E4602 HW8 Chengxi Dong UNI: cd2903

in HW7, we know that

$$D\begin{pmatrix}\ddot{q}_1\\ \ddot{q}_2\end{pmatrix} + C\begin{pmatrix}\dot{q}_1\\ \dot{q}_2\end{pmatrix} + \begin{pmatrix}\Phi_1\\ \Phi_2\end{pmatrix} = \begin{pmatrix}u_1\\ u_2\end{pmatrix}$$

let $\begin{pmatrix}u_1\\ u_2\end{pmatrix} = D(q)\begin{pmatrix}v_1\\ v_2\end{pmatrix} + C\begin{pmatrix}\dot{q}_1\\ \dot{q}_2\end{pmatrix} + \begin{pmatrix}\Phi_1\\ \Phi_2\end{pmatrix}$

then $\begin{pmatrix}\ddot{q}_1\\ \ddot{q}_2\end{pmatrix} = \begin{pmatrix}v_1\\ v_2\end{pmatrix}$ (computed torque method)

where $D(q) = \begin{pmatrix} I_1 + I_2 + m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos q_2) & I_2 + m_2 l_{c2}^2 + m_2 l_1 l_{c2}\cos q_2 \\ I_2 + m_2 l_{c2}^2 + m_2 l_1 l_{c2}\cos q_2 & I_2 + m_2 l_{c2}^2 \end{pmatrix}$

$$C = \begin{pmatrix} -m_2 l_1 l_{c2}\sin q_2\, \dot{q}_2 \ , & -m_2 l_1 l_{c2}\sin q_2 (\dot{q}_1 + \dot{q}_2) \\ m_2 l_1 l_{c2}\sin q_2\, \dot{q}_1 \ , & 0 \end{pmatrix}$$

$$\Phi = \begin{pmatrix} (m_1 l_{c1} + m_2 l_1) g\cos q_1 + m_2 l_{c2} g\cos(q_1+q_2) \\ m_2 l_{c2}\cos(q_1+q_2) \end{pmatrix}$$

trajectory :

$$q_d = \begin{pmatrix} a_1 + a_2 t + a_3 t^2 + a_4 t^3 \\ \\ b_1 + b_2 t + b_3 t^2 + b_4 t^3 \end{pmatrix}$$

in Matlab , Label starts from '1' , not '0'

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & t_f & t_f^2 & t_f^3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2t_f & 3t_f^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & t_f & t_f^2 & t_f^3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2t_f & 3t_f^2 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{\pi}{3} \\ 0 \\ 0 \\ 0 \\ \frac{\pi}{4} \\ 0 \end{pmatrix}$$

solving these equations , then we have the trajectory .

$$v = \begin{pmatrix} v_1 \\ \\ v_2 \end{pmatrix} = \ddot{q}_d + k_d(\dot{q}_d - \dot{q}) + k_p(q_d - q)$$

$$u = Dv + C\dot{q} + \Phi .$$ $\Leftarrow$ compute torque .

in part 3. ① Sensor error ② paras error

I assume the modeling error is normal distributed ($\hat{q}$, $\hat{\dot{q}}$ measured by sensor)

In the test example , $\hat{q} = N(q, (0.05q)^2) = q(1 + \text{randn}/20)$

$\Uparrow$

same for $\hat{\dot{q}}$.

so that $e = q \cdot \text{randn}/20$

Normally distributed .

then. $u = D(\hat{q})v + C(\hat{q}, \hat{\dot{q}})\hat{\dot{q}} + \Phi(\hat{q})$

$v = \ddot{q}_d + k_d(\dot{q}_d - \hat{\dot{q}}) + k_p(q_d - \hat{q})$

$D\ddot{q} + C(q, \dot{q})\dot{q} + \Phi(q) = u(\hat{q}, q_d, \hat{\dot{q}}, \dot{q}_d)$ (physical equation remains) part 3. II paras error.

just follow the given code.

$\hat{M} = 1.5M$, $\hat{I} = 0.5I$ $\hat{l_c} = 0.7l$.

then.

$v = \ddot{q}_d + k_d(\dot{q}_d - \dot{q}) + k_p(q_d - q)$

$u = \hat{D}v + \hat{C}\dot{q} + \hat{\Phi}$

then. $D\ddot{q} + C\dot{q} + \Phi = u$

part 4.

We can see the system cannot track well with a large error.

But tuning is still useful for control with an error.

