UNIVERSITÉ DE LORRAINE

POLYTECH NANCY

Identification of linear continuous-time canonical state-space models

Hugues GARNIER

hugues.garnier@univ-lorraine.fr

State-space continuous-time model representation
A refresher

General form for nonlinear system:

$$\dot{x} = f(x,u)$$
$$y = h(x,u)$$

Where:

$x =$ vector of state variables $(n \times 1)$

$u =$ the control input vector $(r \times 1)$

$y =$ is the output vector $(m \times 1)$

This is for a **n^{th} order system with *r* inputs and *m* outputs.**

This can describe both *linear* and *nonlinear* systems.

Linear system:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

Where:

$A =$ system matrix $(n \times n)$

$B =$ input matrix $(n \times r)$

$C =$ output matrix $(m \times n)$

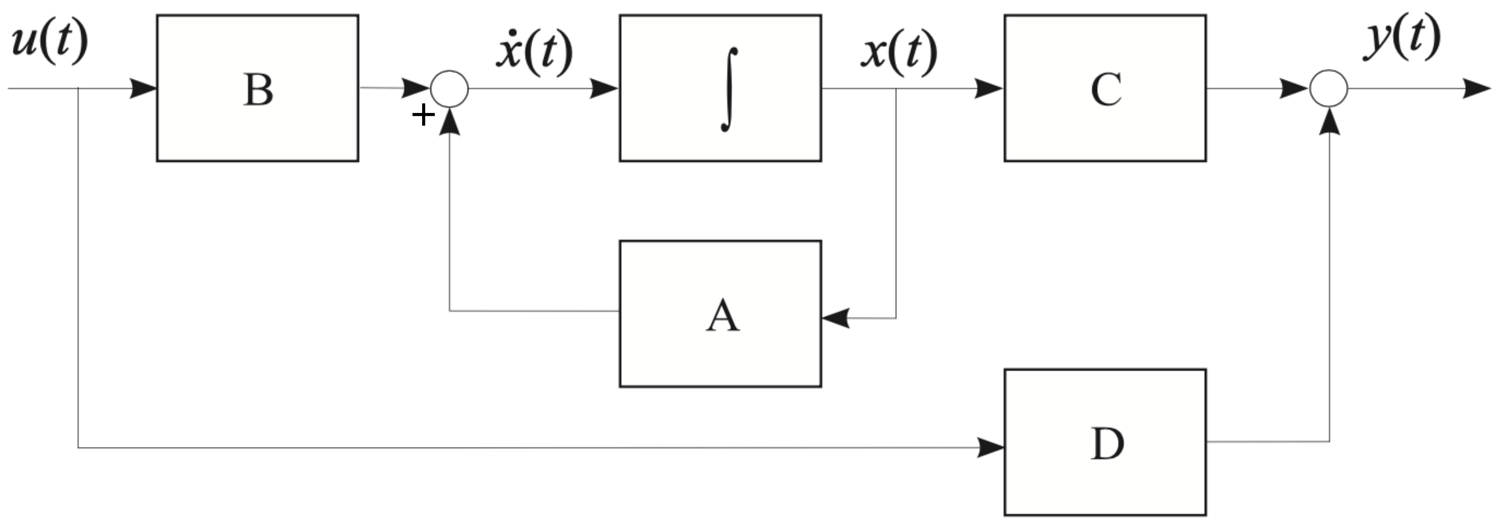
$D =$ feedforward matrix $(m \times r)$

The system state

$$x = [x_1 \quad x_2 \quad x_3 \ldots x_{n-1} \quad x_n]^T$$

Continuous-time linear state-space model

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$



Identification of linear state-space models

It would be nice to be able to identify a state-space model *(A, B, C, D, K)* from measured input/output data *(K matrix to account for the noise effect)*

$$\dot{x}(t) = Ax(t) + Bu(t) + Ke(t)$$
$$y(t) = Cx(t) + Du(t) + e(t)$$

Should work fine:

- For SIMO and MIMO systems
- For continuous-time models (closer to the physics)
- *For discrete-time models (possible but not considered here)*

Main user parameter to be selected: model order (*A* matrix order)

Issue: number of parameters is large when the state-space model has a general form

- 4th-Order system has 9 parameters

$$Y = \left(\frac{c_4 s^4 + c_3 s^3 + c_2 s^2 + c_1 s + c_0}{s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0}\right) U$$

- State-Space has 25 parameters

$$sX_{4x1} = A_{4x4} X_{4x1} + B_{4x1} U$$
$$Y = C_{1x4} X_{4x1} + D_{1x1} U$$

State-space realizations

- There are an infinite number of possible realizations of any system
  - A minimal realization is any form in which A has the smallest possible dimension

- A state transformation is a rotation of the state vector by an invertible matrix T such that

$$z(t) = Tx(t)$$

- State transformation yields an equivalent state-space representation of the system, with

$$\dot{z}(t) = \tilde{A}z(t) + \tilde{B}u(t)$$
$$z(t) = \tilde{C}z(t) + \tilde{D}u(t)$$

where

$$\tilde{A} = TAT^{-1}$$
$$\tilde{B} = TB$$
$$\tilde{C} = CT^{-1}$$
$$\tilde{D} = D$$

