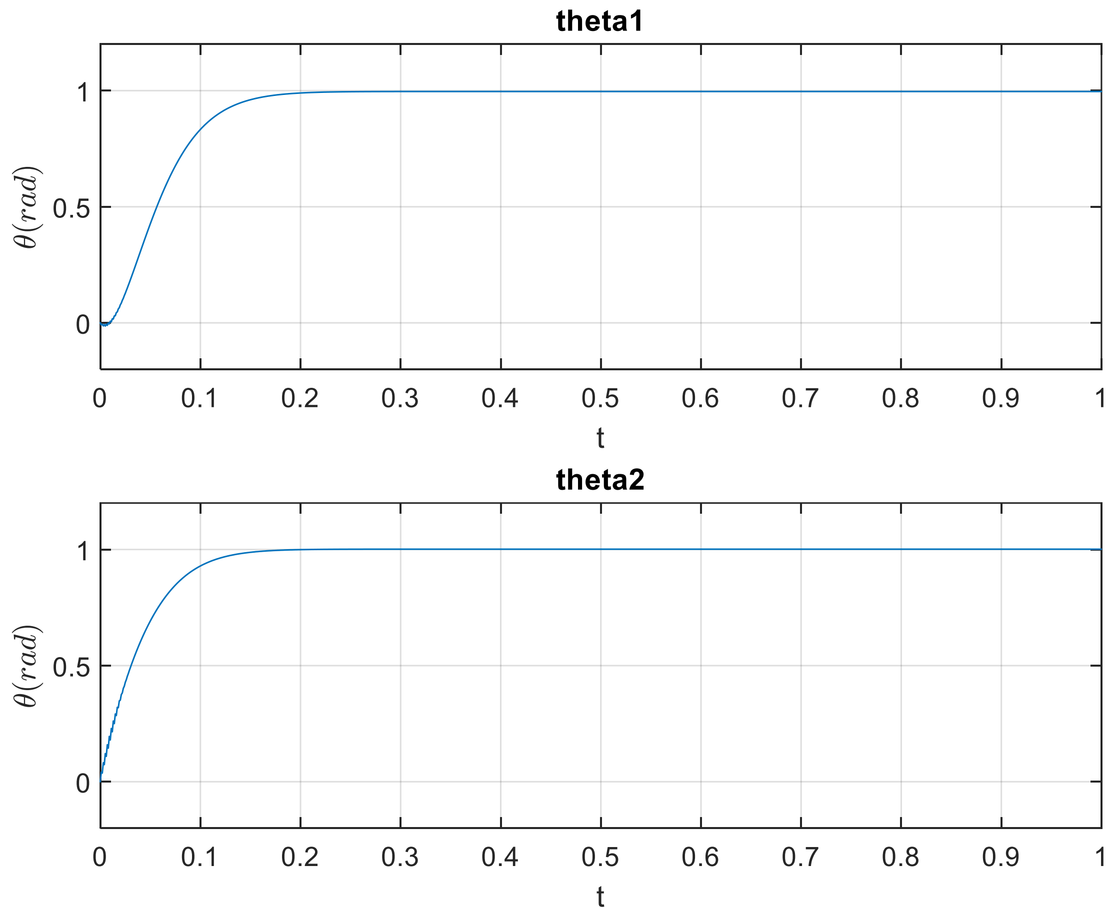
GA result

Tunning factor setting:
Max iteration: 100
Population size:50
Crossover: 0.8
Mutation: 0.1

Error: 0.058121   Elapsed Time: 23.9588
=========optimal solution=========

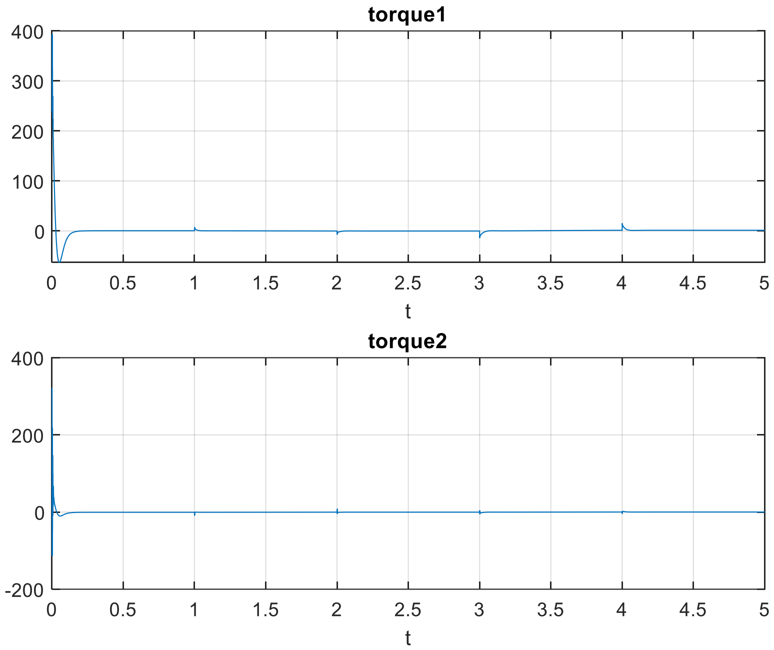
Link1	Link2
Kp1:193.8595	Kp2:223.0862
Ki1:1	Ki2:0.020764
Kd1:11.7447	Kd2:8.6229

