POLITECNICO
MILANO 1863

EXERCISE 5 – NEIGHBORING OPTIMAL CONTROL VIA LINEAR QUADRATIC FEEDBACK

EMANUELE Riva

Course: Mechatronic systems and laboratory

Department of Mechanical Engineering, Politecnico di Milano, Italy

PREVIOUSLY ON M.S.&Lab

AGENDA

- THEORY REVIEW
- EXAMPLE 1
- HANDS-ON: Matlab
- EXAMPLE 2

The optimal control history that minimizes the functional:

$$J = \phi\left(x(t_f)\right) + \int_{t_0}^{t_f} L\big(x(t), u(t)\big)\, dt$$

subjected to the dynamic system $\dot{x} = f\big(x(t), u(t)\big)$ is provided if the three E.L. equations are satisfied, i.e. the first variation of the cost J for control variations is zero.

Let's consider the resulting nominal optimal trajectory:

$$\{x^*(t)\,, u^*(t)\}\;\; for\; t \in \left[t_0\; t_f\right]$$

small linear perturbation:

$$\{x(t)\,, u(t)\} = \{x^*(t) + \Delta x(t)\,, u^*(t) + \Delta u(t)\}\;\; for\; t \in \left[t_0\; t_f\right]$$

Note that we have performed a change of notation w.r.t. the previous exercise:

$$\big(\Delta x(t) = \alpha\xi(t), \Delta u(t) = \alpha\omega(t)\big)$$

Dynamic equation describing the oscillations of the states relative to the reference trajectory (due to control perturbation):

$$\{x(t)\,,u(t)\} = \{x^*(t) + \Delta x(t)\,,u^*(t) + \Delta u(t)\}\ \ for\ t \in [t_0\ t_f]$$

$$\dot{x}(t) = \dot{x}^*(t) + \Delta\dot{x}(t) \cong f\big(x^*(t), u^*(t)\big) + A(t)\Delta x(t) + B(t)\Delta u(t)$$

Reference trajectory: $\dot{x}^*(t) = f\big(x^*(t), u^*(t)\big);\ \ x^*(t_0) = x_0^*$

Small oscillations: $\Delta\dot{x}(t) = A(t)\Delta x(t) + B(t)\Delta u(t);\ \Delta x(t_0) = \Delta x_0$

$$A(t) = \frac{\partial f\big(x^*(t), u^*(t)\big)}{\partial x} \qquad \mathrm{B}(t) = \frac{\partial f\big(x^*(t), u^*(t)\big)}{\partial u}$$

the cost function is expanded around the optimal trajectory in $[t_0\ t_f]$:

$$J \approx J^*\big(x^*(t), u^*(t)\big) + \Delta J\big(\Delta x(t), \Delta u(t)\big) + \Delta^2 J\big(\Delta x(t), \Delta u(t)\big)$$

Where $\Delta J = \frac{\partial J}{\partial x}\Delta x + \frac{\partial J}{\partial u}\Delta u$ *and* $\Delta^2 J = \Delta x^T \frac{\partial^2 J}{\partial x^2}\Delta x + \Delta u^T \frac{\partial^2 J}{\partial x \partial u}\Delta x + \Delta x^T \frac{\partial^2 J}{\partial u \partial x}\Delta u + \Delta u^T \frac{\partial^2 J}{\partial u^2}\Delta u$

Since $\{x^*(t)\ , u^*(t)\}$ *is optimal, J is insensitive to small perturbation (*$\Delta J\big(\Delta x(t), \Delta u(t)\big) = 0$*). AND THIS IS VALID FOR ANY SMALL PERTURBATION OF THE CONTROL (*$\Delta u(t)$*)!*

The second variation is the remaining cost:

$$\Delta^2 J\big(\Delta x(t), \Delta u(t)\big) = \frac{1}{2}\left[\Delta x^T(t_f)\frac{\partial^2 \phi(t_f)}{\partial x^2}\Delta x(t_f) + \int_{t_0}^{t_f} \begin{bmatrix}\Delta x^T(t)\\ \Delta u^T(t)\end{bmatrix}^T \begin{bmatrix} L_{xx} & L_{xu} \\ L_{ux} & L_{uu}\end{bmatrix}\begin{bmatrix}\Delta x(t)\\ \Delta u(t)\end{bmatrix} dt\right]$$

$$Q = L_{xx} \quad M = L_{xu} \quad M^T = L_{ux} \quad R = L_{uu} \quad P(t_f) = \frac{\partial^2 \phi(t_f)}{\partial x^2}$$

We can conclude that:

$$J \approx J^*\big(x^*(t), u^*(t)\big) + \Delta^2 J\big(\Delta x(t), \Delta u(t)\big)$$

Where at this point $\{x^*(t)\ , u^*(t)\}$ *is known. The second-order vartiation is non-zero unless one looks for a new objective, in the attempt to minimize the oscillations around the reference (optimal) trajectory:*