Canonical state-space forms

- Certain minimal realizations known as ***canonical forms*** can be useful for *dynamic system identification*, theory and analysis
  - Modal form
  - Controllable canonical form
  - Observable canonical form
  - Controllable companion form
  - Observable companion form – sometimes known as observability canonical form

- Why use canonical form for state-space model identification
  - **Parameter reduction**: canonical forms **minimize the number of free parameters**, improving computational efficiency and reducing overfitting risks
  - **Numerical stability**: imposing structural constraints ensures better conditioning and robustness in parameter estimation
  - **Physical interpretability**: canonical forms often align better with physical interpretations of the system, especially in control applications

Controllable companion form

Represents a state-space system in a reduced parameter form where many elements of A and B matrices are fixed to zeros and ones.

The free parameters appear in only a few of the rows and columns in state-space matrices A, C and K. The free parameters are identifiable and can be estimated to unique values.

Controllable Companion Form

In companion realizations, the characteristic polynomial of the system appears explicitly in the A matrix. For a SISO system with characteristic polynomial

$$P(s) = s^n + \alpha_{n-1}s^{n-1} + \alpha_{n-2}s^{n-2} + \ldots + \alpha_1 s + \alpha_0,$$

the corresponding controllable companion form has

$$A_{ccom} = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & -\alpha_0 \\ 1 & 0 & 0 & \ldots & 0 & -\alpha_1 \\ 0 & 1 & 0 & \ldots & 0 & -\alpha_2 \\ 0 & 0 & 1 & \ldots & 0 & -\alpha_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & -\alpha_{n-1} \end{bmatrix}, \quad B_{ccom} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

For multi-input systems, A_{ccom} has the same form, and the first column of B_{ccom} is as shown. This form does not impose a particular structure on the rest of B_{ccom} or on C_{ccom} and D_{ccom}.

Obtaining Controllable Companion Form

The command `csys = compreal(H,"c")` computes a controllable companion-form realization of H by using the state transformation `T = ctrb(H.A,H.B)` to put the A matrix into companion form.

When performing system identification using commands such as `ssest` or `n4sid`, obtain companion form by setting `Form` to `companion`.

For more information about the distribution of free parameters in the canonical form, see the Appendix 4A, pp 132-134, on identifiability of black-box multivariable model structures in *System Identification: Theory for the User*, by Lennart Ljung, Prentice Hall PTR, 1999 (equation 4A.16)

Observable companion form

also known as observability canonical form

Represents a state-space system in a reduced parameter form where many elements of A and C matrices are fixed to zeros and ones.

The free parameters appear in only a few of the rows and columns in state-space matrices A, B and K. The free parameters are identifiable and can be estimated to unique values.

Observable Companion Form

A related form is obtained using the observability state transformation `T = obsv(H.A,H.B)` instead of `T = ctrb(H.A,H.B)`. This form is the dual (transpose) of controllable companion form, as follows:

$$A_{ocom} = A^T_{ccom}$$
$$B_{ocom} = C^T_{ccom}$$
$$C_{ocom} = B^T_{ccom}$$
$$D_{ocom} = D^T_{ccom}.$$

In particular,

$$A_{ocom} = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & -\alpha_3 & \ldots & -\alpha_{n-1} \end{bmatrix}, \quad C_{ocom} = [1 \quad 0 \quad \ldots \quad 0].$$

This form is sometimes known as *observability canonical form* [1], but it is different from observable canonical form.

Obtaining Observable Companion Form

The command `csys = compreal(H,"o")` computes an observable companion-form realization of `H` by using the state transformation `T = ctrb(H.A,H.B)` to put the A matrix into companion form.

When performing system identification using commands such as `ssest` or `n4sid`, obtain this form by setting `Form` to `canonical`.

Estimating canonical state-space models with the System Identification toolbox for Matlab

You can estimate state-space models with chosen parameterization at the command line with the **ssest** routine

For example, to specify an observability **canonical** form, use the **'Form'** name-value pair input argument, as follows:

```
>> opt = ssestOptions('Form','canonical');
>> Mss = ssest(data,n,opt);
>> display(Mss);
```

Similarly, set 'Form' as 'modal' or 'companion' to specify 'modal' decomposition and 'companion' canonical forms, respectively

We will set the 'Form' as 'canonical' when using ssest in the following

Algorithm overview of the SSEST routine

The SSEST routine operates in the following steps:

Step 1: Subspace initialization

1. Applies subspace techniques to approximate the observability and controllability matrices
2. Provides an initial estimate for the system matrices A, B, C, D

Step 2: State and noise modeling

Kalman Filter:

1. Estimates the unmeasured states *x(t)* based on the initial model and observed data
2. Computes the state covariance matrix and noise covariance matrices

Noise parameter estimation:

1. Models the covariance of process noise *w(t)* and measurement noise *v(t)*
2. Incorporates these covariances into the likelihood function