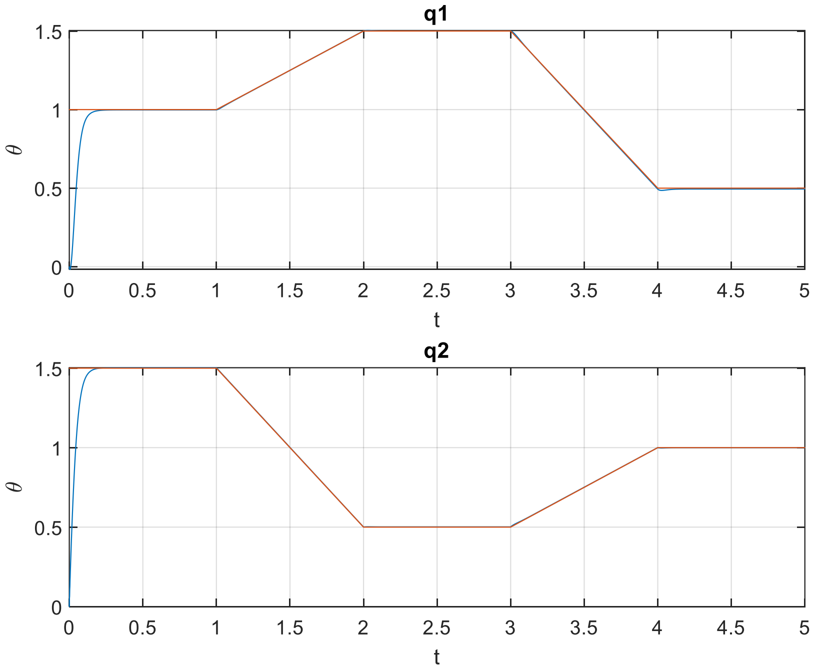


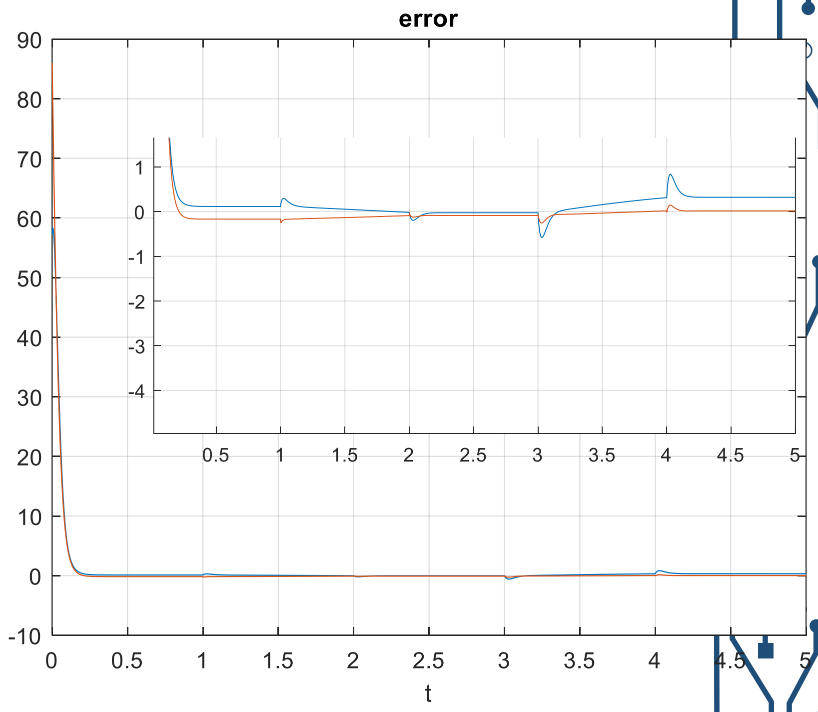


GA result

- VARIOUS TRAJECTORIES







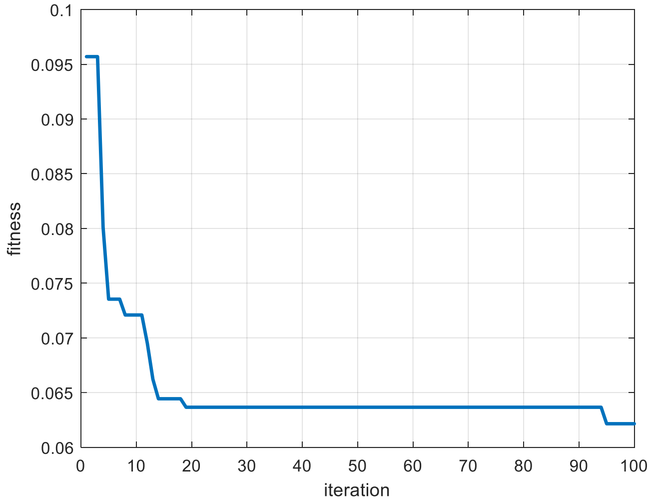
PSO result

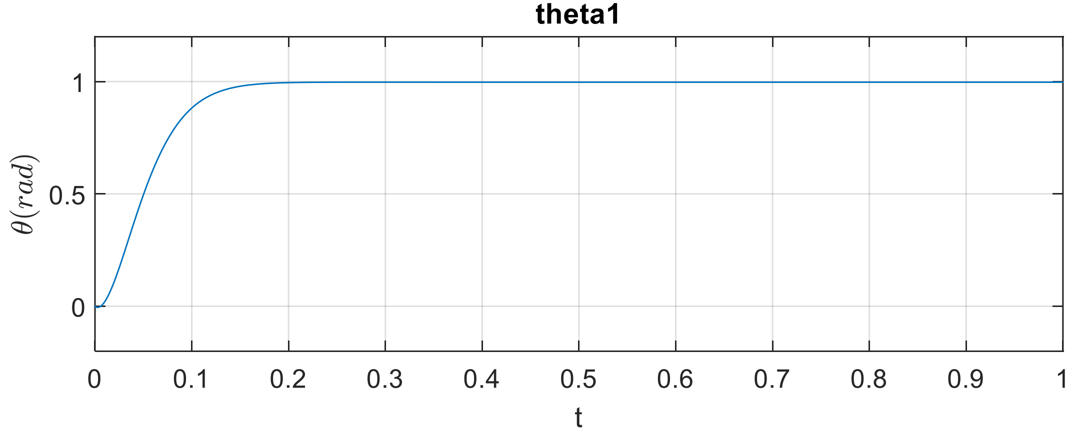
Tunning factor setting:
Max iteration: 100
Population size:50
Personal Learning Coefficient(c1): 1.5
Global Learning Coefficient(c2): 0.12
Inertia Weight Damping Ratio(w): 0.99

