UNIVERSITÀ DELLA CALABRIA
DIPARTIMENTO DI INGEGNERIA INFORMATICA, MODELLISTICA, ELETTRONICA E SISTEMISTICA
DIMES

Optimal Control Techniques applied to the mechanical system

Thesis of the project for the Optimal Control course

Professor:
Domenico Famularo

Student:
Rustemova Aynura
Matricola:234277

Academic year 2021/2022

Index

Summary

The purpose of the following work is to analyze the behavior of a nonlinear system formed by a platform connected to a horizontal axis through a torsional spring and to determine, through the techniques learned on the subject, control laws that make the system stable in an optimal way.

Keeping in mind this goal, the project was divided into two basic phases:

• Modeling, linearization and analysis of structural properties.

• Application of optimal control techniques.

In the first phase, after having modeled the system in a software environment, the non-linear system is linearized around equilibrium from which its stability and structural properties of reachability and observability are analyzed. Then, in the second phase of the project, the optimal control techniques that have been taught on the subject are applied, which are:

• Linear-Quadratic Control (LQ) in discrete time on a finite and infinite time horizon.

• Discrete time LQ tracking problems on a finite time horizon and disturbance rejection with integrator

• Linear Matrix Inequality (LMI) formulation of the LQ control.

To do this, a script was created in Matlab and used the toolbox Simulink to model the system and schemes.

Chapter 1

Modeling, Linearization and Structural Properties

1.1 Mathematical Model

Consider the mechanical system shown in Figure 1.1,

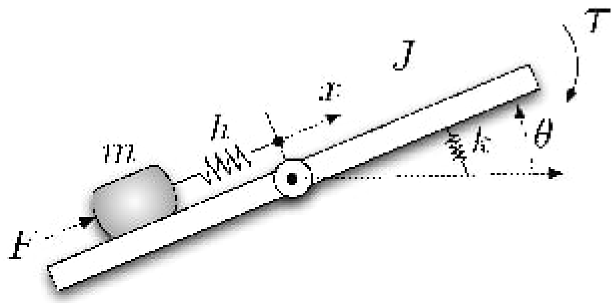


Figure 1.1 Mechanical system

It consists of a platform able to rotate around a horizontal axis on which a mass slide. The angle of rotation of the platform with respect to the horizontal plane is indicated with θ and distance of the mass from the axis of rotation is x. The platform is connected to the rotation axis through a torsional spring with elastic constant k. The mass is free to slide on the platform remaining connected to the rotation axis through spring with elastic constant h and zero rest length. The associated dynamical model is described by the following system of nonlinear equations:

$$m\ddot{x} = F - mg\sin\theta - Ca\dot{x} + mx\dot{\theta}^2 - hx$$
$$\left(J + mx^2\right)\ddot{\theta} = -k\theta - mx\left(g\cos\theta + 2\dot{x}\dot{\theta}\right) - \tau$$

Where J is the moment of inertia associated with the platform, Ca is the coefficient of friction and g is gravitational acceleration.

τ is considered as actuator, while F is regarded as disturbance. The numerical values for parameters are following:

$$m = 0.15kg, J = 750kgm^2, C_a = 1.5\frac{Ns}{m}, g = 9.81\frac{m}{s^2}$$

To represent the system in state space, the following variables are chosen:

$$x = \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \qquad u = \begin{bmatrix} \tau \\ F \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

State-space representation of the model:

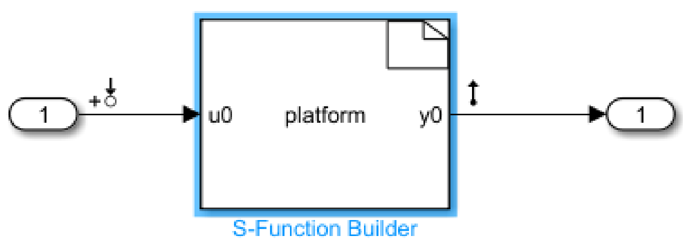
$$\dot{x} = f(x,u) = \begin{bmatrix} x_3 \\ x_4 \\ \dfrac{-mg\sin\theta - C_a x_3 + m x_1 x_4^2 - h x_1 + u_2}{m} \\ \dfrac{-k x_2 - m x_1 (g\cos\theta + 2 x_3 x_4) - u_1}{J + m x_1^2} \end{bmatrix}$$