For example, first I simulate with $k_p = 100$, $k_d = 20$, $\hat{q} = q(1 + 0.05\,\text{radn})$

If we increase the gain to $k_p = 120$, $k_d = 25$. the tracking becomes better.

However, too big gains may cause unstability. We must find a set of acceptable gains for a particular system.

What I observe: 1. we can't eliminate the error, since ① Discrete ODE ② No Integral Controller

2. When k_p is too large. the system excessive oscillates.

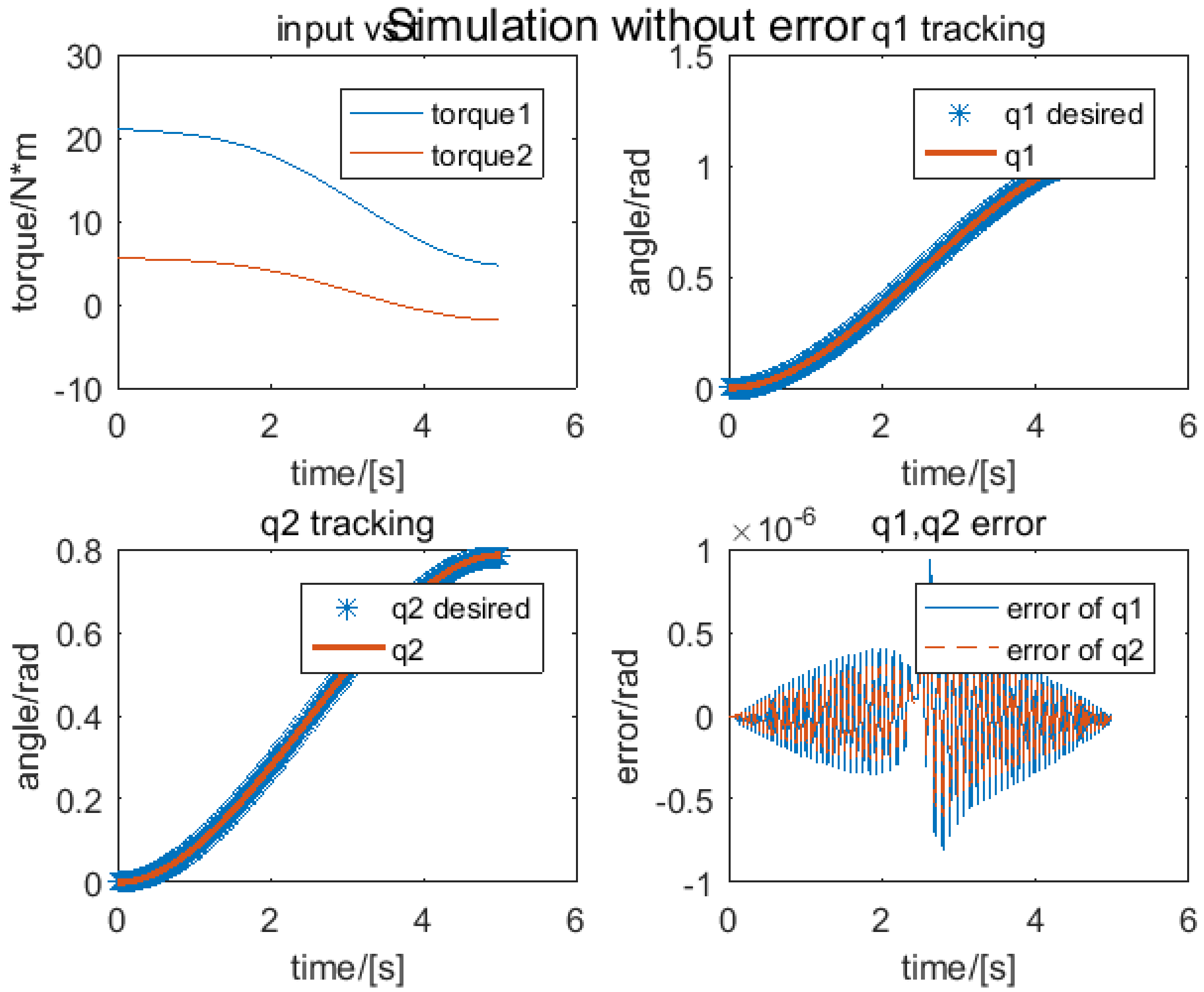
3. k_d can help reduce error. (ealier - reaction)