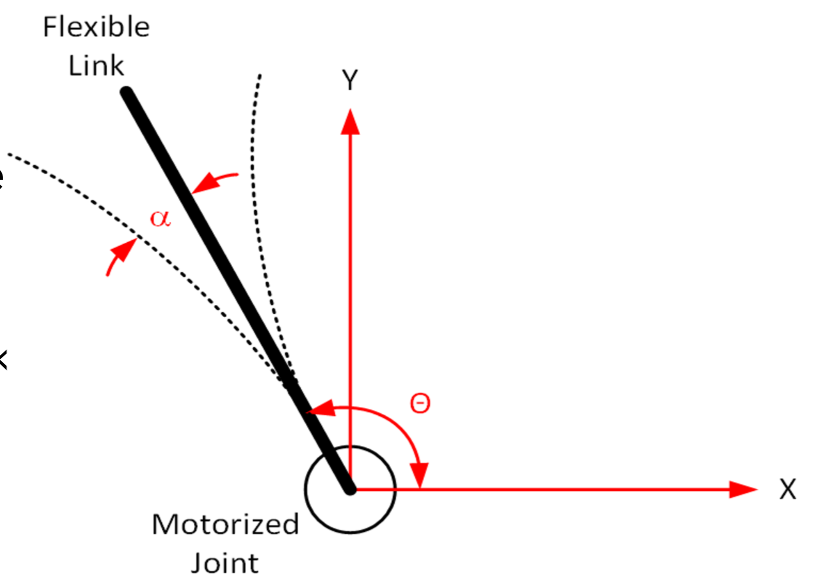
Step 3: Maximum likelihood estimation

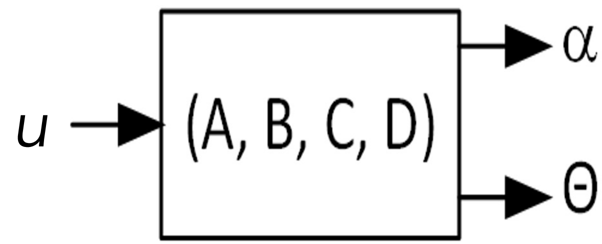
1. Refines the model parameters by maximizing the likelihood of observing the output *y(t)* given the input *u(t)*
2. Uses iterative optimization (e.g., Expectation-Maximization or gradient-based methods)

Identification of a state-space model for a SIMO a flexible link by using SSEST routine

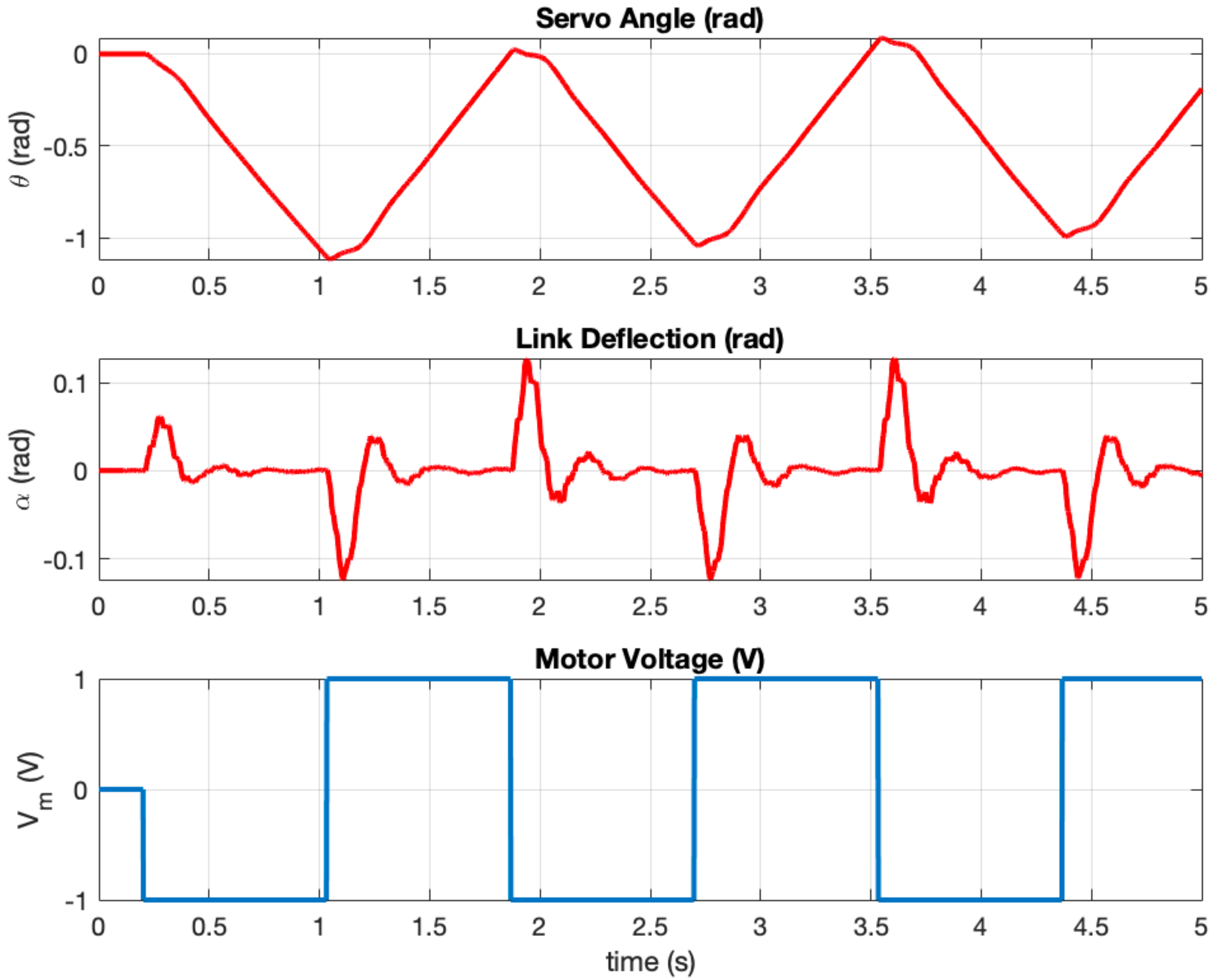
It is a 1 input - 2 outputs (SIMO) system