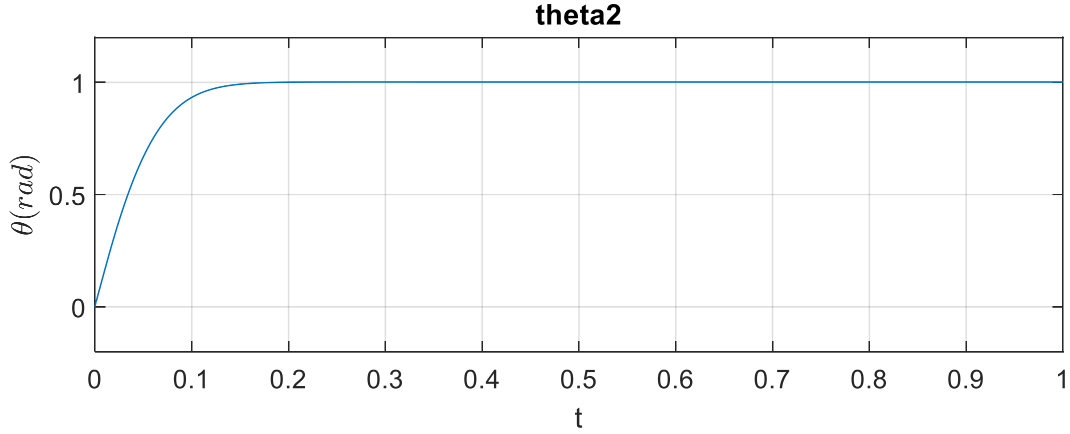
Error: 0.062151 Elapsed Time:23.7006

=============optimal solution============

Link1	Link2
Kp1:218.4591	Kp2:161.4128
Ki1:1	Ki2:0.94015
Kd1:12.492	Kd2:6.4216



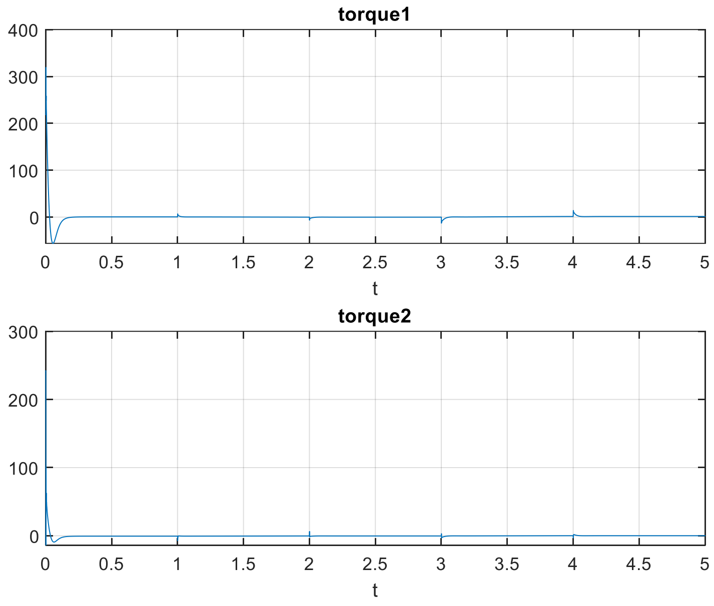


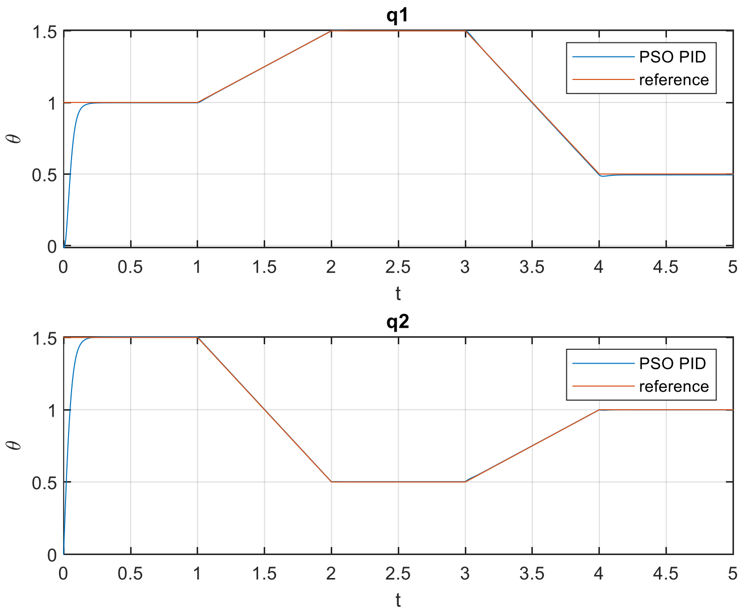


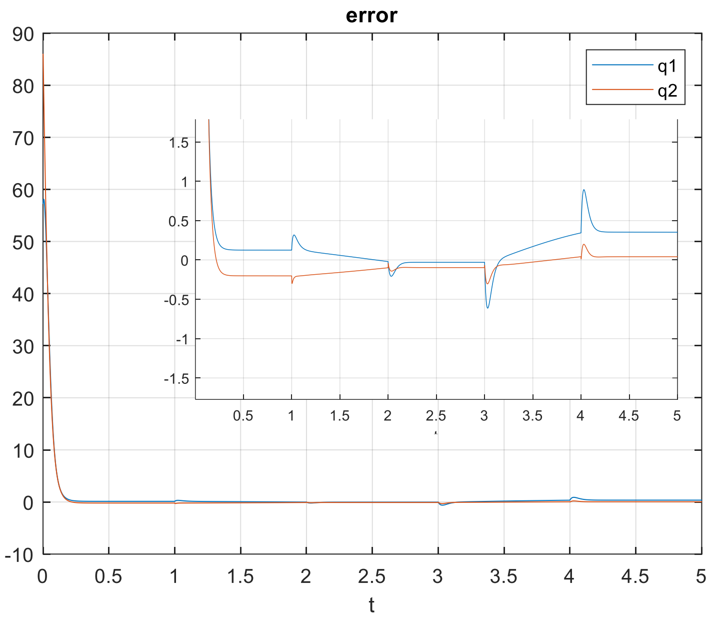
ICALAB
智慧型控制及應用實驗室

PSO result

- VARIOUS TRAJECTORIES







Comparison

		GA-PID	PSO-PID	NNA-PID
Controller Parameters	Link1	$K_P = 193.86$	$K_P = 218.45$	$K_P = 250$