The objective is now: $\min\limits_{\Delta u} \Delta^2 J\left(\Delta x(t), \Delta u(t)\right)$

subjected to: $\Delta \dot{x}(t) = A(t)\Delta x(t) + B(t)\Delta u(t)$

With:

$$\Delta^2 J\big(\Delta x(t), \Delta u(t)\big) = \frac{1}{2}\left[\Delta x^T(t_f)\frac{\partial^2 \varphi(t_f)}{\partial x^2}\Delta x(t_f) + \int_{t_0}^{t_f} \begin{bmatrix}\Delta x^T(t)\\ \Delta u^T(t)\end{bmatrix}^T \begin{bmatrix}L_{xx} & L_{xu}\\ L_{ux} & L_{uu}\end{bmatrix}\begin{bmatrix}\Delta x(t)\\ \Delta u(t)\end{bmatrix} dt\right]$$

$$Q = L_{xx} \quad M = L_{xu} \quad M^T = L_{ux} \quad R = L_{uu} \quad P(t_f) = \frac{\partial^2 \phi(t_f)}{\partial x^2}$$

Hamiltonian: $H\big(\Delta x(t), \Delta u(t), \Delta\lambda(t)\big) = L\big(\Delta x(t), \Delta u(t)\big) + \Delta\lambda^T(t) f(\Delta x(t), \Delta u(t))$

Upon the application of E.L. Equations, where the new variables are $\Delta x(t), \Delta u(t)$

$$(1) \quad \Delta\lambda = P(t_f)\Delta x(t_f)$$

$$(2) \quad \Delta\dot{\lambda} = -Q\Delta x - M\Delta u - A^T\Delta\lambda$$

$$(3) \quad \frac{\partial H}{\partial \Delta u} = M^T\Delta x + R\Delta u + B^T\Delta\lambda = 0 \longrightarrow \Delta u^* = -R^{-1}(M^T\Delta x^* + B^T\Delta\lambda)$$

By assuming $\Delta\lambda = P\Delta x$ *(i.e. we take the same form we have in (1))*

$$\longrightarrow \quad \Delta u^* = -R^{-1}(M^T + B^T P)\Delta x = K\Delta x$$

Feedback control action Δu
with control-gain matrix K

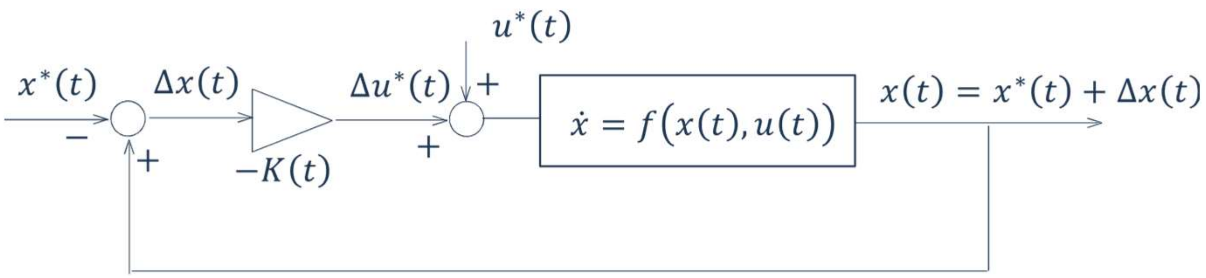
after some mathematical steps we get to the Matrix, Nonlinear Riccati differential equation (DRE).

$$\dot{P}(t) = -(A^T(t)P(t) + P(t)A(t) - P(t)B(t)R^{-1}B^T(t)P(t) + Q)$$

$$P(t_f) = \phi_{xx}$$

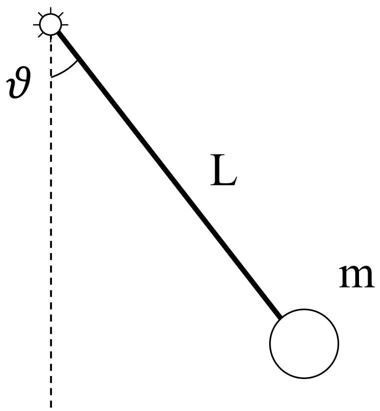
The solution $P(t)$ of the DRE provides the optimal feedback gain matrix (for $M = 0$):