1.2 Linearization

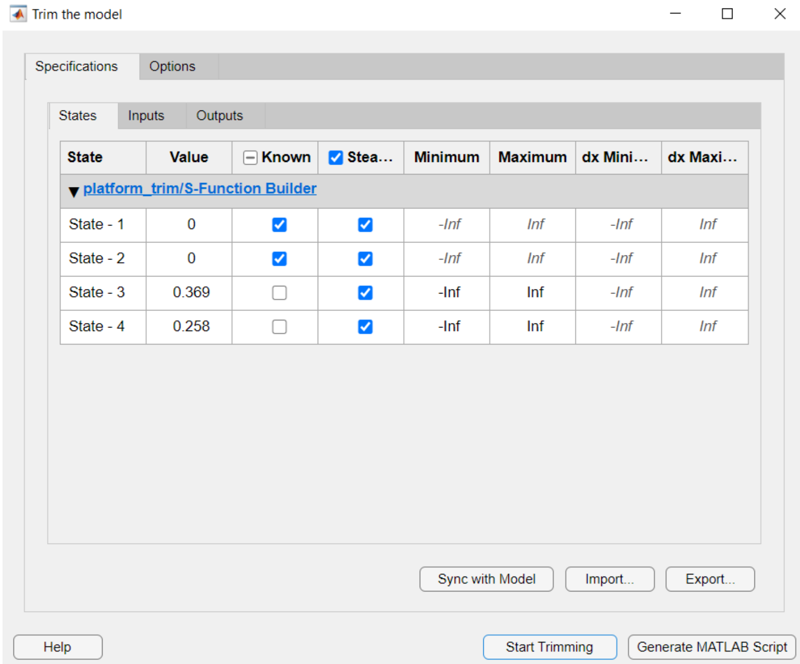
Real dynamical systems are never perfectly linear, but can be approximated to any predetermined movement (such as, for example, an equilibrium point) by means of suitable linear models, called linearized models. One of the main advantages of linearization is that simpler, more powerful and more numerous methodologies are available for the analysis and control of linear dynamic systems than in the non-linear case.

1.2.1 Equilibrium

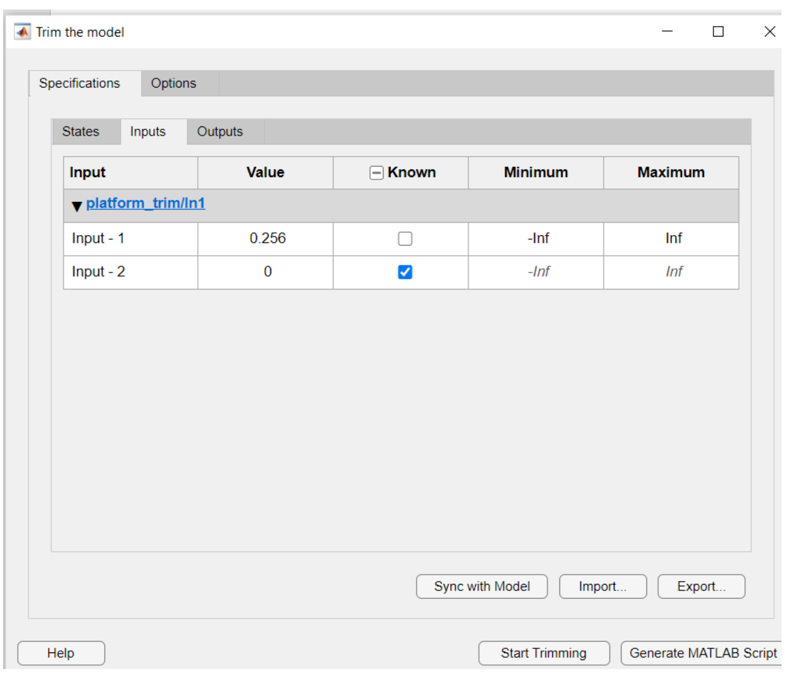
The experimental setup is fixed with respect to $x = 0, \theta = 0$ under the hypothesis that disturbance $F = 0$. We must derive constant torque τ in equilibrium fashion, which is capable of realizing the equilibrium $x = 0, \theta = 0, F = 0$. This task is done using Simulink tool which is called model linearizer.



The values of displacement x and angle θ are kept constant to 0.



The value of disturbance which is the second row of the input matrix is set to 0.



Having trimmed the model, we obtain operating point which contains the equilibrium of the system:

$$x = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, u = 10^{-10} \begin{bmatrix} 0.5215 \\ 0 \end{bmatrix}.$$

1.2.2 Calculation of the linearized system

We linearize the system around the equilibrium utilizing the function *linmod* and obtain the following matrices:

$$Ac = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.666 & -9.809 & -10 & 0 \\ -0.0019 & -0.0066 & 0 & 0 \end{bmatrix}, Bc0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 6.666 \\ -0.0013 & 0 \end{bmatrix},$$

$$Cc = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, Dc0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

We must consider only the input columns related to the actuators, so matrices $Bc0$ and $Dc0$ will be transformed into the following matrices:

$$Bc = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -0.0013 \end{bmatrix}, Dc = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

1.3 Stability

The stability of a system can be thought of as a continuity in its dynamic behavior. If there is a small change in the inputs or initial conditions, a stable system will exhibit small changes in its disturbed response. On the other hand, in an unstable system, any disturbance, no matter how small, will cause states or outputs to grow without limit or until the system burns out, disintegrates or becomes saturated. Therefore, stability is a fundamental requirement of dynamic systems intended to perform process operations or signals, and it is the first thing that must be guaranteed in the design of a control system.
The criterion chosen to verify the stability of the treated system is the Reduced Lyapunov criterion, which in the continuous case is based on the following statements:
1. If the matrix has all stable eigenvalues, that is $Re(\lambda) < 0$, then the equilibrium is locally asymptotically stable.
2. If the matrix has at least one unstable eigenvalue, that is $Re(\lambda) > 0$, then the equilibrium is unstable.
3. If at least one eigenvalue of the matrix is on the imaginary axis, that is $Re(\lambda) = 0$, then the criterion fails to conclude anything.

