# Control of a Flexible Joint

Computer-based Exercises for ME-326

Fall 2021

## Introduction

The objective of these exercises is to illustrate different aspects of controller design on a laboratory setup using the Control System Toolbox of Matlab. The experimental setup is a rotary flexible joint produced by Quanser to demonstrate real-world control challenges encountered in some industrial large-gear robotic equipment. Five computer exercise modules are planned. In the first module a linearized model of the system is used to study the performance of the closed-loop system with a proportional controller. Some PID and cascade controllers are designed and validated in simulation during the second module. The loop shaping method is used to design a PID and a lead-lag compensator for the system in the third module. The state and output feedback controllers are designed and tested in simulation in the fourth module. The last module concerns the design of a digital RST controller for the system. The work is done by the groups of three students and a report (in `mlx` format) should be prepared by each group including the results of each module. For each module there is a module zip file that contains the template `mlx` file and a `readme.md` for help. The final exercise codes should be submitted in moodle as a zip file by the end of semester (January 10th 2022). The reports will be evaluated and counted for 10 points in the final grade.

## System Description

The system contains two units: the Rotary Servo Base Unit which is a geared servo-mechanism system and the Rotary Flexible Joint unit which is a free arm attached to two identical springs that are mounted to an aluminium chassis (see Fig. 1).

The Rotary Servo Base Unit consists of a DC motor that drives a small pinion gear through an internal gearbox. The pinion gear is fixed to a larger middle gear that rotates on the load shaft. The chassis of the Rotary Flexible Joint is mounted on the load gear of the Servo Base Unit and can rotate freely.

Two different variables can be measured:

- The rotation angle of the load shaft and the chassis of the rotary flexible joint $\theta(t)$

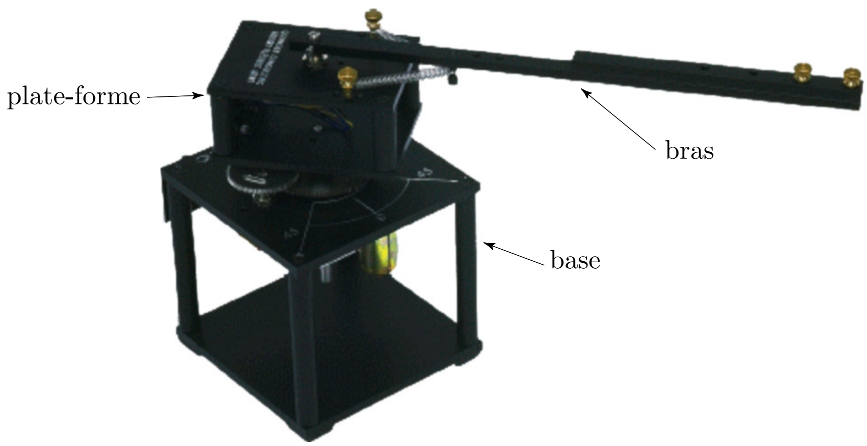



Figure 1: Rotary Servo Base Unit with Rotary Flexible Joint

- The angle between the arm and the chassis of the flexible joint $\alpha(t)$

In the following, we are interested in the angle between the arm and the servo base unit: $\theta(t) + \alpha(t)$. This angle will be considered as the output of our system: $y(t) = \theta(t) + \alpha(t)$. The input of the system $u(t)$ is the dc voltage applied to the DC motor.

# Modeling

The objective of this section is to find the transfer function between the output $y$ and the input of the system $u$. A schematic diagram of the system is shown in Fig. 2.