		$K_I = 1$	$K_I = 1$	$K_I = 0.414$
		$K_D = 11.74$	$K_D = 12.492$	$K_D = 13.42$
	Link2	$K_P = 223.08$	$K_P = 161.41$	$K_P = 250$
		$K_I = 0.02$	$K_I = 0.94$	$K_I = 0.59791$
		$K_D = 8.62$	$K_D = 6.42$	$K_D = 8.8604$
J		0.05812	0.062151	0.055614

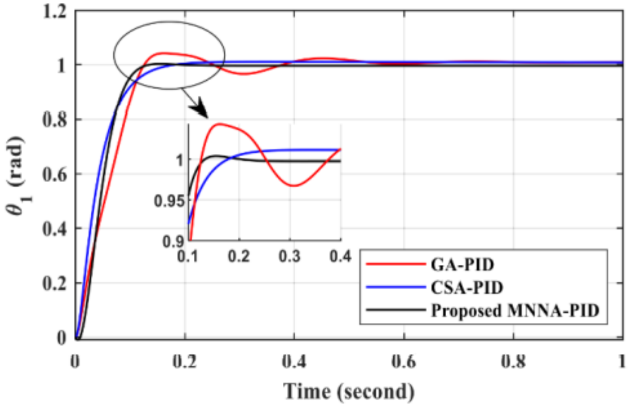


Figure 2 Link1 step response

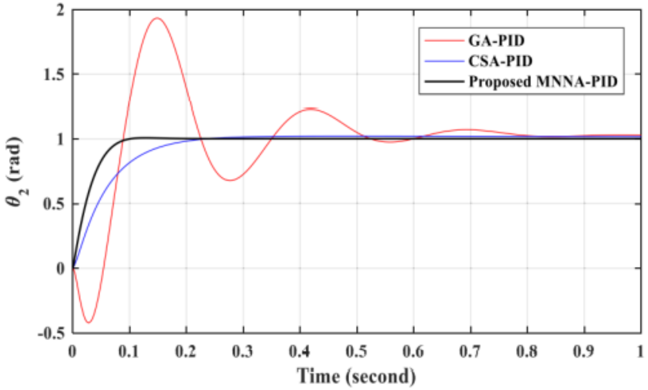
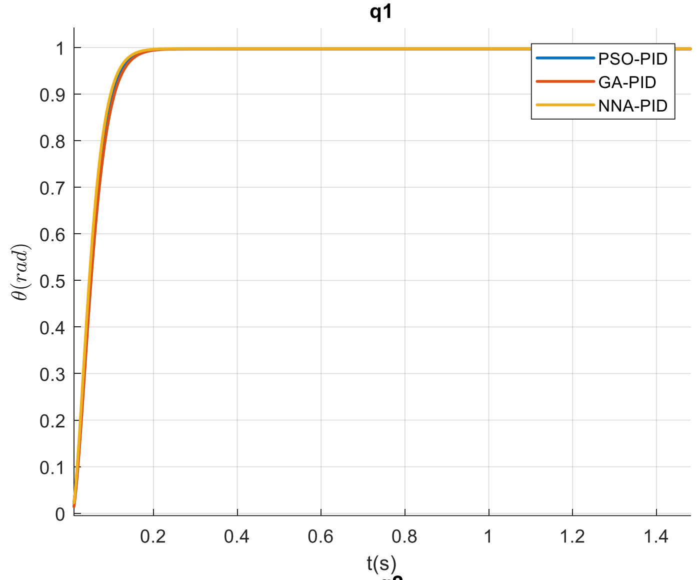
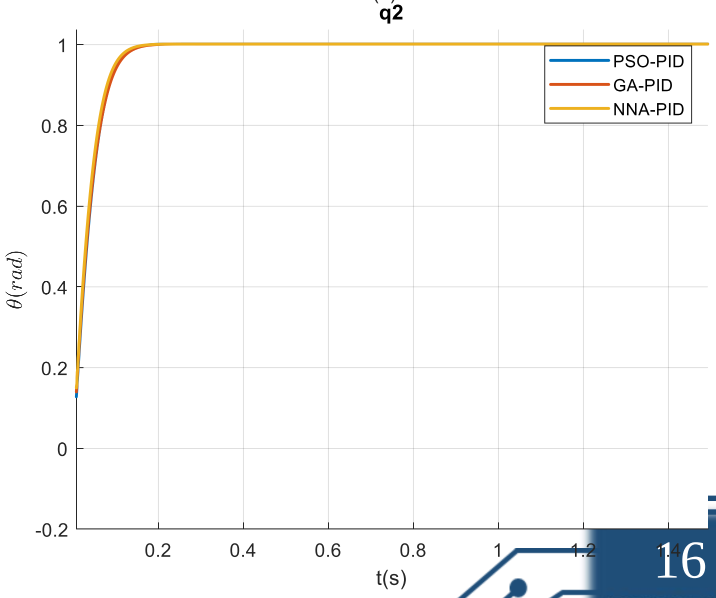
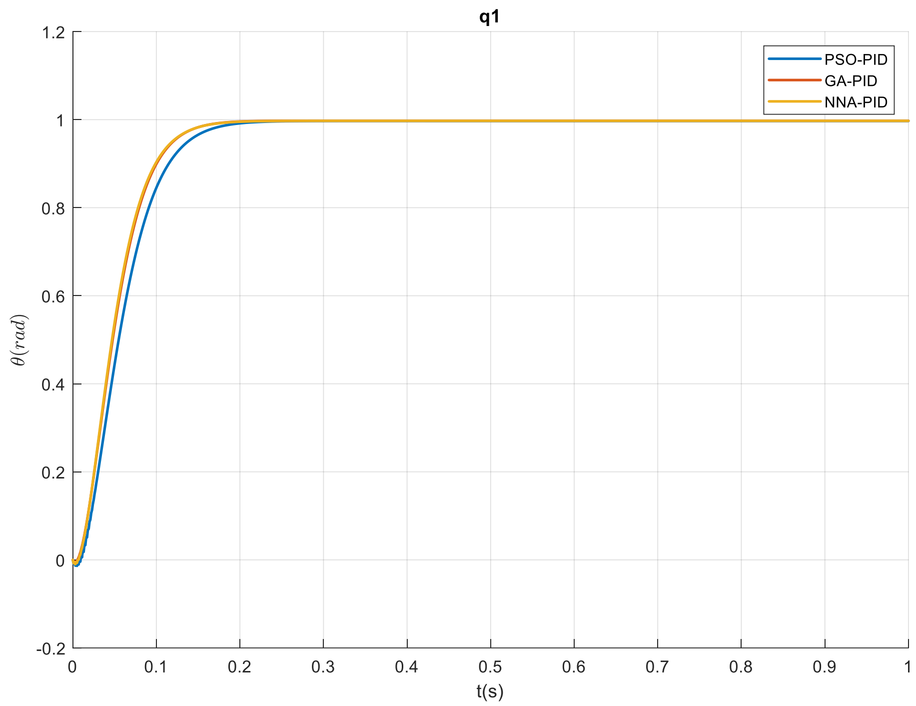