At this point, to verify the asymptotic stability, it is necessary to calculate the eigenvalues of the matrix Ac on MATLAB using the command *eig* resulting:

$$\lambda(A) = \begin{bmatrix} -9.9329 + 0.0000i \\ -0.0735 + 0.1150i \\ -0.0735 - 0.1150i \\ 0.0800 + 0.0000i \end{bmatrix}$$

Clearly, the equilibrium point is unstable. In this case the controller has to perform two tasks:

1. Fulfill the LQ requirement, which is to regulate the transient and minimize the energy
2. Stabilize the operating point

1.4 Structural Properties

In order to achieve the design goal, a fundamental step is to analyze the structural properties of the dynamic system. In particular, attention must be paid to the reachability and observability of the system. Reachability is an input-state property linked to the control law, that is, to the position of the actuators inside the plant, so as to obtain the desired results from it. While observability is closely linked to the sensor side.

1.4.1 Reachability

The reachability property describes the possibility of modifying the system state starting from a particular pre-established initial state by acting appropriately on the input. Therefore, the concept of reachability is related to the effective possibility of being able to guide a given system, through its inputs, to assume conditions arbitrary. The reachability property is related to matrices A and B, in fact the reachability matrix can be calculated as follows:

$$R = [B \quad AB \quad A^2B \dots \quad A^{n-1}B]$$

In MATLAB it is possible to calculate it through the command *ctrb* which naturally receives matrix as parameters A and B of the linearized system. The reachability matrix of the system in this case is:

$$R = \begin{bmatrix} 0 & 0 & 0 & 0.0131 \\ 0 & -0.0013 & 0 & 0 \\ 0 & 0 & 0.0131 & -0.1308 \\ -0.0013 & 0 & 0 & 0 \end{bmatrix}$$

The reachability condition is as follows:

"A linear system is completely reachable if and only if its reachability matrix has maximum rank, i.e. rank equal to n".

$$rank(R) = rank([B \quad AB \quad A^2B \dots \quad A^{n-1}B]) = 4$$

To check if this happens in the system model, we used the command *rank* on MATLAB and it was obtained that $rank(R) = 4$, which means that it has a full rank, and in this way the condition of complete reachability is satisfied.

1.4.2 Observability

The observability property describes the possibility of estimating the initial state of the system by measuring the output (·) and the entrance (·) over a given time interval, and is linked to matrices A and C, in fact the observability matrix can be calculated as follows:

$$\theta = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

In MATLAB it is calculated by function *obsv* which takes arrays as parameters A and C of the linearized system.
Similar to reachability, the condition of complete observability establishes that a linear system is completely observable if the matrix has full rank, that is, equal to n.

$$rank(\theta) = rank\left(\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ \vdots \\ CA^{n-1} \end{bmatrix}\right) = 4$$

This allows us to state that the system is completely observable.

Chapter 2

Application of Optimal Control Techniques

2.1 Finite Time Horizon Linear Quadratic (LQ) Optimal Control - Discrete Time Case

Considering the linear and stationary discrete time model

$$x_{k+1} = Ax_k + Bu_k$$

with initial state $x(0) = x_0$ known. The sequence of control moves

$$\{u_0, u_1 \dots u_{n-1}\} = u_{[0,N)}$$

such that the cost

$$J(x_0, u_{[0,N)}) = \sum_{k=0}^{N-1} \left(x_k{}^T Q x_k + u_k{}^T R u_k + 2u_k{}^T M x_k\right) + x_N{}^T S x_N$$

said LQ cost, is minimal, and where

1. The penalty function $x_N{}^T S x_N = \psi(x_N)$, denotes a distance measure of some terminal state components with respect to the desired equilibrium 0_x.
2. $x_k{}^T Q x_k + u_k{}^T R u_k + 2u_k{}^T M x_k = l(k, x_k, u_k)$ is called the loss function:
   a. $x_k{}^T Q x_k$, is a quadratic form measuring the distance of the intermediate states x_k with respect to 0_x and makes it possible to evaluate in an indirect fashion, the transient behavior of the regulated state plant (or a part of it);
   b. $u_k{}^T R u_k$, is a quadratic form, measuring the control effort necessary to drive state plant from the initial to the terminal value;
   c. $u_k{}^T M x_k$, is a mixed term representing energy/power transfer phenomena between the plant and the outside environment.

So, the optimal cost value

$$\min_{u_{[0,N)}} J(x_0, u_{[0,N)}) = x_0{}^T P_0 x_0$$

depends on the shaping matrix P_i, positive semidefinite, which is equivalent to the solution of the Riccati difference equation:

$$P_i = \mathrm{Q} + A^T P_{i+1} A - A^T P_{i+1} B (R + B^T P_{i+1} B)^{-1} B^T P_{i+1} A, \qquad 0 \le i \le N-1$$

Then, the optimal control move is a linear feedback of the time-varying state:

$$u_i = -F_i x_i$$

where $F_i = (R + B^T P_{i+1} B)^{-1} B^T P_{i+1} A$ is called Kalman Gain.

To derive the values of F_i and P_i we make use of the so-called Backwards Iterations, calculated in the MATLAB script, in order to do this, we need to choose the weight matrices of the LQ cost: S, Q and R:

1. S is equal to $C_N^T C_N$, where is $C_N \in \mathbb{R}^{l \times n_x}$, denotes a fake output matrix which selects the terminal state vector components we are interested to measure with respect to the equilibrium.