$$K(t) = -R^{-1}B(t)P(t) \qquad \Delta u^*(t) = -K(t)\Delta x(t)$$



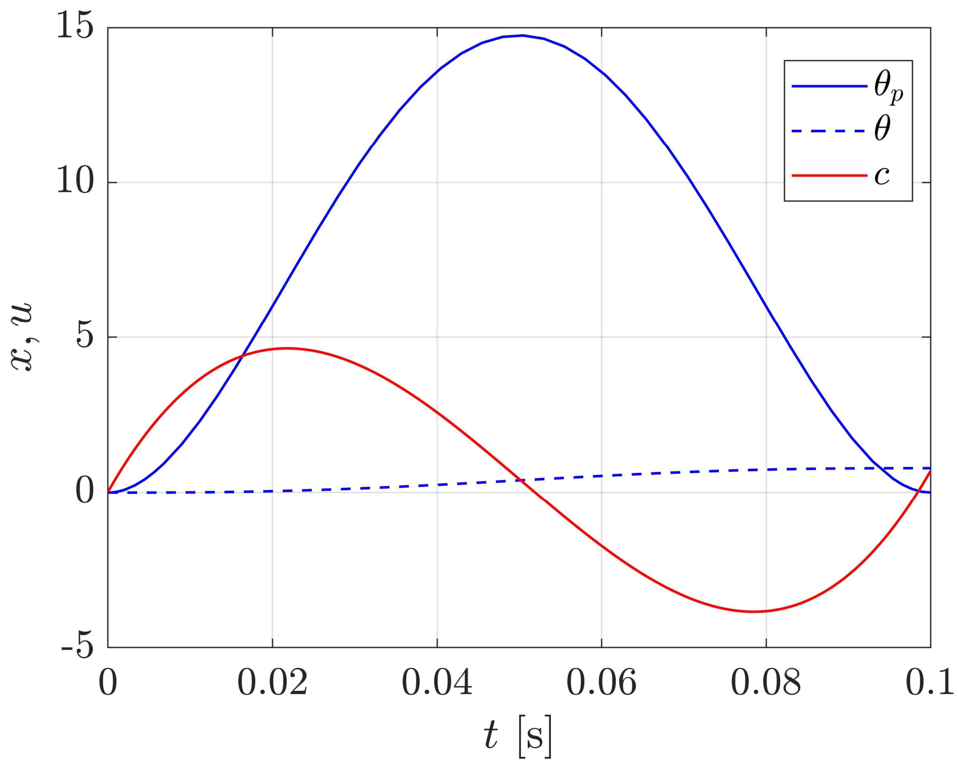
Exercise 1

<u>Goal</u>: *Find the optimal neighboring feedback control to the nonlinear optimal trajectory Optimal_Trajectory.mat studied in the previous exercise.*



$$\begin{cases} \dot{x}_2 = -2\zeta\omega_0 x_2 - \omega_0^2 \sin(x_1) + \dfrac{c(t)}{mL^2} \\ \dot{x}_1 = x_2 \end{cases}$$

$$\boldsymbol{x} = [x_2, x_1]^T = \left[\dot{\theta}, \theta\right]^T \qquad \boldsymbol{x}_i = [0, 0]^T$$



Exercise 1

Trajectory: $\{x^*(t)\,, u^*(t)\}$.

Linearized dynamics: $\Delta\dot{x}(t) = A(t)\Delta x(t) + B(t)\Delta u^*(t)$ $\qquad$ $\Delta u^* = -R^{-1}B^T P \Delta x$

The state and control matrices $A(t)$ *,*$B(t)$ *vary along the trajectory, therefore the differential equation has time varying parameters:*

$$\dot{P}(t) = -(A^T(t)P(t) + P(t)A(t) - P(t)B(t)R^{-1}B^T(t)P(t) + Q)$$

With:

$$A = \frac{\partial f}{\partial \boldsymbol{x}} = \begin{bmatrix} -2\zeta\omega_0 & -\omega_0^2\cos(x_1) \\ 1 & 0 \end{bmatrix} \qquad B = \frac{\partial f}{\partial u} = \begin{bmatrix} \dfrac{1}{mL^2} \\ 0 \end{bmatrix}$$

Exercise 1

The DRE is integrated as follows:

```
% initial and final time
t0 = T(1);
tf = T(end);

% boundary conditions
p0 = P(1:end)';

% Integration of the matrix Riccati equation
[Tp,PP] = ode23(@(t,p) DRE(t,p,Q,R,X,Tx,c,Tc), flip(T), p0, options);

% From backward to forward dynamics (the solution is stored in reversed order)
PP = flipud(PP);
Tp = flipud(Tp);
```

Exercise 1

Where the DRE function is:

```
function [OutVect] = DRE(t,p,Q,R,X,Tx,c,Tc)

% evaluation of the state at time @t
Nstates = size(X,2);
xk = zeros(1,Nstates);
for ii = 1:Nstates
    xk(:,ii) = interp1(Tx,X(:,ii),t);
end

% evaluation of the control action at time @t
Nc = size(c,2);
ck = zeros(1,Nc);
for ii = 1:Nc
    ck(:,ii) = interp1(Tc,c(:,ii),t);
end

A = fx(xk);
B = fc();
```

- *Interpolation of the state and control at integration time @t*

- *Evaluation of the state and control matrix given the state @x(t) and control @c(t)*

$$A = \frac{\partial f}{\partial x} \qquad B = \frac{\partial f}{\partial x}$$

Exercise 1

Where the DRE function is:

```
% transformation vector -> matrix
P = zeros(Nstates,Nstates);
P(1:end) = p(1:end);

% DRE
Out = -(A.'*P + P*A - P*B*R^-1*B.'*P + Q);


% transformation matrix -> vector
OutVect = Out(1:end)';
```

- *Transformation from vector to matrix of P.*
- *Differential Riccati Equation (DRE)*
- *Transformation from matrix to vector*