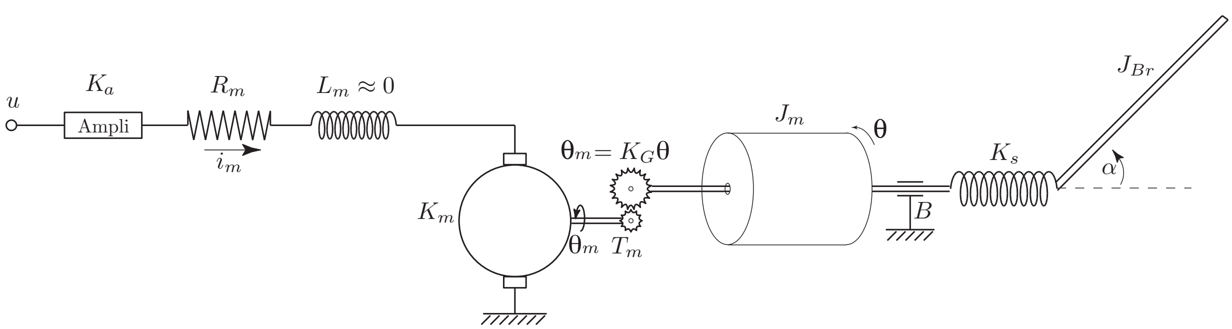



Figure 2: Schematic diagram of the whole system

In the first step, the dynamic equations of the electrical and mechanical part of the DC motor is obtained. It is supposed that the motor's stator consists of permanent magnets that provide a constant magnetic field and the armature inductance can be neglected. Using

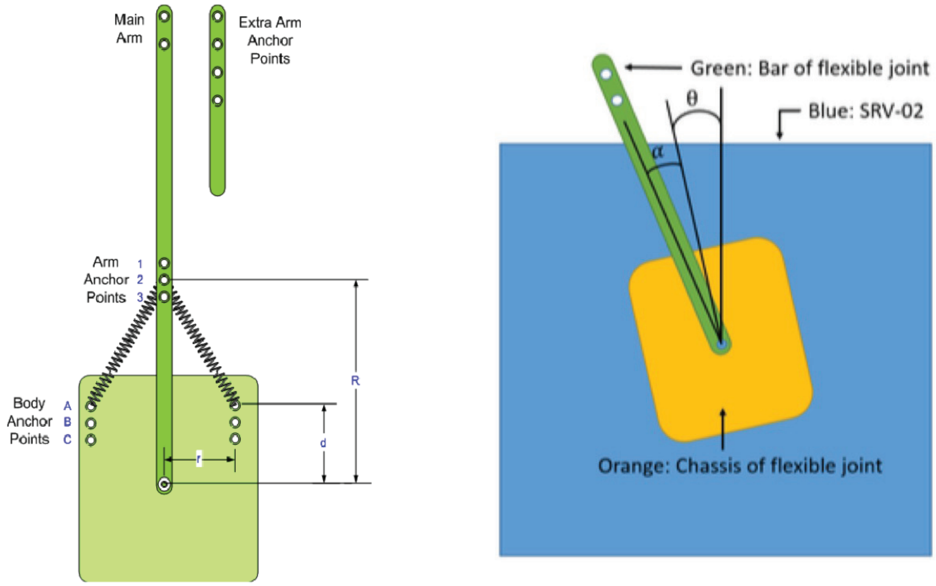



Figure 3: Schematic diagram of the flexible joint module

Kirchhoff's voltage law, the following equation can be written for the electrical component of the motor:

$$K_a u(t) = R_m i_m(t) + E_{emf}(t) \tag{1}$$

where the electromotive force induced voltage is equal to $E_{emf}(t) = K_m \dot{\theta}_m(t)$. From the above equation the armature current is obtained as:

$$i_m(t) = \frac{K_a u(t) - K_m \dot{\theta}_m(t)}{R_m} \tag{2}$$

By applying Newton's law to the motor shaft we get:

$$J_{\text{mot}} \ddot{\theta}_m(t) + \frac{T_L(t)}{K_G} = T_m(t) \tag{3}$$

where $J_{\text{mot}}$ is the motor inertia, $T_m(t) = K_m i_m(t)$ the motor torque and $T_L(t)/K_G$ the load torque seen through the gear (considering no loss in the gear).