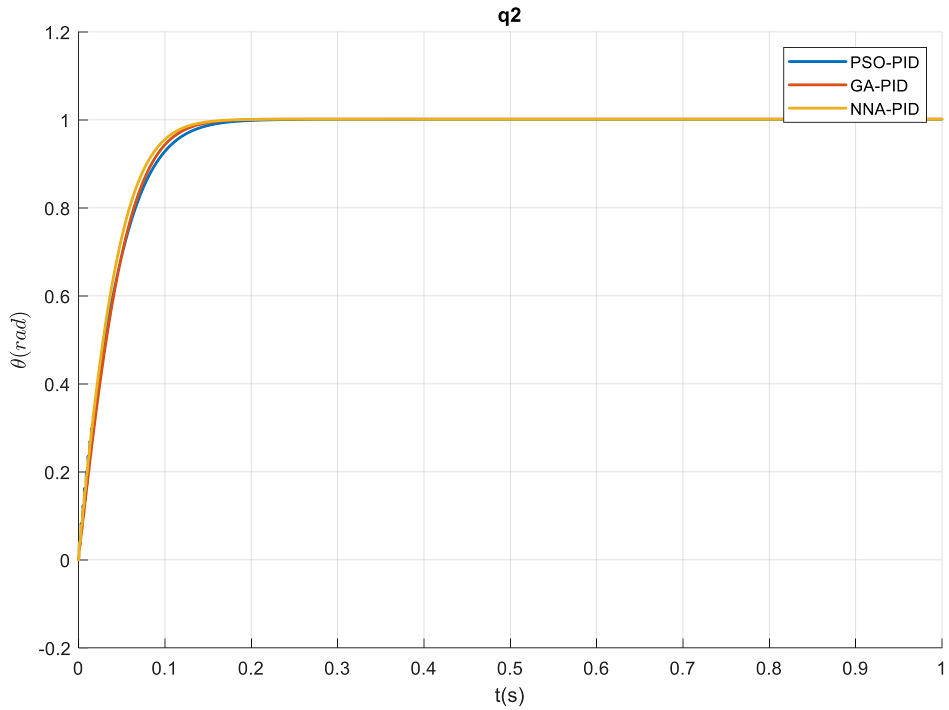
Figure 1 Link2 step response





Comparison

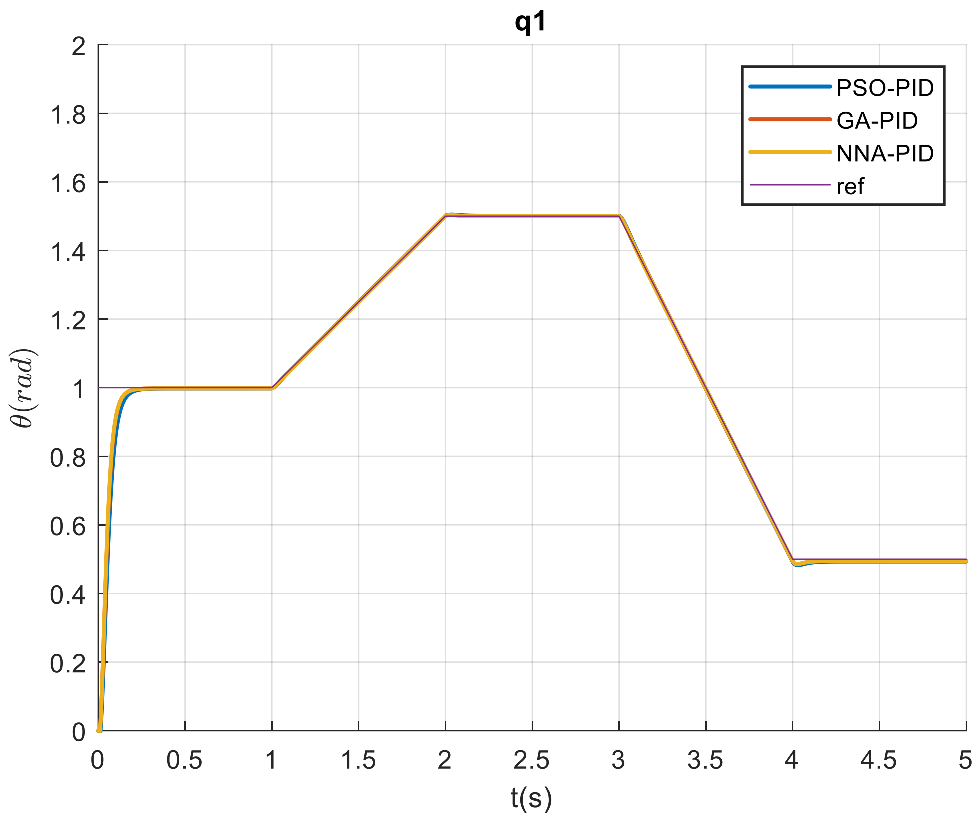


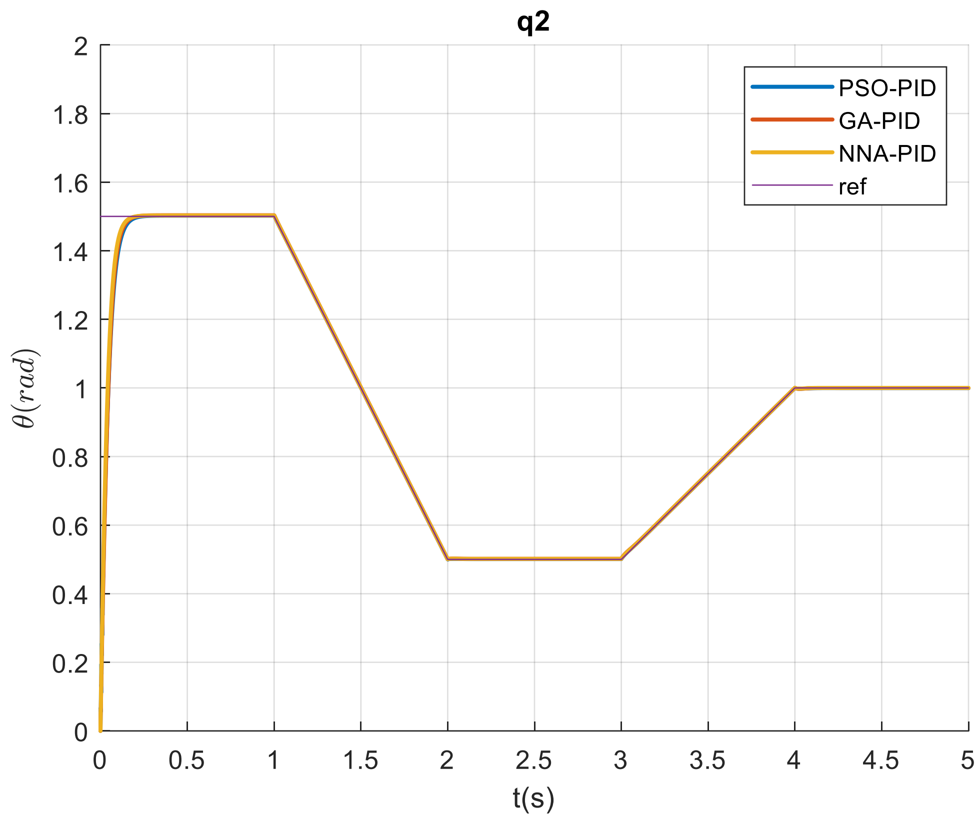


		GA-PID	PSO-PID	NNA-PID
Controller Parameters	Link1	$K_P = 494.94$	$K_P = 415.52$	$K_P = 497.45$
		$K_I = 0.995$	$K_I = 0.297$	$K_I = 0.0057$
		$K_D = 19.25$	$K_D = 18.02$	$K_D = 18.99$
	Link2	$K_P = 212.34$	$K_P = 112.80$	$K_P = 379.07$
		$K_I = 0.017$	$K_I = 0.187$	$K_I = 0.0307$
		$K_D = 6.58$	$K_D = 4.34$	$K_D = 10$
J		0.041568	0.050615	0.038131