2. Q and R they are obtained starting from a fictitious output

$$z_k = C_z x_k + D_{zu} u_k$$

also called "performance output" with $n_z = n_x + n_u$ components, $Q = C_z^T C_z$ and $R = D_{zu}^T D_{zu}$. The entries of C_z identify which states we are interested to weight during transient; the entries of D_{zu} characterize the way the actuators are used to generate the control effort term by means of the positive parameter ρ; $C_z \in \mathbb{R}^{n_z \times n_x}$ is organized in blocks by rows, which must be orthogonal to the columns of $D_{zu} \in \mathbb{R}^{n_z \times n_u}$ so that there are no mixed terms in the cost (M=0).

Then, we are interested to weight position x and angle θ:

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

$$D_{zu} = \begin{bmatrix} 0 \\ 0 \\ \sqrt{\rho} \end{bmatrix}$$

In this case, the choice of ρ is based on the physical characteristics of the system and its moving components, a very fast transient that could cause structural damage is not desired. In particular, if we want faster transient, it is necessary to change the value of $\sqrt{\rho}$, which influences the efficiency of the actuators that make the control "strong" if R increases or "mild" if R decreases.

At this point it is necessary to find the sampling time and the number of steps of the LQ algorithm.

In order to calculate sampling time, unstable eigenvalue is ruled out and the nearest to imaginary axis eigenvalue is chosen, the natural frequency of which is considered as an estimate of bandwidth of the plant.

$$\omega_{BW} = abs(-0.0735 + 0.1150i) = 0.1365$$

The sampling time must satisfy the following inequality:

$$T_s \leq \frac{2\pi}{20\omega_{BW}}, \qquad T_s \leq 2.3s$$

So, $T_s = 0.5s$ is chosen.

The final time is fixed as = 30.

The number of steps is evaluated as the division between the final time and the sampling time

$$N = \frac{t_f}{T_s}$$

We now proceed to calculate the system discretized by the linear model and the sampling time chosen, which results:

$$A = \begin{bmatrix} 0.9417 & -0.8608 & 0.0948 & -0.3960 \\ -0.0010 & 0.9969 & -0.0001 & 0.9989 \\ -0.0624 & -0.9273 & -0.0063 & -0.8608 \\ -0.0019 & -0.0059 & -0.0002 & 0.9969 \end{bmatrix}, B = \begin{bmatrix} 0.0002 \\ -0.0007 \\ 0.0005 \\ -0.0013 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

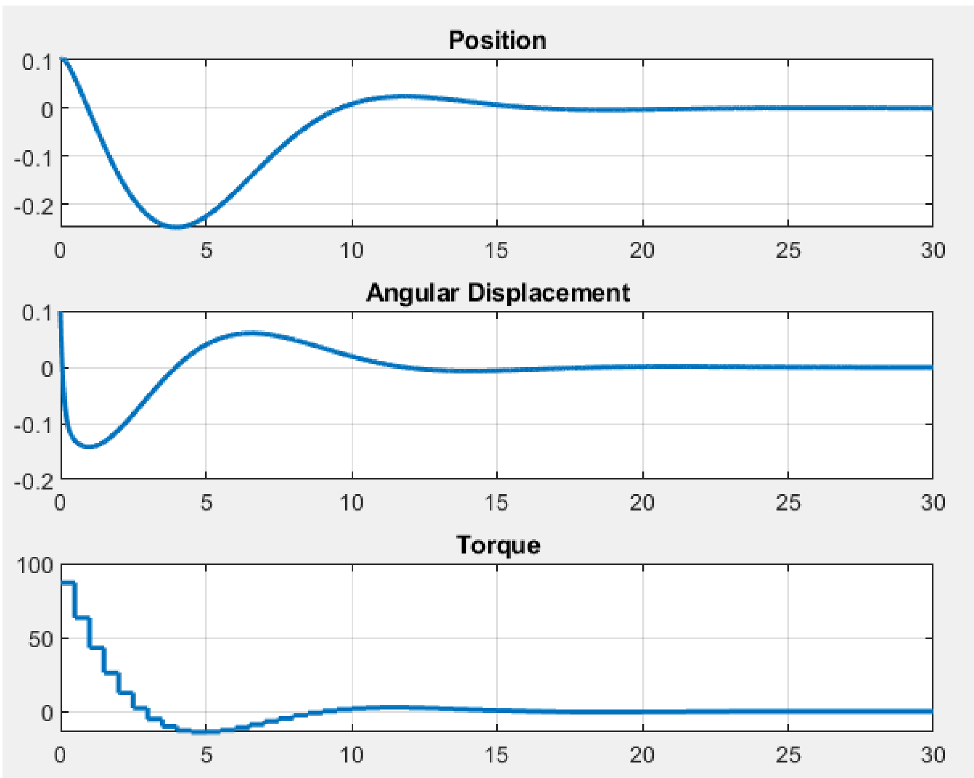


Figure 2.1 Response of the feedback system with finite time horizon LQ

Figure shows the result obtained from the simulation of the system, which states are perturbed by the values dx0 = [0.1;0.1;0.1;0.1]. It is evident that, despite these perturbations, the system returns to its state of equilibrium, which corroborates the accuracy of the parameters calculated for the LQ control.