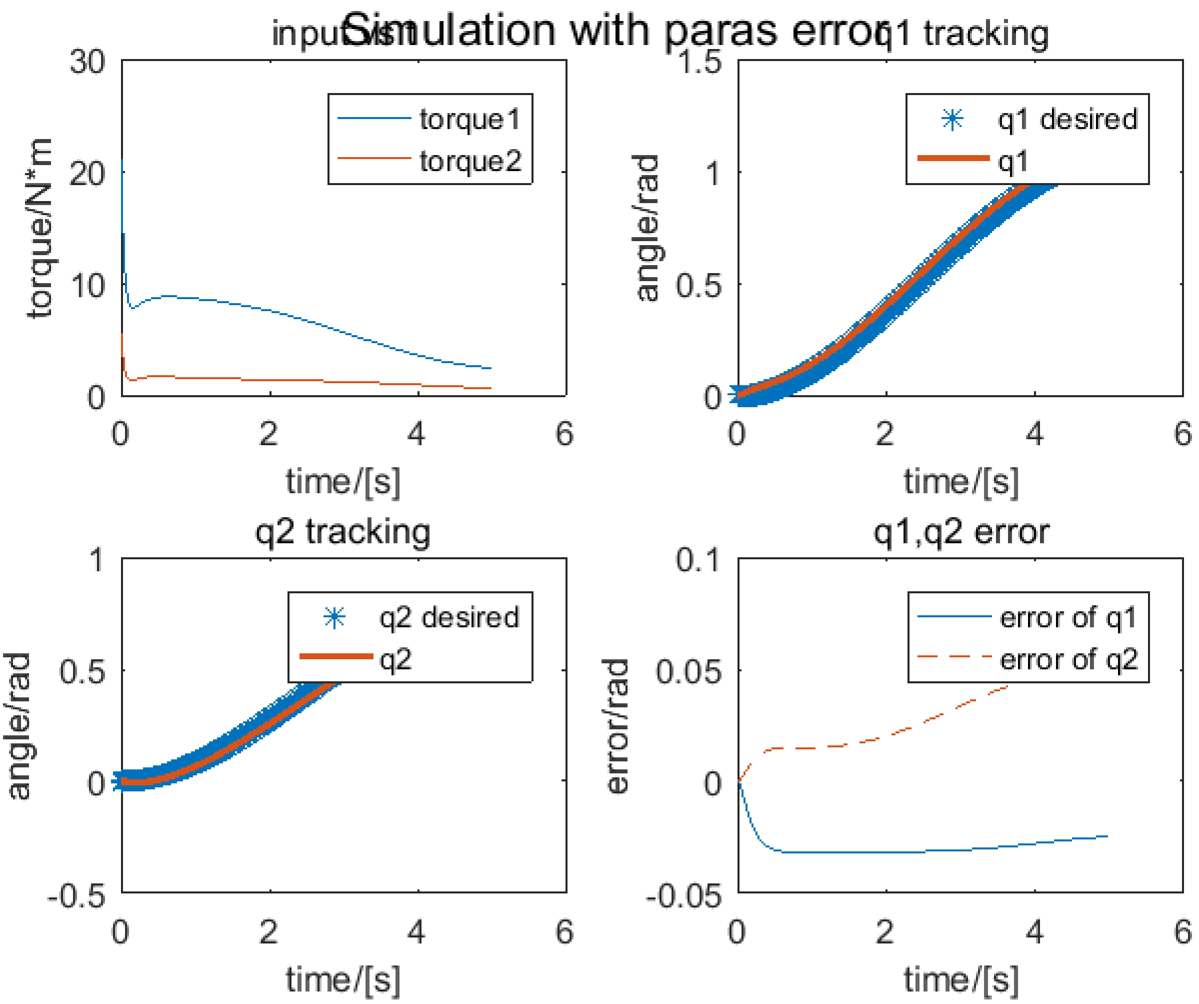
4. increase standard error, the tracking error also increase.