The arm of the Flexible Joint module is attached to the body unit by two identical springs (see Fig. 3). At a given $\alpha$, one of the springs is compressed while the other is stretched (with respect to the joint's stationary state). This, generates two forces that acts on the arm anchor point. These two forces will cause a torque about the joint which is nonlinear with respect to $\alpha$ (it depends on $\sin\alpha$ and $\cos\alpha$). This relation can be linearized around small $\alpha$ to give the following simplified equation:

$$T_s(t) = K_s \alpha(t) \tag{4}$$

where $K_s$ represent the equivalent stiffness of the flexible joint. Newton's second law for the rotating body around the joint is:

$$J_{mod} \ddot{\theta}(t) + b\dot{\theta}(t) + J_{Br}(\ddot{\theta}(t) + \ddot{\alpha}(t)) = T_L \tag{5}$$

and for the arm:

$$J_{Br}(\ddot{\theta}(t) + \ddot{\alpha}(t)) + K_s\alpha(t) = 0 \tag{6}$$

where $J_{\text{mod}}$ and $J_{Br}$ as the inertia of the module and the arm respectively. Only a viscous damping $b$ for the movement of the joint body is considered.

Note that $\theta_m = K_G\theta$ and the total equivalent inertia at the load side is $J_m = J_{\text{mod}} + K_G^2 J_{\text{mot}}$. In order to find the transfer function $G(s) = Y(s)/U(s)$ we take the Laplace transform from Eq. 6 and we obtain:

$$J_{Br}s^2\theta(s) + (J_{Br}s^2 + K_s)\alpha(s) = 0 \quad \Rightarrow \quad \alpha(s) = -\frac{J_{Br}s^2}{(J_{Br}s^2 + K_s)}\theta(s) \tag{7}$$

Then, we take the Laplace transform of Eq. 5:

$$(J_{mod}s^2 + bs)\theta(s) + K_s\frac{J_{Br}s^2}{(J_{Br}s^2 + K_s)}\theta(s) = T_L(s) \tag{8}$$

Next, we go to the motor side and take the Laplace transform of Eq. 3 :

$$K_G J_{\text{mot}} s^2\theta_m(s) + T_L(s) = K_G K_m i_m(s) \tag{9}$$

and we replace

$$i_m(s) = \frac{K_a u(s) - K_m s\theta_m(s)}{R_m} \quad \text{and} \quad \theta_m(s) = K_G\theta(s)$$

which leads to

$$\left[K_G^2 J_{\text{mot}} s^2 + J_{mod}s^2 + bs + K_s\frac{J_{Br}s^2}{(J_{Br}s^2 + K_s)}\right]\theta(s) = \frac{K_G K_m}{R_m}[K_a u(s) - K_m K_G s\theta(s)]$$

or equivalently

$$\begin{aligned}\left[(J_{Br}s^2 + K_s)(R_m J_m s^2 + R_m bs + K_m^2 K_G^2 s) + R_m K_s J_{Br}s^2\right]\theta(s)& \\ = (J_{Br}s^2 + K_s)K_G K_m K_a u(s)&\end{aligned}$$

where $J_m = J_{mod} + K_G^2 J_{\text{mot}}$. Finally we obtain:

$$G_\theta(s) = \frac{(J_{Br}s^2 + K_s)K_G K_m K_a}{(J_{Br}s^2 + K_s)(R_m J_m s^2 + R_m bs + K_m^2 K_G^2 s) + R_m K_s J_{Br}s^2} \tag{10}$$

$$G_\alpha(s) = \frac{-J_{Br}K_G K_m K_a s^2}{(J_{Br}s^2 + K_s)(R_m J_m s^2 + R_m bs + K_m^2 K_G^2 s) + R_m K_s J_{Br}s^2} \tag{11}$$

$$G(s) = \frac{K_s K_G K_m K_a}{(J_{Br}s^2 + K_s)(R_m J_m s^2 + R_m bs + K_m^2 K_G^2 s) + R_m K_s J_{Br}s^2} \tag{12}$$

**Numerical values:**

| | | | |
|---|---|---|---|
| Gain of power amplifier | $K_a$ : | 2 | [-] |
| Motor resistance | $R_m$ : | 2.2 | [$\Omega$] |
| Torque constant | $K_m$ : | 0.00787 | [Nm/A] |
| Gearbox ratio | $K_G$ : | 50 | [-] |
| Motor Inertia | $J_{\text{mot}}$ : | $3.87 \times 10^{-7}$ | [kgm$^2$] |
| Equivalent rotational stiffness | $K_s$ : | 1.3 | [Nm/rad] |
| Viscous damping | $b$ : | 0.004 | [Nm/(rad/s)] |
| Module Inertia | $J_{\text{mod}}$ : | $3.944 \times 10^{-4}$ | [kgm$^2$] |
| Arm Inertia | $J_{Br}$ : | 0.0037 | [kgm$^2$] |

the transfer function $G(s)$ can be defined in Matlab. In all modules of computer exercises the model $G(s)$ is used as the plant model to be controlled.