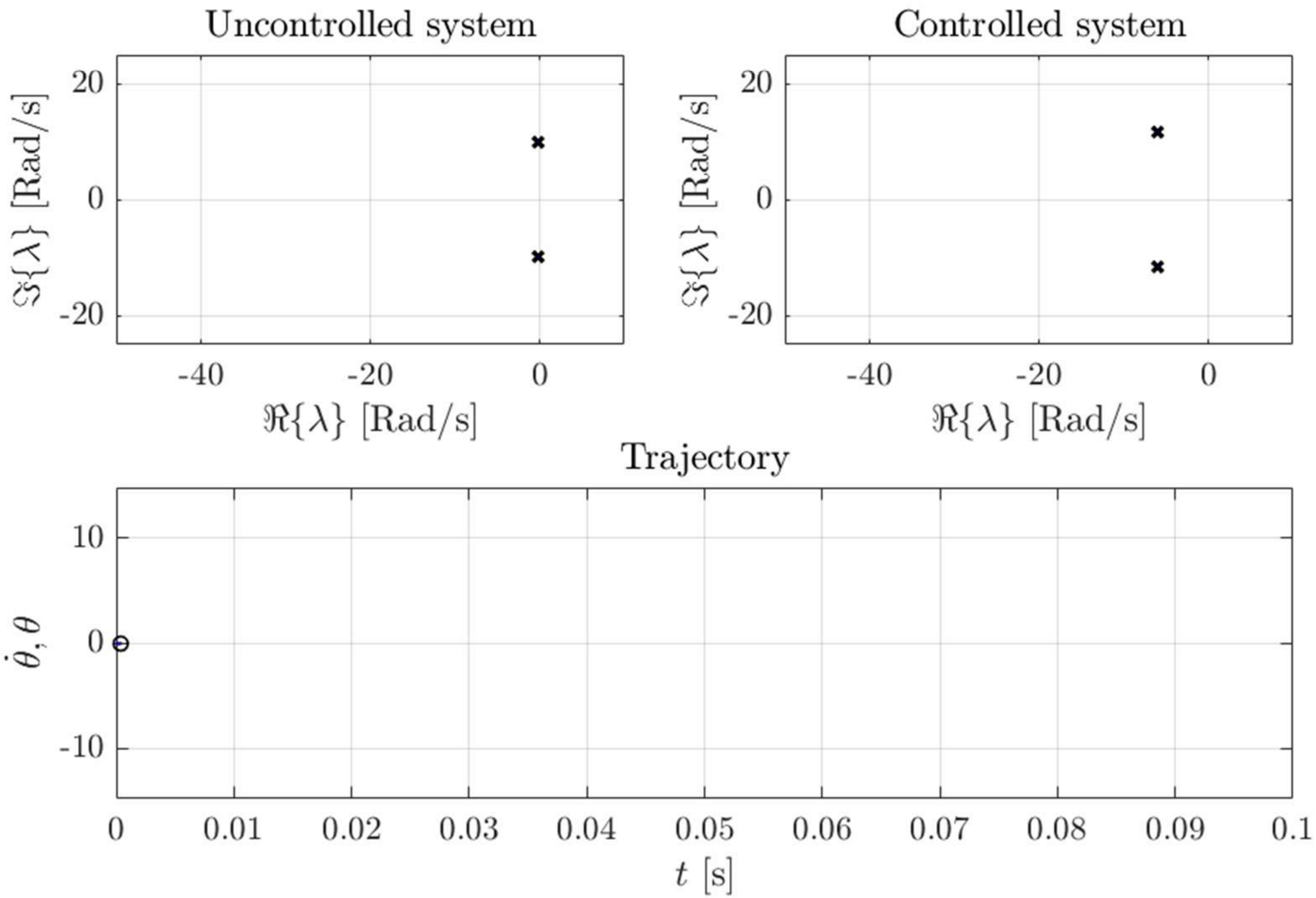
Note: it is necessary to transform the matrix differential equation into a system of differential equations. The gain matrix is then evaluated from P in matrix form:

$$K(t) = -R^{-1}B(t)P(t)$$

And given the control gain history it is possible to investigate the stability properties along the trajectory:

$$A_C(t) = A(t) - B(t)K(t)$$

Exercise 1

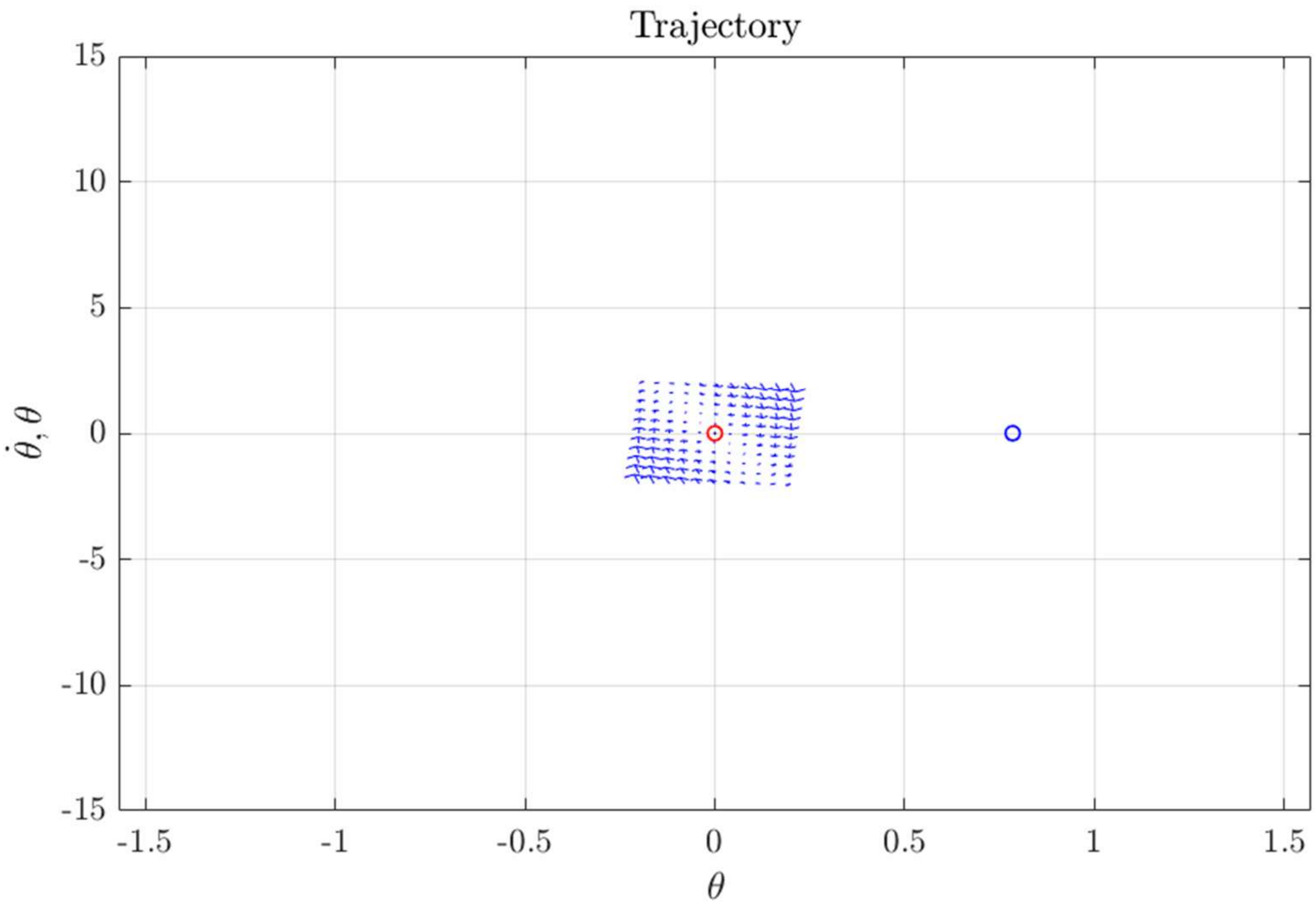


$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$R = 1$$

$$P = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$$

Exercise 1

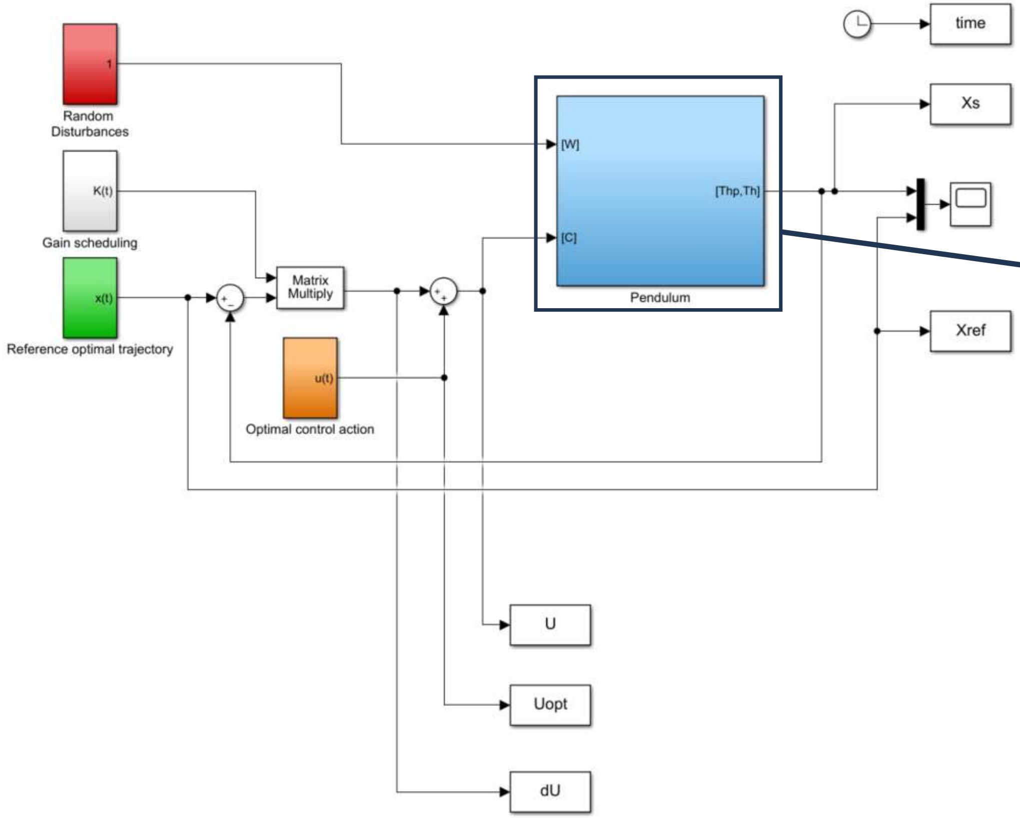


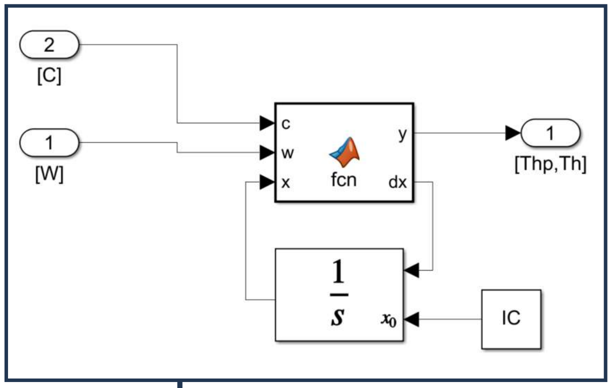
$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$R = 1$$