Comparison

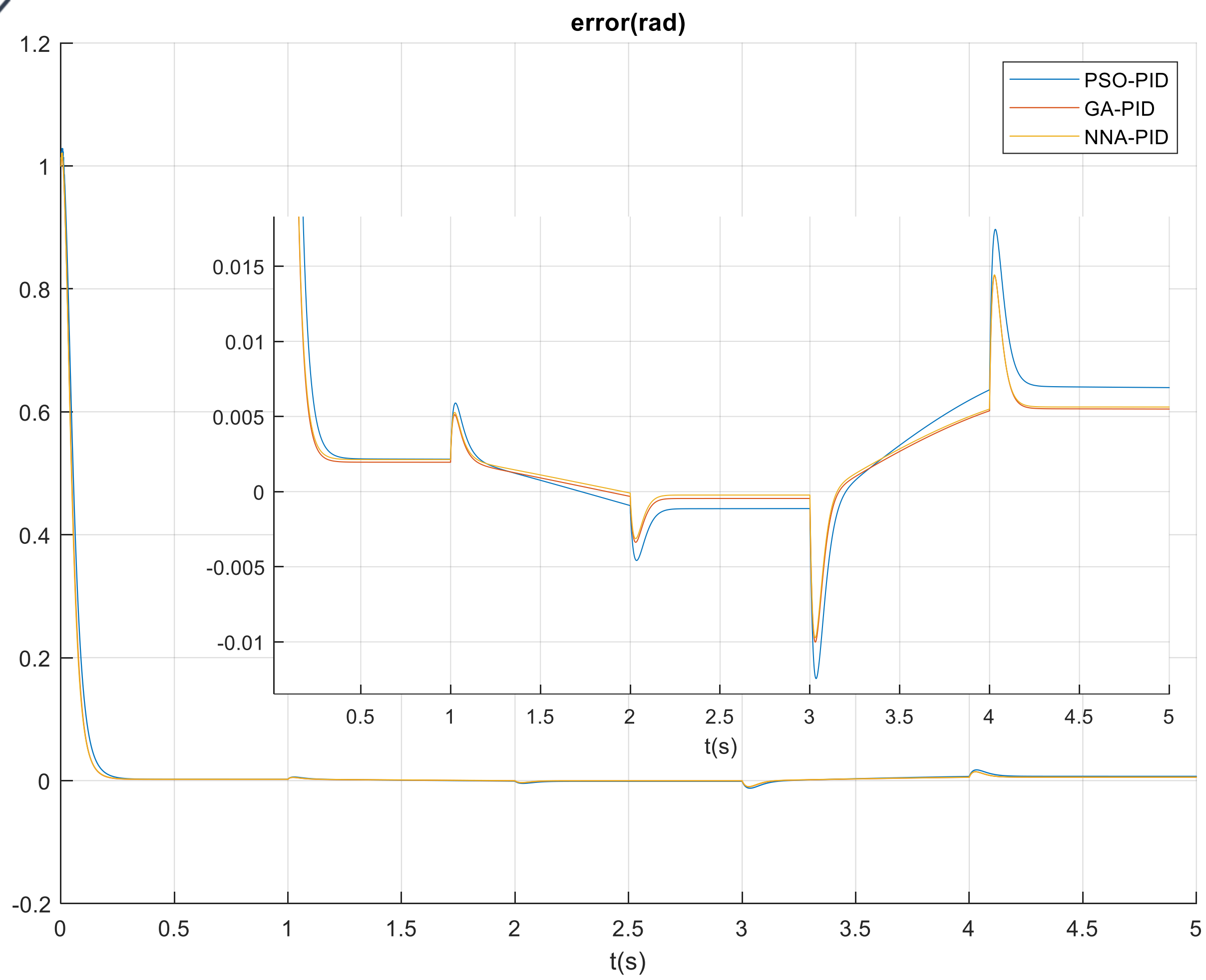
- VARIOUS TRAJECTORIES

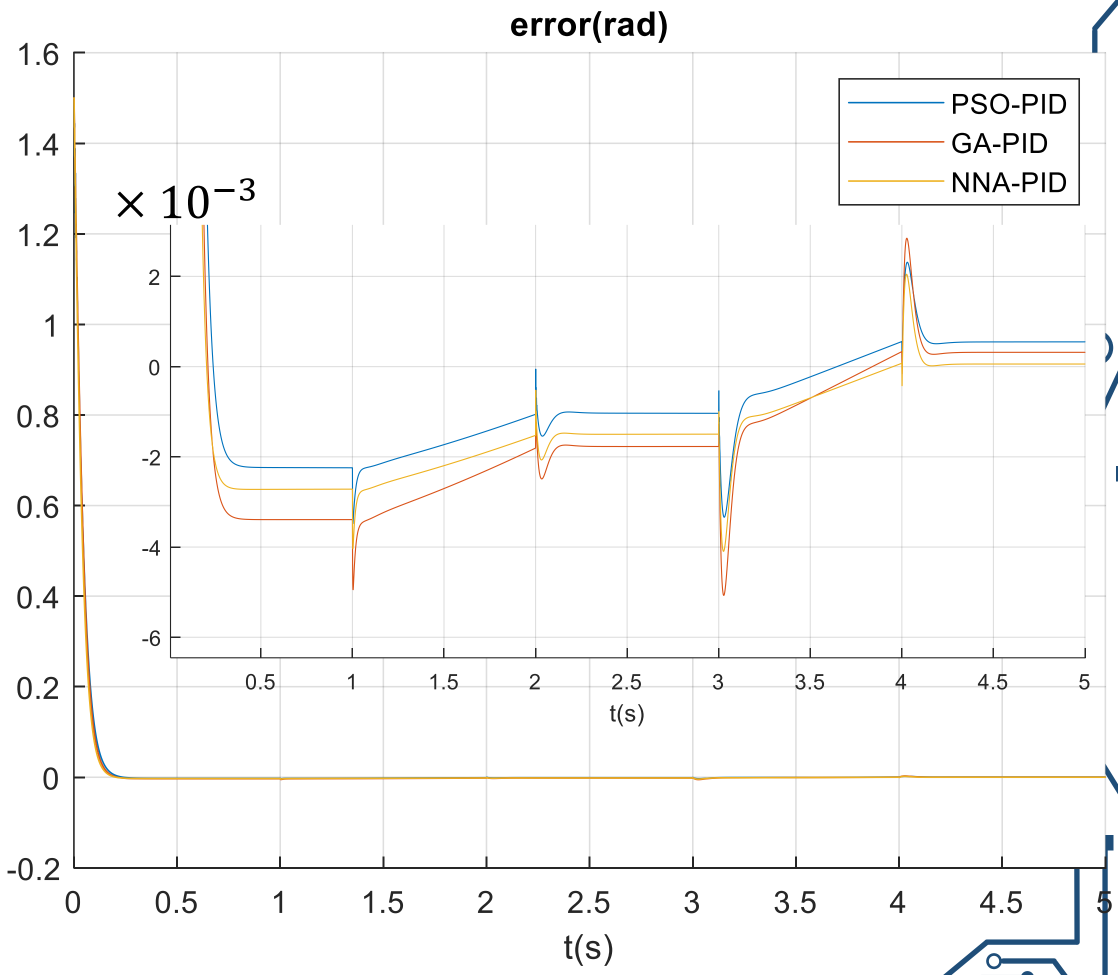




ICALAB
智慧型控制及應用實驗室

Comparison





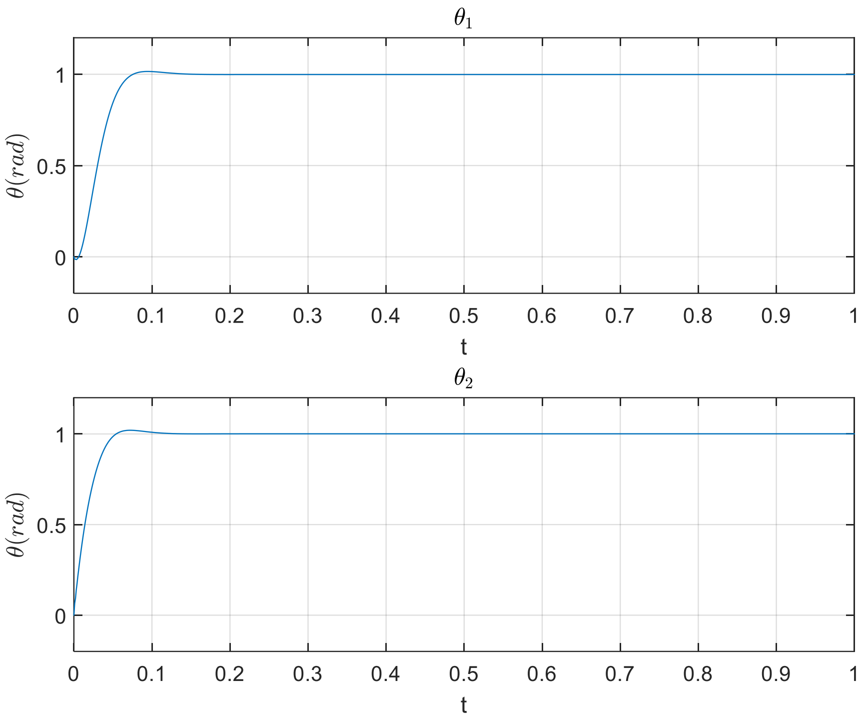
ICALAB
智慧型控制及應用實驗室

Performance index

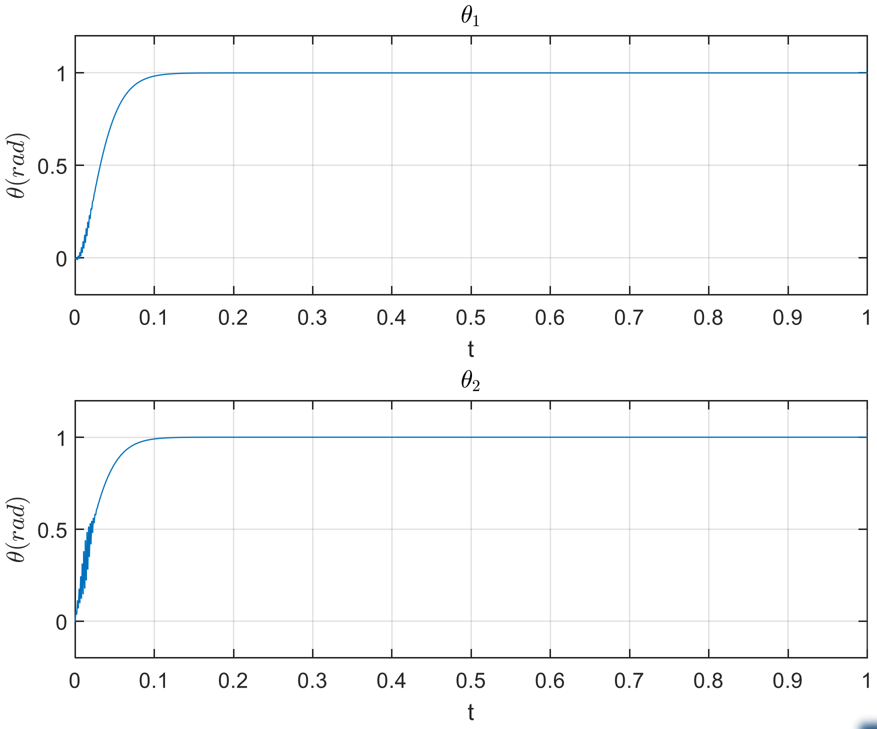
- **Performance index**
  Goal: decrease settling time 't_s' and maximum overshoot 'M_p'

$$J = \sum_{i=1}^{2}\left(1-e^{-\psi}\right)\left(M_{P,i}+E_{ss,i}\right)+e^{-\psi}\left(t_{s,i}-t_{r,i}\right)$$

$M_{P,i}$	Overshoot
$E_{ss,i}$	Steady-state error
$t_{s,i}$	settling time
$t_{r,i}$	rise time
$\psi(=0.7)$	weight constant