# 1 Module 1: Analysis of Feedback Control Systems

The objective of this module is to make the students familiar with some commands of Matlab Control Toolbox for analysis of feedback control systems.

## 1.1 Define a transfer function

The easiest way to define a transfer function is to use the `tf` command. For example in order to define $F(s) = \frac{s+1}{s+2}$ the following commands can be used:

```
s=tf('s');
F=(s+1)/(s+2)
```

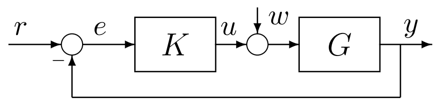
```
F =

  s + 1
  -----
  s + 2

Continuous-time transfer function.
```

## 1.2 Time- and frequency-domain analysis

Using the numerical values for the parameters of the flexible joint define the transfer function $G(s)$ and compute its step response, impulse response, poles, zeros and Bode diagram etc. using `step(G)`, `impulse(G)`, `pole(G)`, `zero(G)`, `bode(G)`, `zpk(G)`, `damp(G)`.

## 1.3 Analysis of the feedback systems

Consider the following feedback control system with a proportional controller $D_c(s) = K$.




Choose $K = 1.2$ and compute the following closed-loop transfer functions:

1. The transfer function between $r$ and $e$.
2. The transfer function between $r$ and $u$.
3. The transfer function between $r$ and $y$.
4. The transfer function between $w$ and $y$.

**Remark:** The arithmetic operations of transfer functions can be done in Matlab control toolbox. However, the result should be simplified if there exist some zero-pole cancellations in the final transfer function. This can be carried out using the `minreal` command (minimal realization).

For example:

```
F1=(s+1)/(s+2);F2=(s+2)/(s+3);
F1*F2

ans =

  s^2 + 3 s + 2
  -------------
  s^2 + 5 s + 6

Continuous-time transfer function.

>> minreal(F1*F2)

ans =

  s + 1
  -----
  s + 3

Continuous-time transfer function.
```

The closed-loop transfer functions can be computed using the `feedback` command as well.

**a)** Plot the unit step responses of the closed-loop system:

- from reference signal to the output,
- from reference signal to the control signal (plant input),
- from reference signal to the tracking error signal (the input of the controller),
- from the disturbance signal (added to the plant input) to the output.

**b)** Compute the closed-loop poles.

**c)** Is the closed-loop system stable?

### 1.4 Computing the ultimate gain

The minimum value of the proportional gain that destabilizes the closed-loop system is called the ultimate gain $K_u$. For the flexible joint, compute the ultimate gain. The ultimate gain can be computed by:

- Increasing gradually the proportional gain $K$ and observing the closed-loop step response. For $K = K_u$ the step response starts to diverge.
- Computing the closed-loop poles. For $K = K_u$ some closed-loop poles are located on the imaginary axis. This can be done with the command `rlocus(G)` that plots the loci of the closed-loop poles (the zeros of $1 + KG$) for the values of $K$ from zero to infinity. The value of $K$ for which the loci traverse the imaginary axis gives the value of the ultimate gain.
- Using the Routh stability criterion we can find the range of $K$ for which the closed-loop system is stable. Then the smallest $K$ that makes the closed-loop system unstable is $K_u$.

### 1.5 Closed-loop step response analysis

Choose $K = 0.6K_u$ and compute the step response of the closed-loop system (between $r$ and $y$). Compute the rise time, settling time and the overshoot (use `stepinfo`). Compute the closed-loop bandwidth (use `bandwidth`).

## 2 Module 2: PID Controller Design

The objective of this module is to tune some PID controllers for the flexible joint. In the first part the ZN methods are tried and then a cascade controller will be designed. The performance of the controllers in tracking and disturbance rejection will be compared.