- $u(t)$: the motor voltage proportional to the torque $\tau(t)$ applied
- $\alpha(t)$: the angle deflection of the flexible link
- $\theta(t)$: the angular position of the servo base





Input/output data for model estimation



Model structure selection and estimation by SSEST

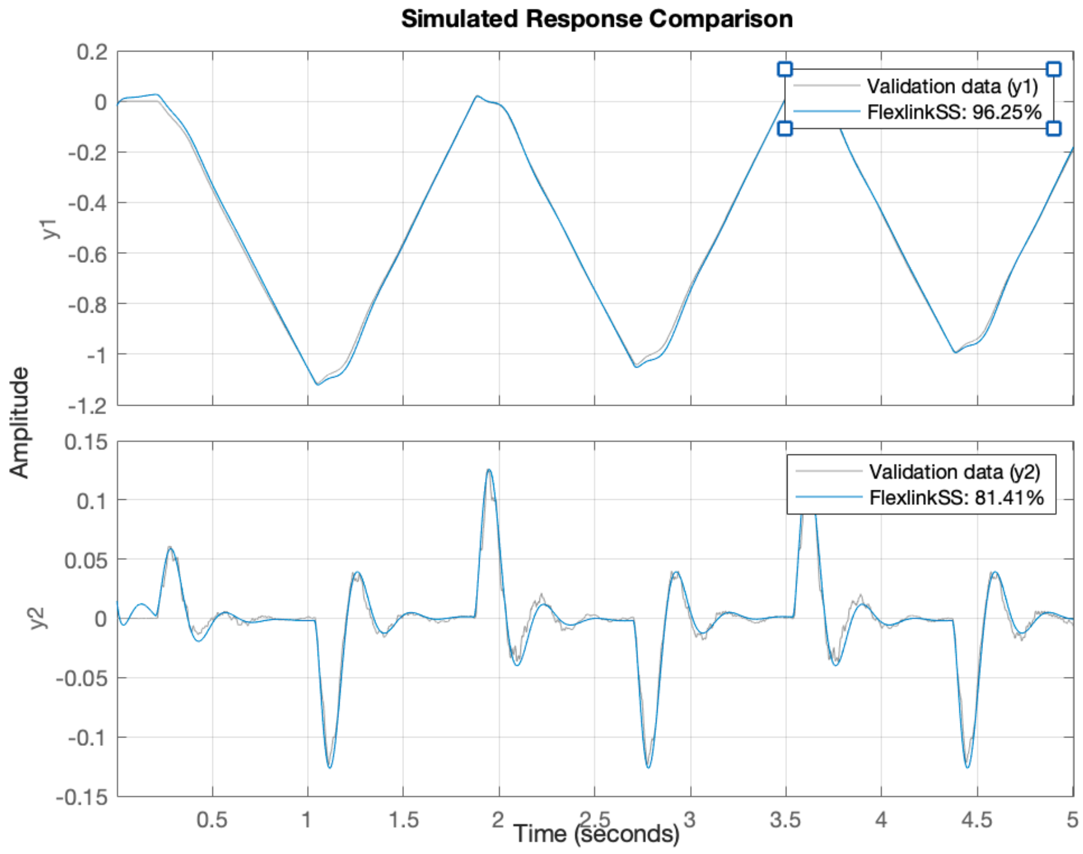
- A 4th order continuous-time model has been selected and identified by using SSEST *(from the SID toolbox)*

$$\dot{x}(t) = Ax(t) + Bu(t) + Ke(t)$$
$$y(t) = Cx(t) + e(t)$$

```
>> opt = ssestOptions('Form','canonical');
>>[FlexlinkSS] = ssest(MeasuredData,4,opt);
>> display(FlexlinkSS);
A =
            x1       x2       x3       x4
   x1        0        1        0        0
   x2     -3.9   -47.42    559.8   0.3877
   x3        0        0        0        1
   x4    4.445    50.55    -1007   -2.485


  C =
         x1   x2   x3   x4
   y1     1    0    0    0
   y2     0    0    1    0

B =
             u1
   x1   -0.1807
   x2      79.6
   x3   -0.2154
   x4    -84.14

K =
            y1       y2
   x1    316.7   -4.734
   x2     7535    762.3
   x3    23.13    769.4
   x4    -5842     1845
```