$$P = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$$

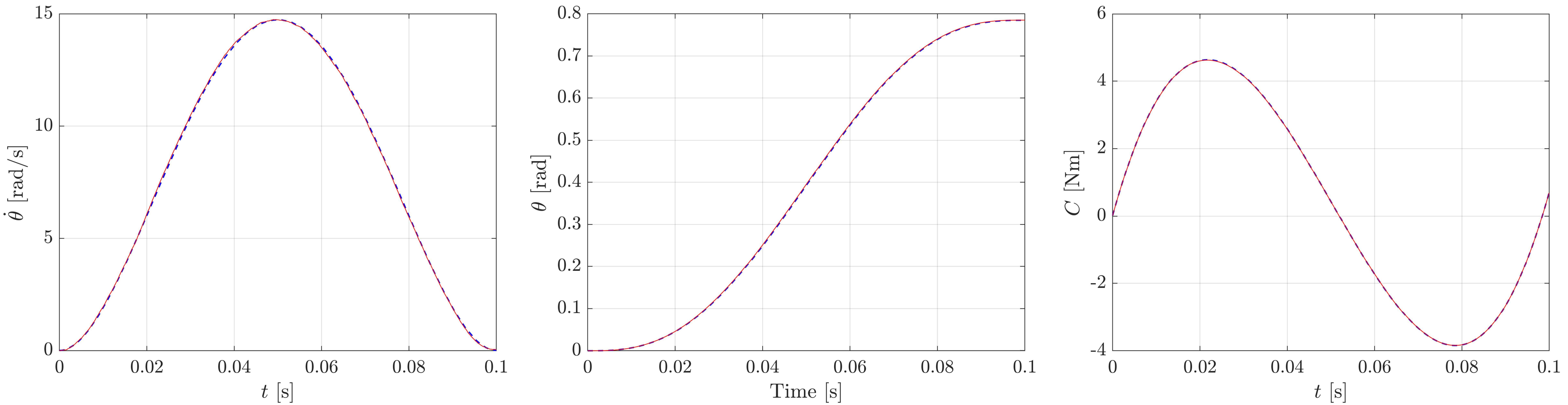
Exercise 1



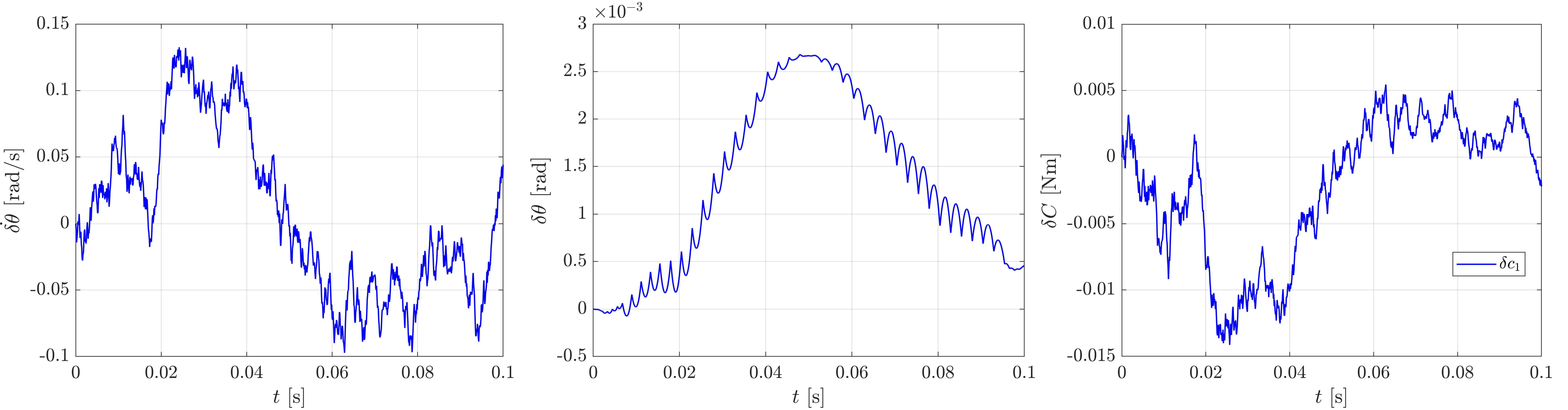


$$\dot{x} = f\big(x(t), u(t)\big)$$

Exercise 1



Exercise 1


0.15
0.1
0.05
0
-0.05
-0.1
$\dot{\delta\theta}$ [rad/s]
0
0.02
0.04
0.06
0.08
0.1
t [s]
$\times 10^{-3}$
3
2.5
2
1.5
1
0.5
0
-0.5
$\delta\theta$ [rad]
0
0.02
0.04
0.06
0.08
0.1
t [s]
0.01
0.005
0
-0.005
-0.01
-0.015
δC [Nm]
δc_1
0
0.02
0.04
0.06
0.08
0.1
t [s]