### 2.1 ZN First method

Design a PID controller by the first method of Ziegler-Nichols. From the step response of the flexible joint model compute the parameters $L$ and $R$, and then find the PID controller parameters from the ZN table. Plot the step response of the closed-loop system from the reference signal to the output.

## 2.2 ZN Second method

Design a PID controller by the second method of Ziegler-Nichols. Compute the ultimate gain, $K_u$) and the ultimate period $P_u$ and find the PID controller parameters from the ZN table. Plot the step response of the closed-loop system from the reference signal to the output. Plot also the output of the closed-loop system after a step disturbance at the input of the plant.

## 2.3 Cascade Controller

A cascade controller can be designed for the flexible joint. Inner loop consists in angle speed control of the flexible arm and the outer loop is dedicated to angle position control. Figure 4 shows the implementation of such a cascade controller where $G_1(s) = G(s) * s$.

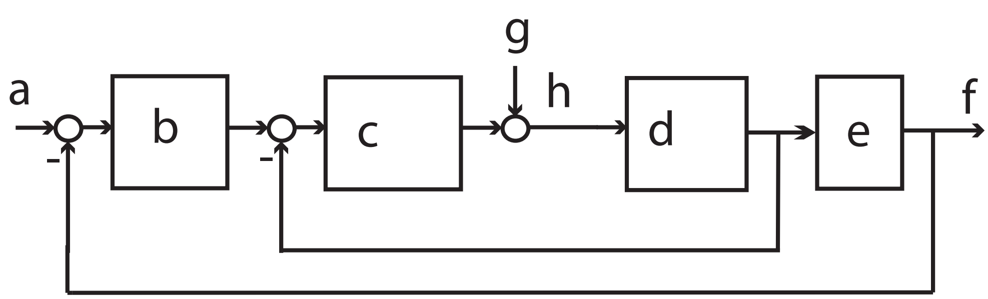



Figure 4: Cascade controller

1. From the step response of $G_1$, identify the parameters $\gamma, \zeta$ and $\omega_n$ of an approximate second-order model for the system (Chapter 3, slide 41):

$$G_1(s) \approx \frac{\gamma\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

2. Design a PID controller, $D'_c(s)$, for the inner loop using the model reference method to achieve a bandwidth of 50 rad/s.

3. Design a proportional controller for the outer loop, $D_c(s) = k_P$, to achieve a desired bandwidth of 2.5 rad/s. Note that the inner loop is much faster than the outer loop so its dynamic can be ignored in the design of the controller for the outer loop.

4. Compute the transfer function between the reference signal $R$ and the output $Y$.

5. Compute the transfer function between the disturbance, $W$, and the output $Y$.

6. Compare the performance of the cascade controller with the ZN-PID controller in terms of tracking and disturbance rejection.

# 3 Module 3: Loop Shaping Method

The objective of this module is to design some controllers for the flexible joint using the loop shaping method.

## 3.1 Proportional controller

Compute a proportional controller $k_P$ to have a crossover frequency of 5 rad/s. Determine the gain margin, the phase margin and the modulus margin of the system.

## 3.2 Lead-lag controller

Compute a lead-lag compensator $D_c(s)$ that :

- Rejects the input step disturbance,
- Gives a crossover frequency of 5 rad/s,
- Ensures a phase margin of 55°.

Validate your results by checking the obtained margins and crossover frequency. Compute the modulus margin.

## 3.3 Comparison with cascade controller

Compare the performance of the cascade controller in 2.3 with the lead-lag controller in 3.2 in terms of tracking and disturbance rejection.

# 4 Module 4: State-Space Method

The objective of this module is to use the state-space approach for modelling and control of the flexible joint.

## 4.1 State-space model of the flexible joint

Let's define the state of the system as $\boldsymbol{x} = [\dot{\theta} \quad \theta \quad \dot{\alpha} \quad \alpha]^T$. We can find $T_L$ from Eq. (3) and (2) as follows:

$$T_L = -K_G^2 J_{\mathrm{mot}} \ddot{\theta} + K_G K_m \left( \frac{K_a u - K_m K_G \dot{\theta}}{R_m} \right)$$

Then we can replace $T_L$ in Eq (5) to find one of the state equations. The other state equation is (6).