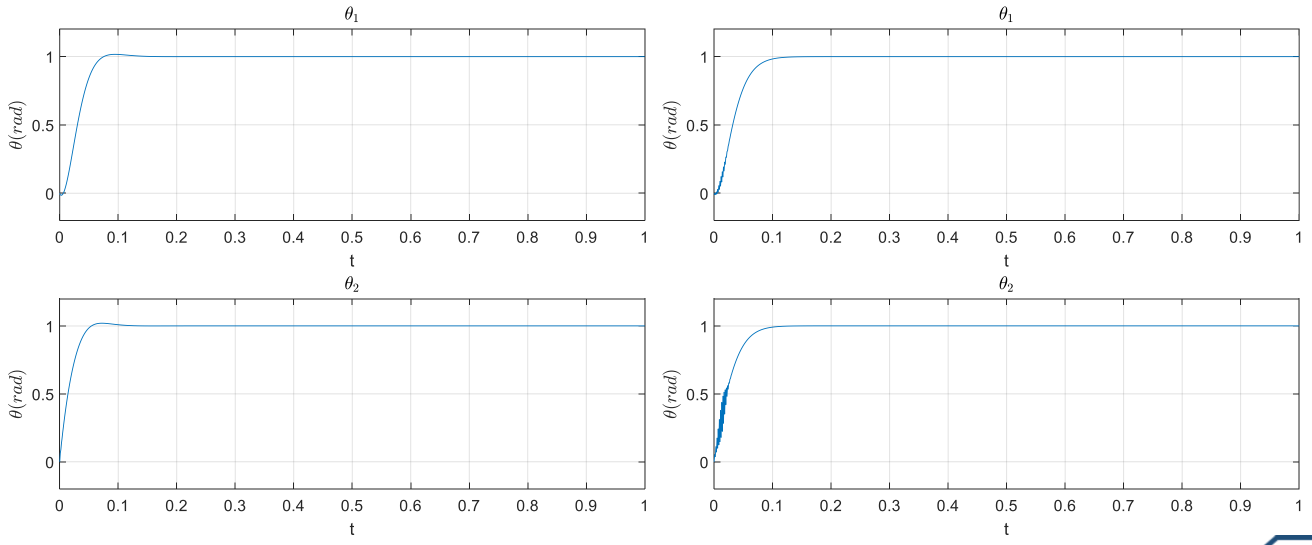
ψ =0 loss: 0.03989



ψ =1 loss: 0.028892

Performance index

	ψ =0		ψ = 1	
	Link1	Link2	Link1	Link2
RiseTime	**0.0409**	**0.0328**	0.0550	0.0541
SettlingTime	**0.0662**	**0.0462**	0.0962	0.0865
Overshoot	2.035%	1.939%	**0.021%**	**0.052%**
Steady-state error	0.0013	**0.0006**	**0.0012**	0.0009
Total loss	0.03989		**0.0289**	



ICALAB 智慧型控制及應用實驗室

Performance index

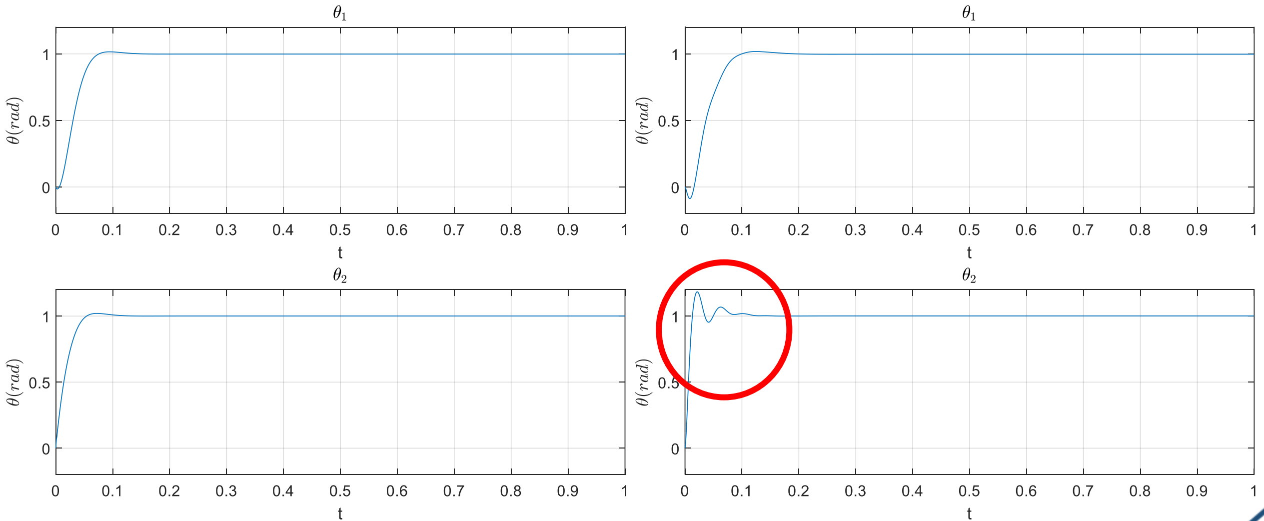
- **Performance index** $(let\ \psi = 0)$

Goal: decrease settling time 't_s' and maximum overshoot 'M_p' $\quad J = \sum_{i=1}^{2}(t_{s,i} \mp t_{r,i})$

	$(t_{s,i} - t_{r,i})$		$(t_{s,i} + t_{r,i})$	
	Link1	Link2	Link1	Link2
RiseTime	**0.0409**	0.0328	0.0518	**0.0091**
SettlingTime	**0.0662**	**0.0462**	0.0883	0.0779
Overshoot	2.035%	**1.939%**	**1.995%**	19.751%
Steady-state error	**0.0013**	**0.0006**	0.0025	0.0013

Why?

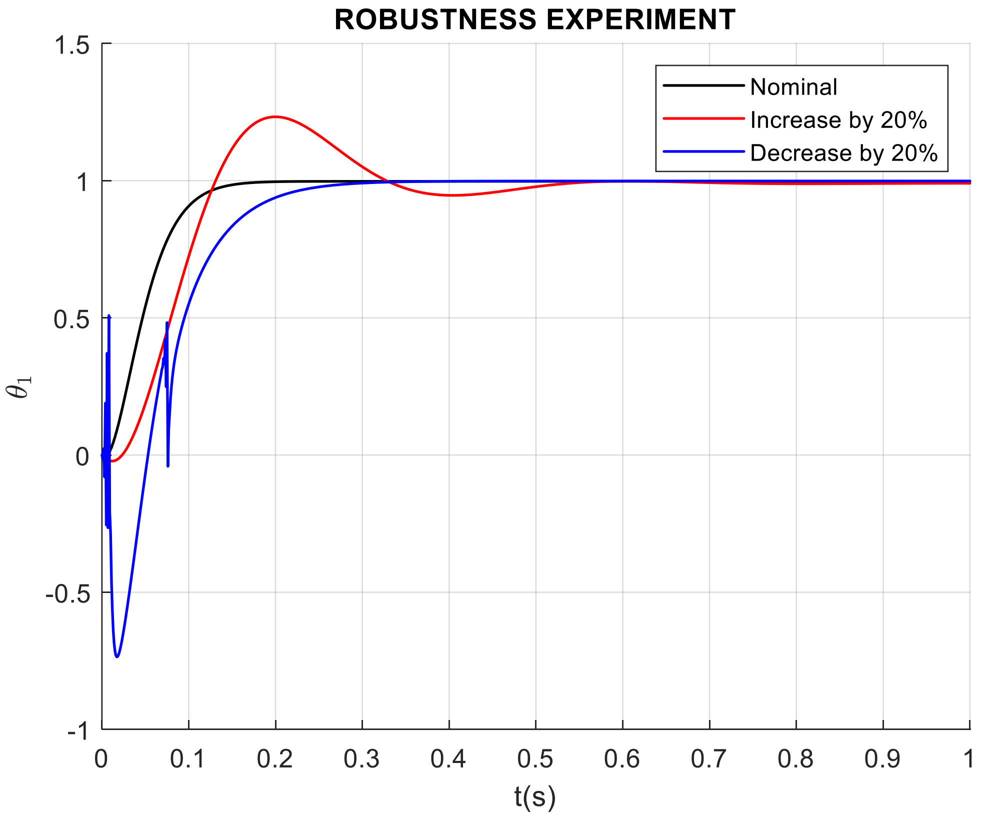
Why?