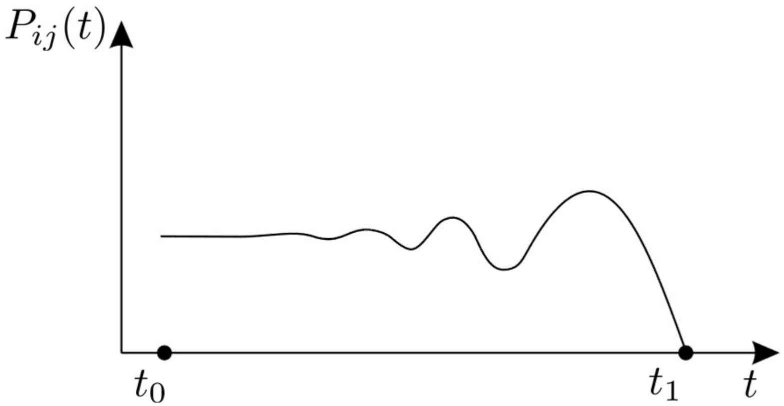
HANDS-ON

- *Explore the Matlab functions MainEs1.M, DRE.M, f.M, fx.M, fc.M.*
- *Explore the Simulink model SimExe1.slx. In particular, discuss the implementation of the nonlinear system and how the gain matrix and trajectory are imported to the Simulink model.*
- *Try to change the weighting matrices and discuss how the stability of the system changes along the trajectory.*

Analysing the typical behavior of the solution of the DRE one can recognize an asymptotic behavior for $t \to 0$.

$$\dot{P}(t) = -(A^T(t)P(t) + P(t)A(t) - P(t)B(t)R^{-1}B^T(t)P(t) + Q)$$

Therefore, the time derivative $\dot{P}(t)$ *is zero for* $t \to 0$ *(i.e. far enough from the transient zone).*



Thus, for an infinite time interval it is reasonable to approximate $\dot{P}(t) = 0$. *The DRE becomes an algebraic equation:*

$$-(A^T P(t) + P(t)A - P(t)BR^{-1}B^T P(t) + Q) = 0$$

Exercise 2 – infinite time control

In order to solve the nonlinear, quadratic matrix algebraic equation (ARE) it is possible to use the lqr.M Matlab function.

```
% Gain matrix and Poles of the controlled system
[K,PP,PolesC] = lqr(A,B,Q,R);
```

- A, B *stability and control matrices*
- Q, R *weighting matrices*
- K *Gain matrix*
- *PP* *Solution of the ARE*
- $PolesC$ *Poles of the controlled system*

Exercise 2 – hints for the choice of the weights

Weighting matrices:

- *State penalty term*

$$x(t)^T Q x(t) = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_N \end{pmatrix}^T \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1N} \\ q_{21} & q_{22} & \cdots & q_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ q_{N1} & q_{N2} & \cdots & q_{NN} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_N \end{pmatrix}$$

$q_{ij} = q_{ji}$ *(Q symmetric)*
$\det(Q) \geq 0$ *(Q positive semi-definite)*

$$x(t)^T Q x(t) = q_{11}x_1^2 + q_{12}x_1x_2 + q_{21}x_1x_2 + q_{22}x_2^2 + \cdots$$

- *Input penalty term*

$$u(t)^T R u(t) = \begin{pmatrix} u_1 \\ u_2 \\ \cdots \\ u_N \end{pmatrix}^T \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1N} \\ r_{21} & r_{22} & \cdots & r_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ r_{N1} & r_{N2} & \cdots & r_{NN} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \cdots \\ u_N \end{pmatrix}$$

$r_{ij} = r_{ji}$ *(R symmetric)*
$\det(R) > 0$ *(R positive definite)*

$$u(t)^T R u(t) = r_{11}u_1^2 + r_{12}u_1u_2 + r_{21}u_1u_2 + r_{22}u_2^2 + \cdots$$

Exercise 2 – hints for the choice of the weights

➢ *Diagonal terms*

In order to reduce the oscillations of the state variable, the matrix Q can be chosen diagonal:

$$(t)^T Qx(t) = q_{11}x_1^2 + q_{22}x_2^2 + \cdots + q_{NN}x_N^2$$

An increase of the term q_{ii} gives a gain matrix K that tend to reduce the oscillation of the state variable x_i.

NOTE:
For the state weight matrix Q it is possible to introduce some diagonal elements q_{ii} equal to zero, i.e. it is possible not to enter the variable x_i in the weight function J (Q is positive semi-definite). For the control matrix R this is not possible (R is positive definite): all the control actions must be weighted in J.

Exercise 2 – hints for the choice of the weights

- *Off-diagonal terms*

Off-diagonal term are used to reduce combinations of the state variables.
For example: we want a control that tends to reduce the oscillation of the difference between x_1 *and* x_2*. We can define:*

$$w = x_1 - x_2$$

$$x(t)^T Q x(t) = a w^2 = a x_1^2 + a x_2^2 - 2a x_1 x_2$$

The correspondent state weighting matrix is:

$$Q = \begin{bmatrix} a & -a & \\ -a & a & \cdots \\ & \cdots & \cdots \end{bmatrix}$$

Exercise 2 – hints for the choice of the weights

➢ *Off-diagonal terms: generalization*

In general, if we want to investigate on a vector **w** *of m quantities that are linear combination of state variables, defined as:*

$$\boldsymbol{w} = [\Lambda]\boldsymbol{x}$$

The cost function can be defined as:

$$J = \int_0^\infty (\boldsymbol{w}^T[\bar{Q}]\boldsymbol{w} + \boldsymbol{u}[R]\boldsymbol{u})dt$$