Validation results with the estimation data

```
>>compare(MeasuredData, FlexlinkSS)
   % very good fit to data for both outputs
```

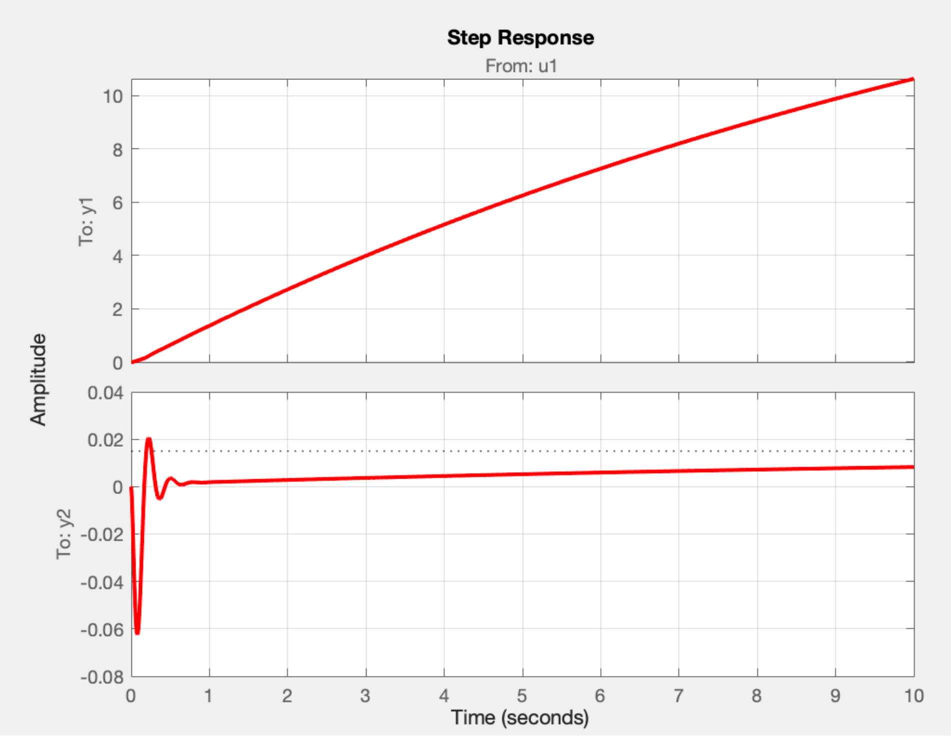


Poles of the identified model

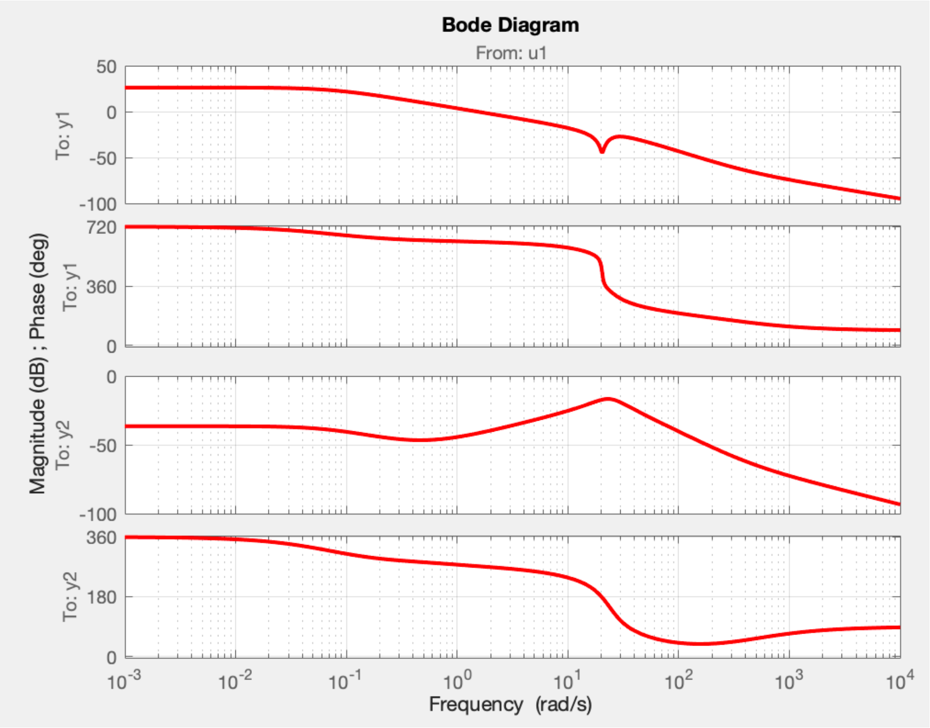
```
>> damp(FlexlinkSS)

        Pole                    Damping       Frequency       Time Constant
                                            (rad/seconds)       (seconds)

 -7.42e-02                     1.00e+00        7.42e-02          1.35e+01
 -7.88e+00 + 2.25e+01i         3.30e-01        2.38e+01          1.27e-01
 -7.88e+00 - 2.25e+01i         3.30e-01        2.38e+01          1.27e-01
 -3.41e+01                     1.00e+00        3.41e+01          2.93e-02
```