2.2 Infinite Time Horizon Discrete Time Linear Quadratic Control

For the Linear-Quadratic control discrete time on an infinite time horizon, that is, when N → ∞, the model is considered with the addition of the performance output

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ z_k = C_z x_k + D_{zu} u_k \end{cases}$$

And it's supposed (A, B) is stabilizable and (A,C_z) detectable, then, on the cost, it is assumed S=0:

$$\sum_{k=0}^{\infty} (x_k{}^T Q x_k + u_k{}^T R u_k) = \sum_{k=0}^{\infty} z_k^T z_k$$

So, the law that minimizes it is a stabilizing state feedback:

$$u_{k,\infty} = -F_\infty x_k$$

Where F_∞ depends on P_∞, which is limit for N → ∞ is the solution of the Riccati difference equation. The LQ cost in correspondence of $u_{k,\infty}$ is equal

$$x_0{}^T P x_0$$

On MATLAB, the optimal gain and Riccati solution are calculated with the command *lqr*, then there is no need to recalculate the weight matrices and simply recycle them. Once this is done, the following results are obtained:

$$P_\infty = \begin{bmatrix} 7.0064 & -11.8818 & 0.6038 & -11.7460 \\ -11.8818 & 40.9512 & -1.1950 & 53.0414 \\ 0.6038 & -1.1950 & 0.0607 & -1.1804 \\ -11.7460 & 53.0414 & -1.1804 & 109.5547 \end{bmatrix}$$

$$F_\infty = [60.1336 \ \ -290.2118 \ \ 6.0417 \ \ -645.0078]$$

This constant Kalman gain is equal to the first row of Backward iterations:

$$F_1 = [60.1336 \quad -290.2117 \quad 6.0417 \quad -645.0076]$$

The equivalence is described by the following graph, depicting the same behavior as in case of finite time horizon.

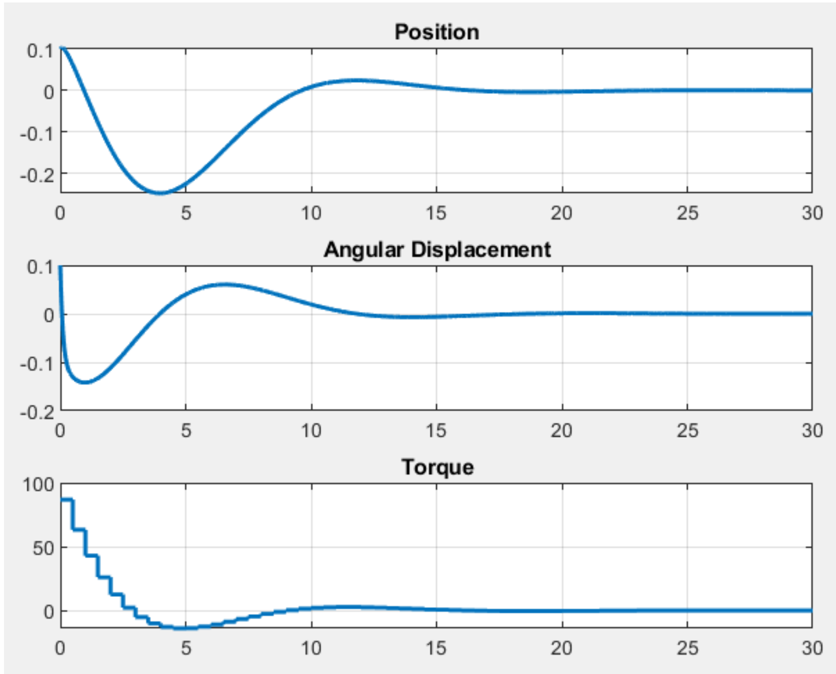


Figure 2.2 Response of the feedback system with infinite time horizon LQ

2.3 Linear-Quadratic tracking problem

Let us consider a discrete time LTI dynamical system endowed with an output (not to be confused with the performance output)

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k$$

an optimal finite time horizon tracking problem consists in the determination of an input sequence $u_{[0,N)}$ such that the performance index

$$\sum_{k=0}^{N-1}((y_k - \mathrm{r}_k)_k^T \mathcal{Q}(y_k - \mathrm{r}_k)_k + u_k^T R u_k) + (y_k - \mathrm{r}_k)_N^T \mathcal{S}(y_k - \mathrm{r}_k)_N$$

is minimum. We will suppose that Q_y >0, R >0, S_y>0. Matrices Q_y are S_y are positive definite since, in a tracking problem, we are interested in weighting all the plant outputs, no exception. The signal $r_{[0,N)}$ is the reference a priori known or available. A tracking problem can be restated, in some sense, as a regulation problem by considering the following variable ε_k = $y_k - \mathrm{r}_k$ which is the tracking error and the performance becomes

$$\sum_{k=0}^{N-1}(\xi_k^T \mathcal{Q}\xi_k + u_k^T R u_k) + \xi_N^T \mathcal{S}\xi_N$$