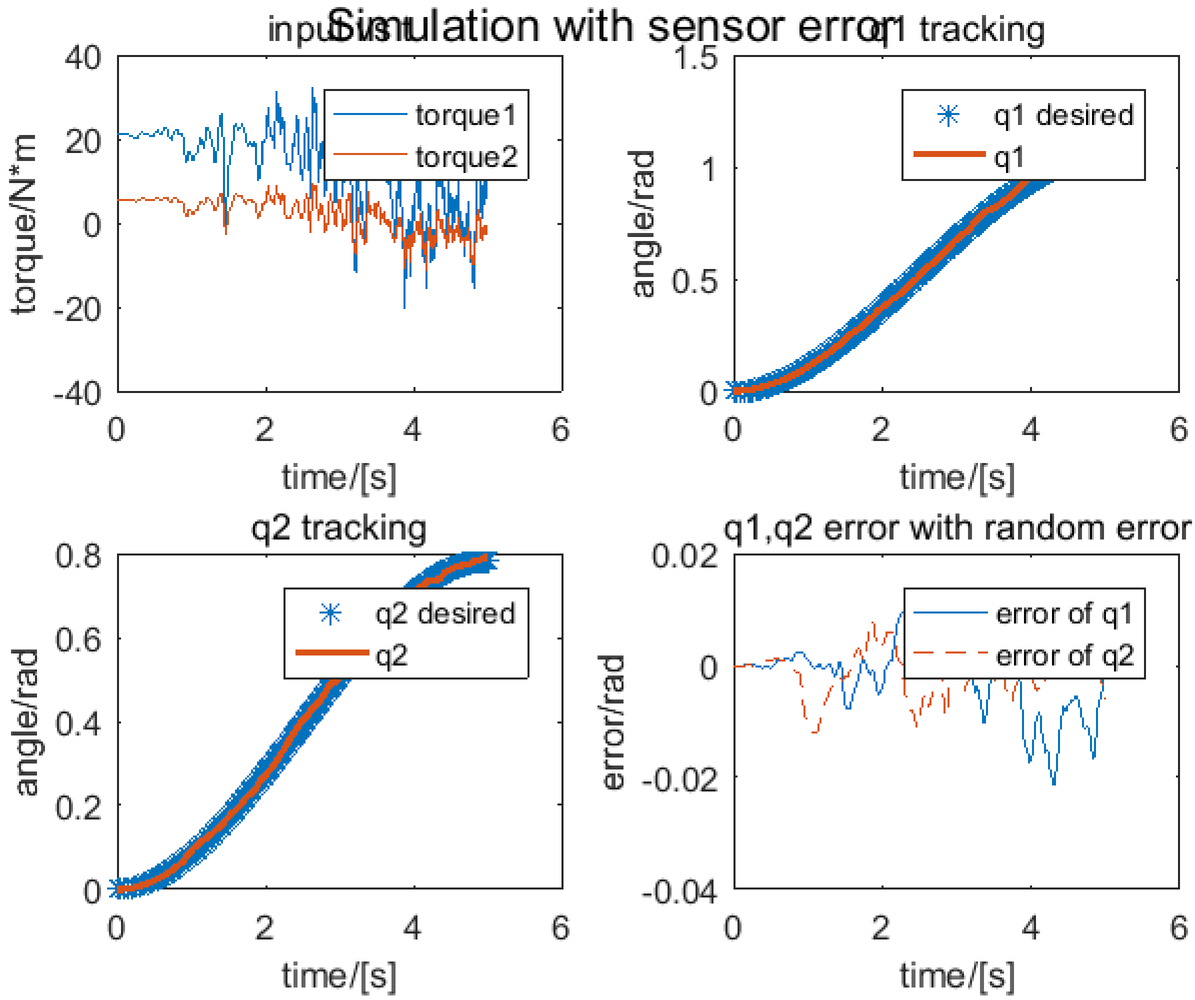
Below are the figures for ① no error ② paras error ③ sensor error!

② $\hat{M} = 1.5M$, $\hat{I} = 0.5I$, $\hat{l_c} = 0.7l$

③ $\hat{q} = q(1 + 0.05\,\text{radn})$


Simulation without error
input vs t
q1 tracking
q2 tracking
q1,q2 error
torque/N*m
angle/rad
error/rad
time/[s]
torque1
torque2
q1 desired
q1
q2 desired
q2
error of q1
error of q2
×10^-6


Simulation with paras error
input vs t
q1 tracking
q2 tracking
q1,q2 error
torque/N*m
angle/rad
error/rad
time/[s]
torque1
torque2
q1 desired
q1
q2 desired
q2
error of q1
error of q2



Simulation with sensor error
input
torque/N*m
torque1
torque2
time/[s]
q1 tracking
angle/rad
q1 desired
q1
time/[s]
q2 tracking
angle/rad
q2 desired
q2
time/[s]
q1,q2 error with random error
error/rad
error of q1
error of q2
time/[s]