Step and Bode responses of the identified model

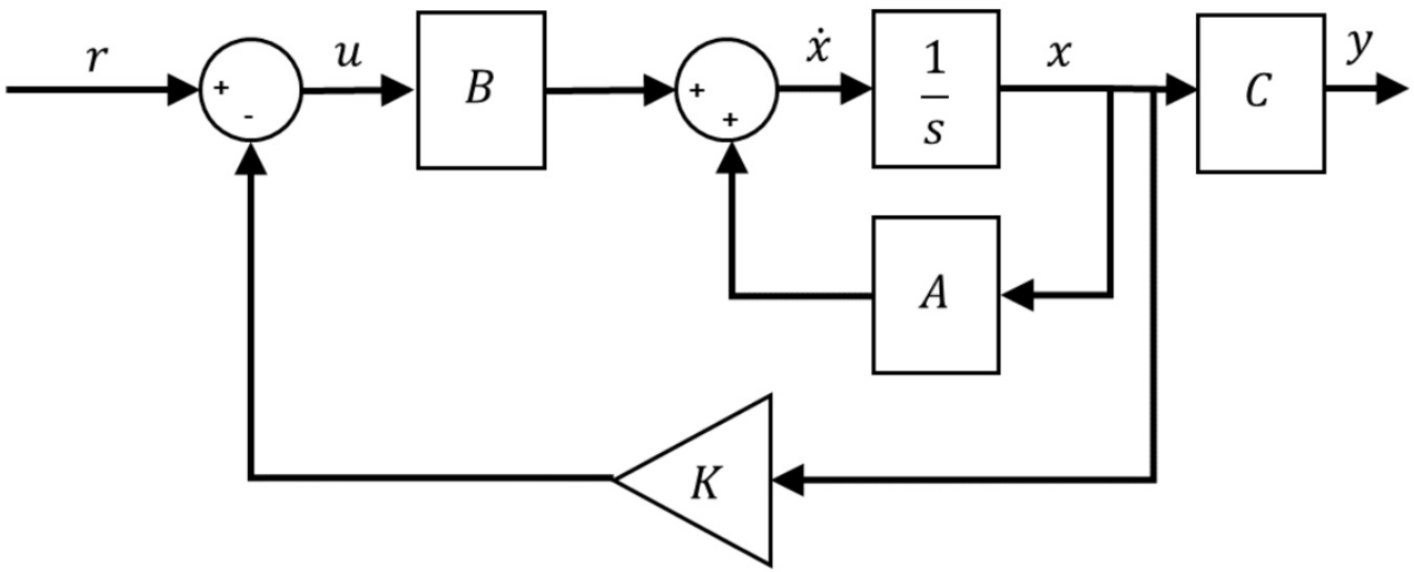
```
>>step(FlexlinkSS)
```




```
>>bode(FlexlinkSS)
```



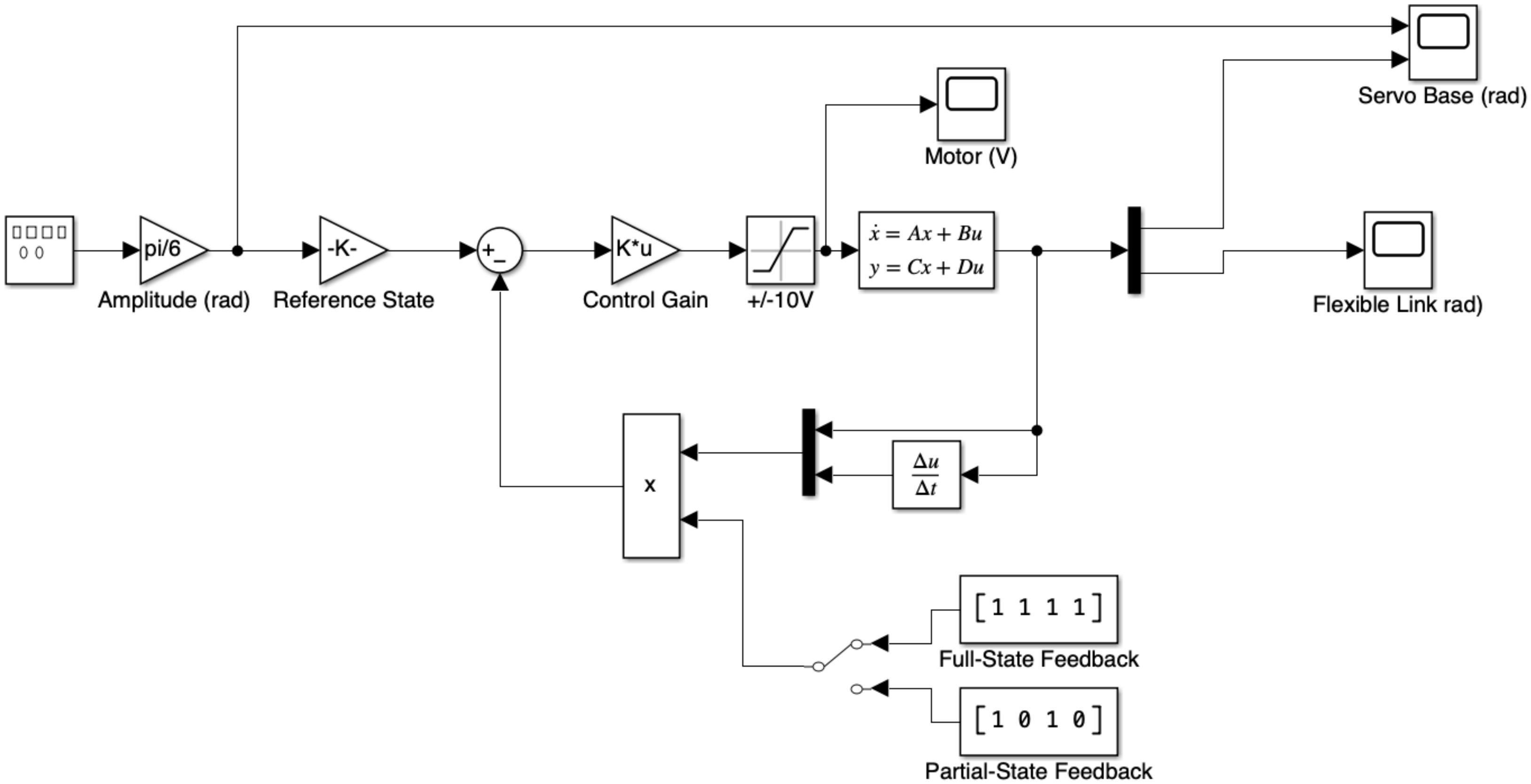
Model-based LQR control




```
sys_ol = ss(FlexlinkSS);

% Set Q and R LQR weighting matrices
Q = diag([500 0 0 2]);
R = 1;

% Generate feedback control gain using LQR
K = lqr(sys_ol,Q,R);
```