2.3.1 LQ tracking over finite time horizon

The servo problem

If the reference r_k is obtained as a free response of a dynamical system named exosystem or reference generator

$$x_{r,k+1} = A_r x_{r,k}$$
$$r_k = C_r x_{r,k}$$

which means that the reference is obtained as a linear combination from some free modes. In particular if some of the eigenvalues are located on the unit circle then it is possible to obtain step and/or periodical signals. The tracking performance index can then be written as

$$\sum_{k=0}^{N-1}\left(\begin{bmatrix} x_k^T & x_{r,k}^T \end{bmatrix}\begin{bmatrix} C^T \\ -C_r^T \end{bmatrix} Q_y \begin{bmatrix} C & -C_r \end{bmatrix}\begin{bmatrix} x_k \\ x_{r,k} \end{bmatrix} + u_k^T R u_k\right)$$
$$+ \begin{bmatrix} x_N^T & x_{r,N}^T \end{bmatrix}\begin{bmatrix} C^T \\ -C_r^T \end{bmatrix} S_y \begin{bmatrix} C & -C_r \end{bmatrix}\begin{bmatrix} x_N \\ x_{r,N} \end{bmatrix}$$

and, by defining an augmented state

$$\xi_k = \begin{bmatrix} x_k \\ x_{r,k} \end{bmatrix} \in \mathbb{R}^{n_x + n_r}$$

we obtain a fake LTI model, instrumental for our purposes

$$\xi_{k+1} = \mathcal{A}\xi_k + \mathcal{B}u_k$$

where,

$$\mathcal{A} = \begin{bmatrix} A & 0 \\ 0 & A_r \end{bmatrix}, \mathcal{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

and the performance index becomes an LQ regulation problem

$$\sum_{k=0}^{N-1} (\xi_k^T \mathcal{Q} \xi_k + u_k^T R u_k) + \xi_N^T \mathcal{S} \xi_N$$

with

$$\mathcal{Q} = \begin{bmatrix} C^T \\ -C_r^T \end{bmatrix} Q_y [C \quad -C_r]$$

$$\boldsymbol{\mathcal{S}} = \begin{bmatrix} C^T \\ -C_r^T \end{bmatrix} S_y [C \quad -C_r]$$

So, in the augmented state, Riccati iterations are defined

$$\begin{cases} P_N = S \\ P_k = \mathrm{Q} + A^T P_{k+1} A - A^T P_{k+1} B (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A, \qquad 0 \le k \le N-1 \end{cases}$$

to generate the optimal control moves, which will be a time-varying feedback

$$u_k = -\mathcal{F}_k \xi_k$$

where $\mathcal{F}_k$ is the Kalman gain of the augmented state.

Then the control law can be decomposed into the sum of two components:

$$\mathcal{F}_k = [F_k \quad F_k^v]$$

with $F_k \in \mathbb{R}^{n_u \times n_x}$, $F_k^v \in \mathbb{R}^{n_u \times n_r}$

$$u_k = -\mathcal{F}_k \xi_k = [F_k \quad F_k^v] \begin{bmatrix} \mathrm{x_k} \\ \mathrm{x_{r,k}} \end{bmatrix} = -F_k x_k - F_k^v x_{r,k}$$

the first term is a standard state feedback law (effective regulation) whereas the second, which depends on the reference generator is the feedforward component. The feedback component term is in charge to regulate the time behavior of the tracking error ε_k, $k = 0,1 \ldots N$ with respect to perturbations related to the initial model state x_0 and initial discrepancy between the reference and the output to be tracked. The feedforward terms have the task to fix the shape of the output y_k as near as possible, once the transient on the tracking error is over, to the reference r_k.

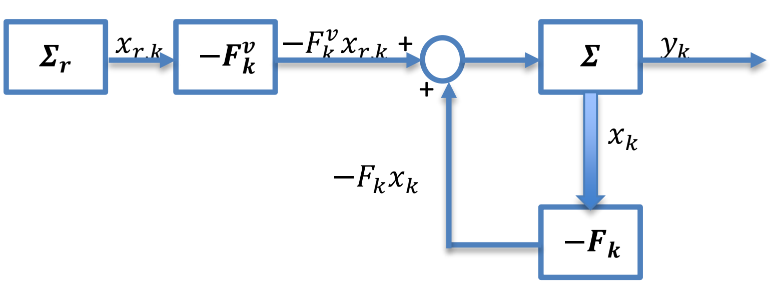


Figure 2.3 Servo problem, feedback and feedforward components

We want to track position x angle θ.

Firstly, we consider an angle and therefore $C_y = [0\ 1\ 0\ 0]$, a reference signal is $x_{r,k} = \frac{\pi}{18}$.

In order to generate the exosystem we start from the continuous time counterpart:

$$\begin{cases} \dot{x}_r(t) = A_{r,c}x_r(t) \\ r(t) = C_y x_r(t) \end{cases}$$

When $A_{r,c} = 0$ we obtain $\mathrm{A_r} = \mathrm{e}^{A_{r,c}T_s} = 1$ and $C_r = 1$.

Feedback and feedforward gains can be derived by considering the structure of the augmented state plant:

$$\mathcal{A} = \begin{bmatrix} A & 0 \\ 0 & A_r \end{bmatrix}, \mathcal{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

$$\mathcal{A} = \begin{bmatrix} 0.9417 & -0.8608 & 0.0948 & -0.3960 & 0 \\ -0.0010 & 0.9969 & -0.0001 & 0.9989 & 0 \\ -0.0624 & -0.9273 & -0.0063 & -0.8608 & 0 \\ -0.0019 & -0.0059 & -0.0002 & 0.9969 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathcal{B} = \begin{bmatrix} 0.0002 \\ -0.0007 \\ 0.0005 \\ -0.0013 \\ 0 \end{bmatrix}$$

and the related Q and S matrices in the performance index:

$$\mathcal{Q} = \begin{bmatrix} C^T \\ -C_r^T \end{bmatrix} Q_y [C \quad -C_r]$$

$$\boldsymbol{\mathcal{S}} = \begin{bmatrix} C^T \\ -C_r^T \end{bmatrix} S_y [C \quad -C_r]$$

The weight matrices of the LQ cost are calculated in the augmented form with Q_y=S_y = 1.