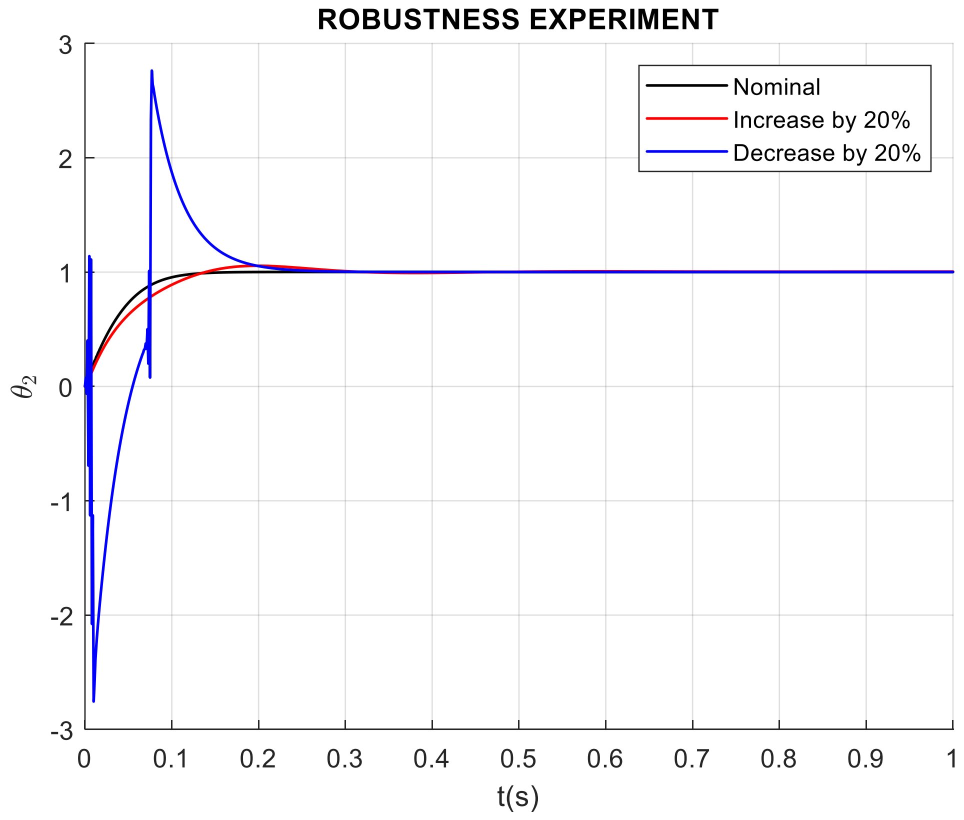


ICALAB 智慧型控制及應用實驗室

Robustness Experiment

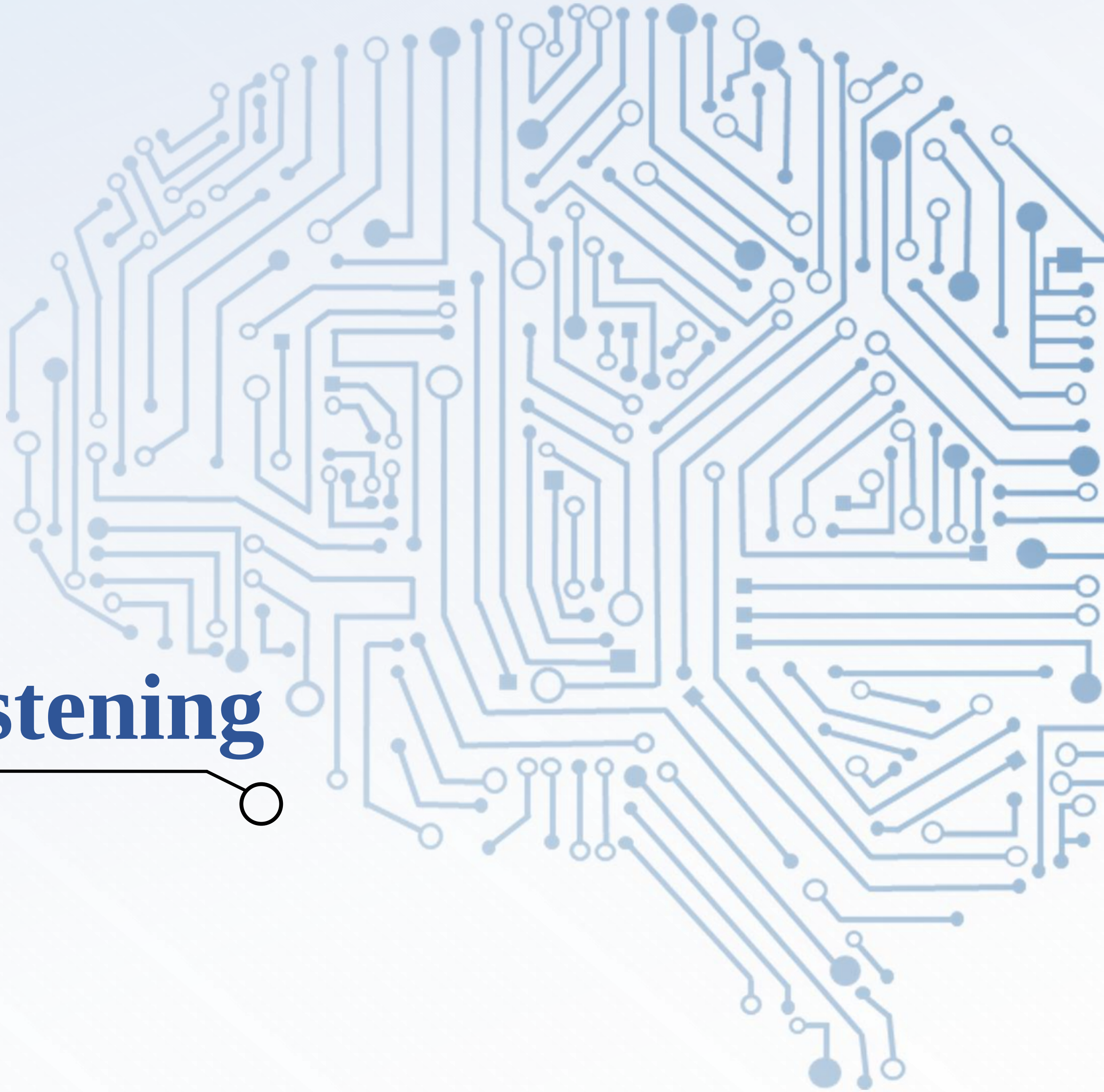
- Making uncertainty in masses and lengths of the robotic arms by ±20%





智慧型控制及應用實驗室

ICALab



Thanks for Your Listening