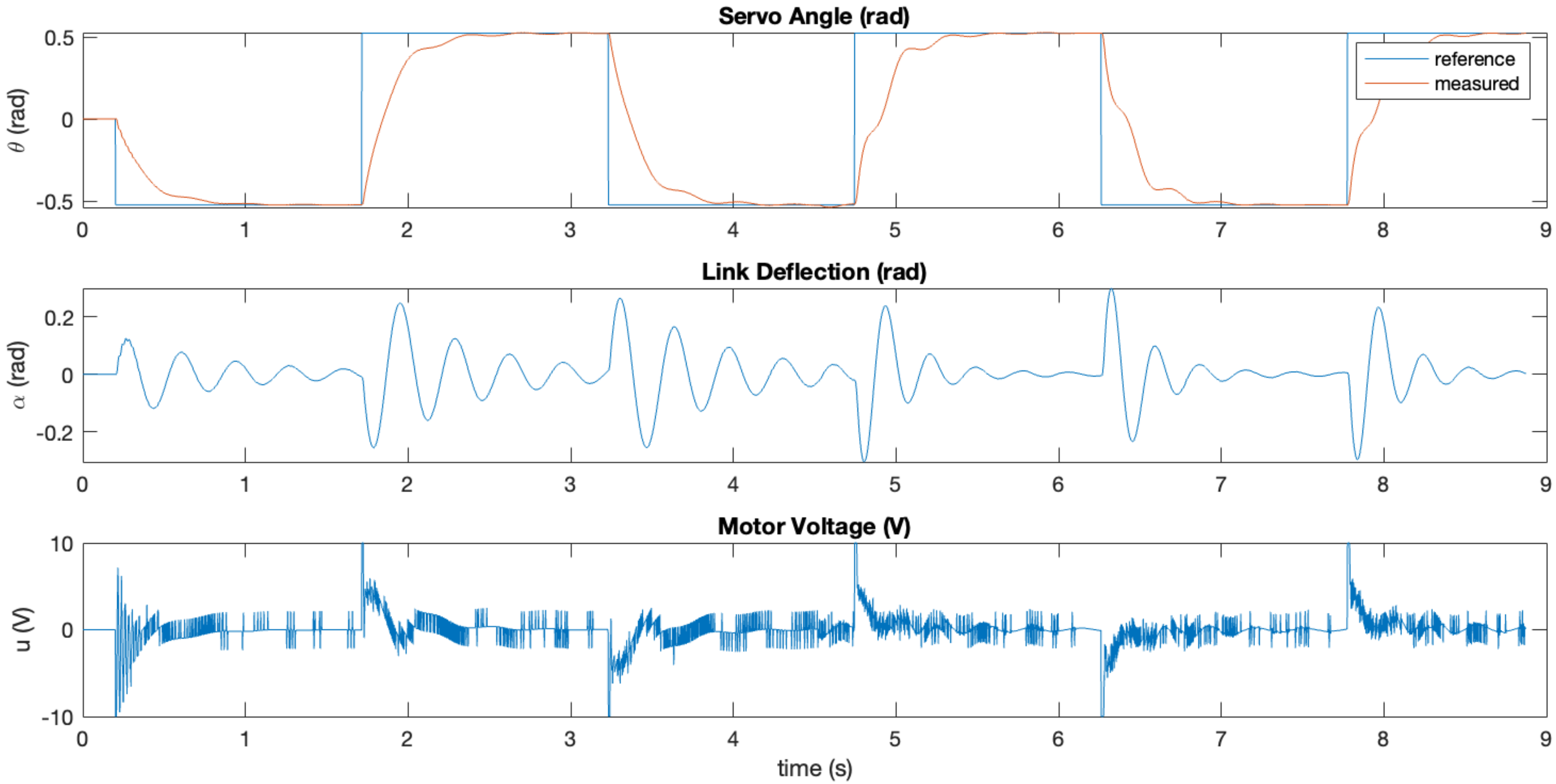

H. Garnier

Model-based LQR control



H. Garnier

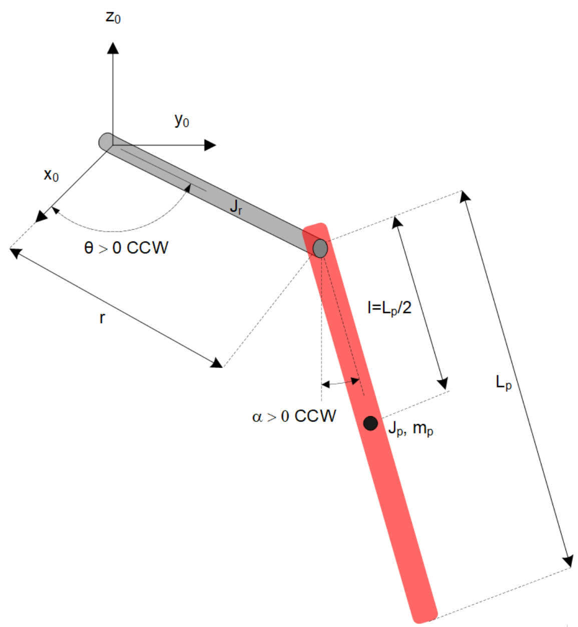
Performance evaluation of the LQR control

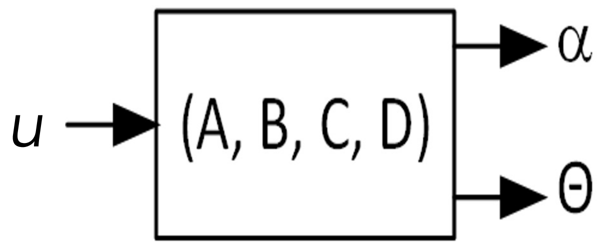


Identification of a state-space model for a SIMO tower crane by using SSEST routine

Motions of a rotary pendulum are similar to that of a tower crane. It is a 1 input - 2 outputs system

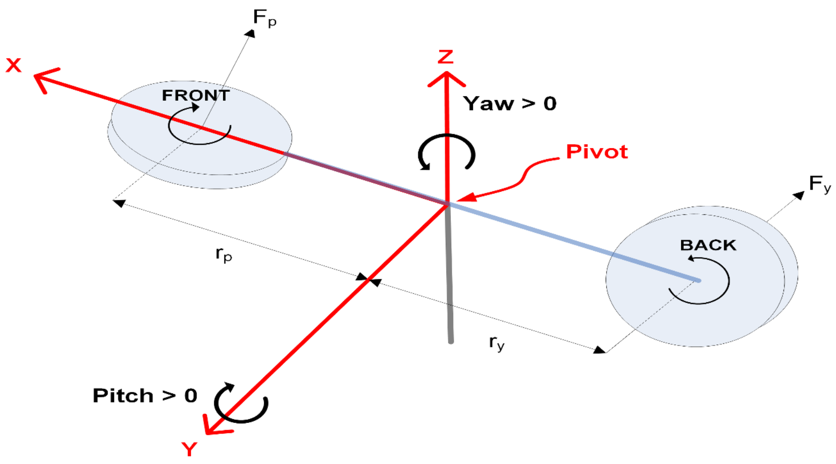
- *u(t)*: the motor voltage
- α*(t)*: the angular position of the pendulum
- θ*(t)*: the angular position of the servo base/arm



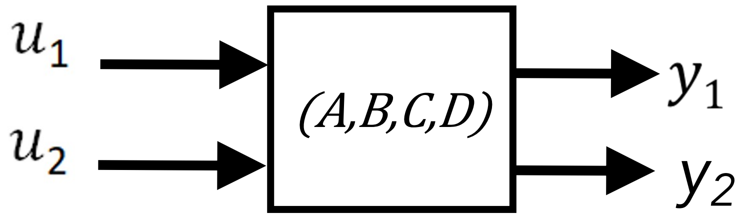


H. Garnier

Example: 2 inputs - 2 outputs AERO helicopter



- Two inputs
  - Front rotor thrust
  - Read rotor thrust
- Two outputs *(no roll)*
  - Pitch
  - Yaw
- Coupled dynamics
  - Pitch/yaw affect each other



We will study the identification and control of the AERO helicopter