- Compute the state-space model of the system $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, D)$.
- Is the system controllable?
- Is the system observable?

## 4.2 State-space controller design

The objective is to design a state space controller with the following specifications:

- Compute a state feedback control using the LQR method. Choose $Q = C^T C$ and determine $R$ such that the control signal remains less than 10 for a unit step reference signal (it should be done iteratively by observing the control signal computed in the last item).
- Design a state estimator using the pole placement technique (use `acker` command). Note that the poles of the estimator (observer) should be faster than the dominant control poles.
- Compute the feedforward gain $\bar{N}$ to have zero steady-state error for a step reference signal.
- Compute the transfer function between the reference signal $r$ and the output $y$.
- Compute the transfer function between the reference signal $r$ and the input $u$.
- Plot the output $y(t)$ and the input $u(t)$ when a unit step signal is applied to $r(t)$.

## 4.3 State-space controller with integrator

In order to have zero steady-state error for a step disturbance signal, the controller should include an integrator. In the state-space framework, this is done by augmenting the state-space model with a new artificial state, which is the integral of the tracking error.

- Find the augmented state-space model of the plant $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$.
- Design a state feedback controller by the pole placement method. The closed-loop dynamic is represented by a second-order polynomial $s^2 + 2\zeta\omega_n s + \omega_n^2$ with $\omega_n = 10$ rad/s and $\zeta = 0.8$ and some fast poles.
- Design a state estimator by the pole placement technique. Note that only the states of the plant model are estimated.
- Consider the state equations for the closed-loop system as:

$$
\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u \\
\dot{\hat{\boldsymbol{x}}} &= \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}u + \boldsymbol{L}(y - \boldsymbol{C}\hat{\boldsymbol{x}}) \\
\dot{x}_I &= r - y \\
u &= -\boldsymbol{K}_0\hat{\boldsymbol{x}} - K_1 x_I \\
y &= \boldsymbol{C}\boldsymbol{x}
\end{aligned}
$$

  Compute the transfer function between $r$ and $y$.

- Compute the transfer function between the reference signal $r$ and the input $u$.
- Plot the control signal $u(t)$ and the output $y(t)$ when a unit step signal is applied to the reference.

# 5 Module 5: Digital Control

The objective of this method is to design an RST digital controller using the pole placement technique for the flexible joint.

## 5.1 Discrete-time model

1. Assume that a closed-loop bandwidth of 10 rad/s is desired for the position control of the flexible joint system. Find an appropriate sampling period $T_s$ for the closed-loop output.

2. Compute a discrete time model for the system using zero-order hold, zero-pole matching and Tustin methods. Use `c2d` command.

3. Compare the step response of the continuous-time system with that of discrete-time models.

4. Compare the Bode diagram of the continuous-time system with that of discrete-time models.

## 5.2 RST controller design

The objective is to design an RST controller with integrator for controlling the position of the flexible joint using the pole placement technique. The desired closed-loop polynomial should have a natural frequency of 10 rad/s and a damping factor of 0.8.

1. Compute the desired closed-loop polynomial.

2. Use the discretized model using the zero-order hold method. Extract the coefficients of numerator and denominator in two column vectors `B` and `A`. Use `tfdata(Gz,'v')`.

3. In order to have an integrator in the controller, compute the coefficients of `A'=AHs` using the convolution command `conv`. Compute the order of polynomials R and S'.

4. Construct the Sylvester matrix and compute R and S polynomials (do not forget to put back the integrator in the controller). Compute the polynomial T.

5. Compute the closed-loop poles and compare them with the desired ones. The coefficients of the closed-loop polynomial can be computed using the `conv` command as `P=conv(A,S)+conv(B,R)`.

6. Compute the output of the system when a unit step is applied to the input of T. The transfer function can be constructed using `tf(conv(B,T),P',Ts,'variable','z^-1')`. Compute the control signal as well.

7. Compare this controller with the stat-space controller in terms of tracking a unit step reference.

## 5.3 Global comparison

Compare this controller with the loop shaping, cascade and state-space controllers in terms of the complexity of the controller structure, the clarity of the design methods, their advantages and disadvantages.