Where Q *is again a diagonal matrix to weight the different components of the vector w:*

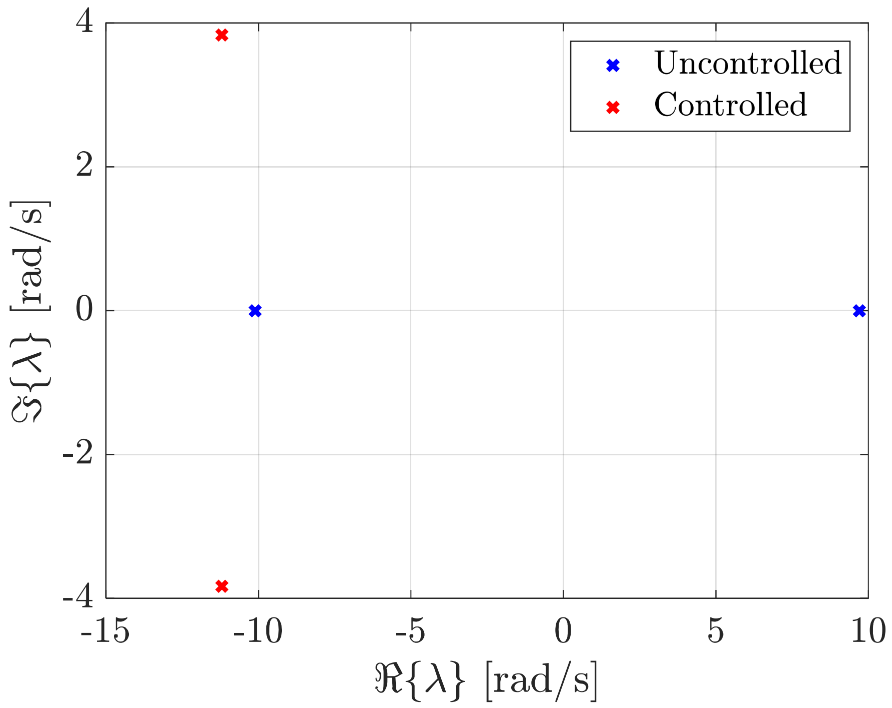
$$w^T Q w = \bar{q}_{11} w_1^2 + \bar{q}_{22} w_2^2 + \cdots$$

Substituting the definition of w, the problem assumes the usual form:

$$J = \int_0^\infty (\boldsymbol{x}^T[\bar{Q}]\boldsymbol{x} + \boldsymbol{u}[R]\boldsymbol{u})dt \qquad \text{Where: } [Q] = [\Lambda]^T[\bar{Q}][\Lambda]$$

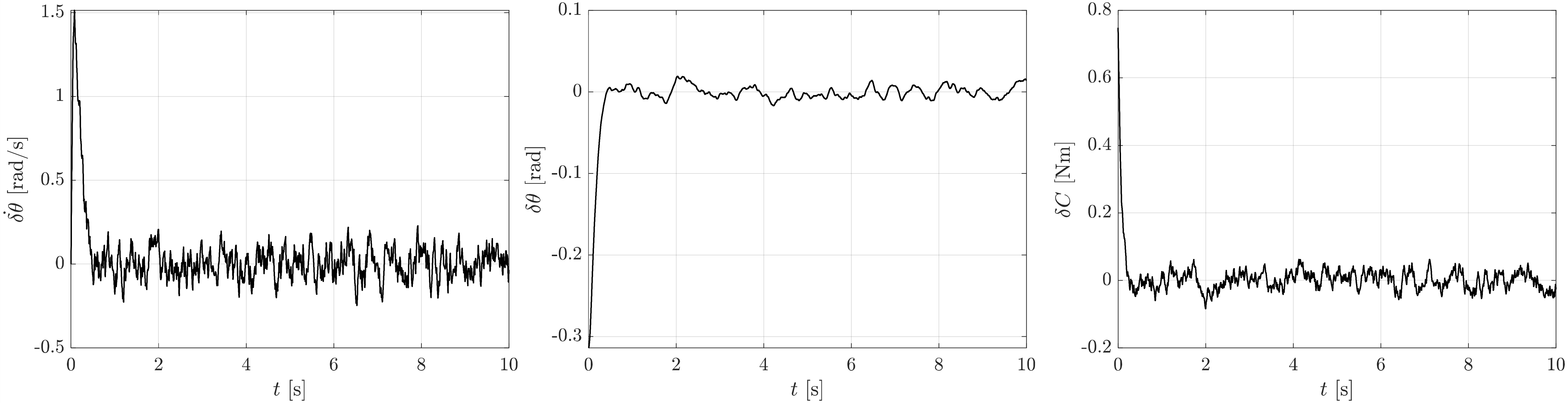
Exercise 2 – numerical results

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad R = 1$$



Exercise 2 – numerical results

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad R = 1$$



HANDS-ON

- *Explore the Matlab functions MainEs2.M*

- *Explore the Simulink model SimExe2.slx. In particular, discuss the implementation of the nonlinear system and how the gain matrix and trajectory are imported to the Simulink model.*

- *Try to change the weighting matrices and discuss on the stability of the system. Try to plot the vector field for a different choice of the gain matrix.*