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

Since, the optimal Kalman gain converges towards the constant value, it is possible to implement suboptimal strategy, which is done by taking the first row as F and discarding other rows.

The value of tuning parameter $\rho = 0.0001$ to make the signal as close as possible to reference signal.

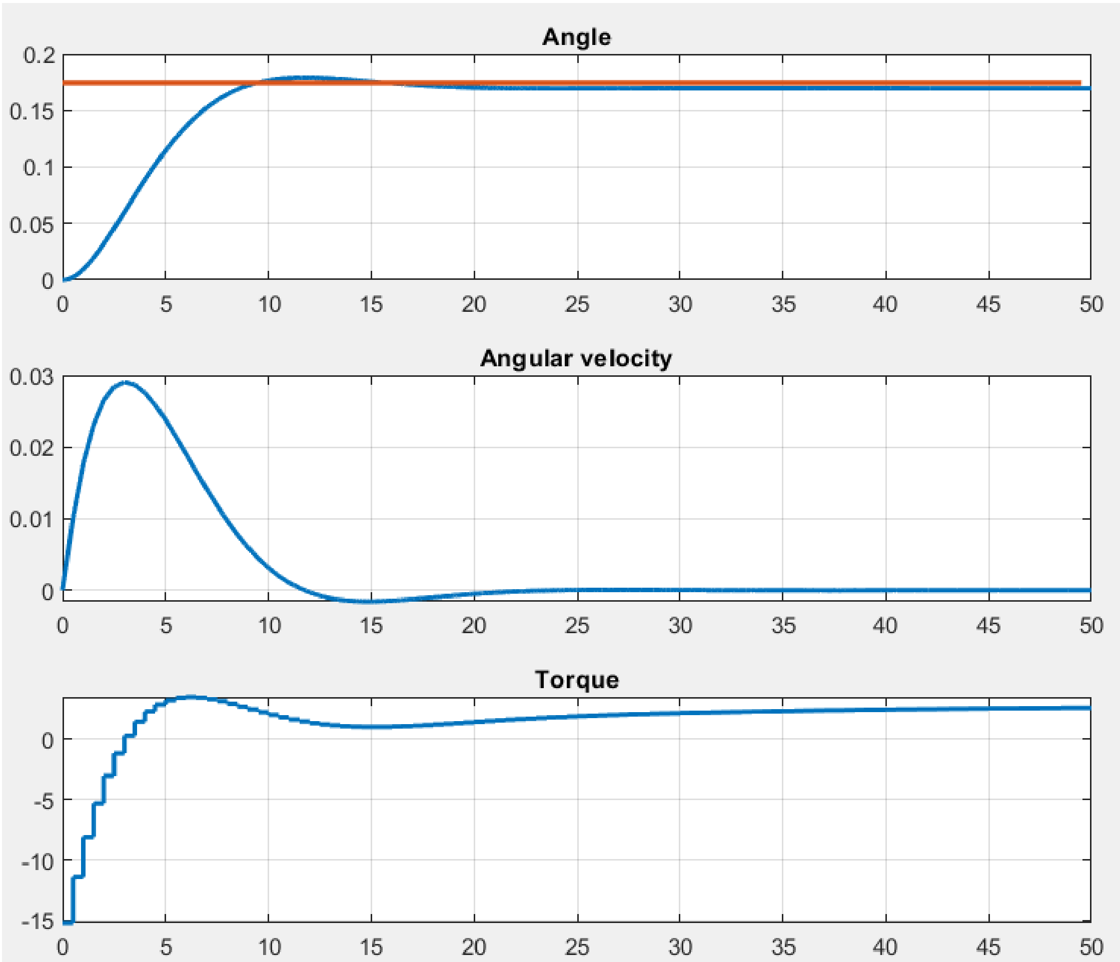


Figure 2.4 Angle tracking

As can be seen from the graph, there is a slight steady-state error.

Next, we track the position $C_y = [\ 1\ 0\ 0\ 0]$ applying the same technique with reference signal $x_{r,k} = 0.1$ and tuning parameter $\rho = 0.01$

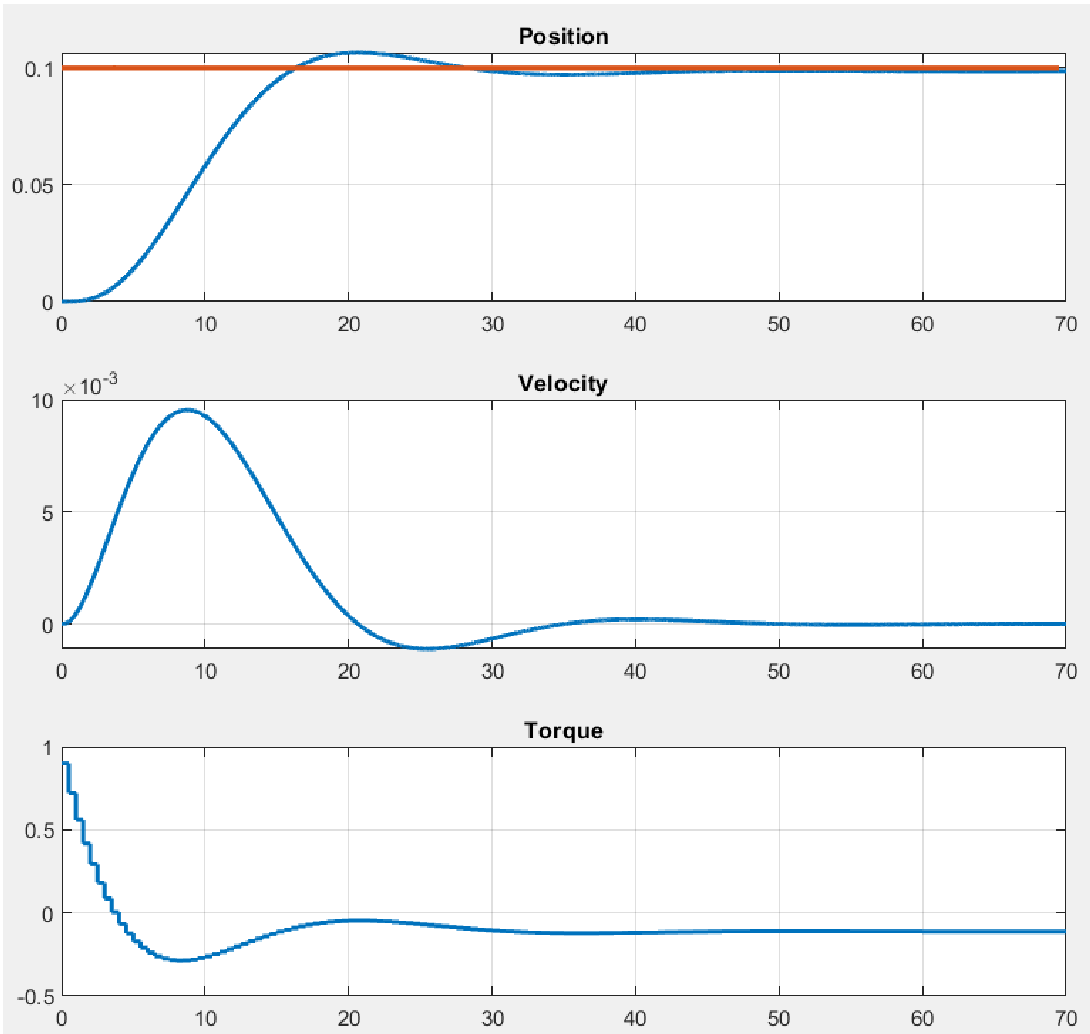


Figure 2.5 Position tracking

A marginal steady-state error on position can be observed.

In both cases we are forced to move tuning parameter to lower value in order to decrease steady-state error, paying price for the voltage whose behaviour can become unacceptable.

2.3.2 LQ tracking with integral control

If the reference signal is a step whose amplitude is r (setpoint tracking) it is possible to design a feedback control strategy such that the tracking error asymptotically goes to zero

$$\lim_{k\to\infty} y_k = r$$

Within a state space approach this can be accomplished by considering the following augmented state space model:

$$\begin{cases} x_{k+1} = Ax_k + Bu_{k-1} + B\Delta u_k \\ u_k = u_{k-1} + \Delta u_k \end{cases}$$

the plant input u_k is obtained by the incremental quantity $u_k = u_{k-1} + \Delta u_k$ which denotes the actuator signal exploited to design the control strategy. The input u_{k-1} becomes an additional state and we will retake the previous section servo approach with $\mathrm{A_r} = 1$. The introduction of the incremental input Δu_k is instrumental to zeroing the tracking error which conversely will be asymptotically finite but not zero. The feedback-feedforward decomposition of the control law holds true and we will obtain:

$$\Delta u_k = -\mathcal{F}_k \xi_k = [F_k \quad F_{u,k} \quad F_k^v] \begin{bmatrix} \mathrm{x_k} \\ u_{k-1} \\ \mathrm{x_{r,k}} \end{bmatrix} = -F_k x_k - F_{u,k} u_{k-1} - F_k^v x_{r,k}$$

Feedback and feedforward gains can be derived by considering the structure of the augmented state plant:

$$A_0 = \begin{bmatrix} A_{00} & B_{00} \\ 0 & I_{n_u} \end{bmatrix} \; B_0 = \begin{bmatrix} B \\ I_{n_u} \end{bmatrix}$$

$$A = \begin{bmatrix} A_0 & 0 \\ 0 & A_r \end{bmatrix}, B = \begin{bmatrix} B_0 \\ 0 \end{bmatrix}$$

and the related Q and S matrices in the performance index:

$$\mathcal{Q} = \begin{bmatrix} C^T \\ 0_{n_u \times n_r} \\ -C_r^T \end{bmatrix} Q_y [C \quad 0_{n_r \times n_u} \quad -C_r], \quad \mathcal{S} = \begin{bmatrix} C^T \\ 0_{n_u \times n_r} \\ -C_r^T \end{bmatrix} S_y [C \quad 0_{n_r \times n_u} \quad -C_r]$$

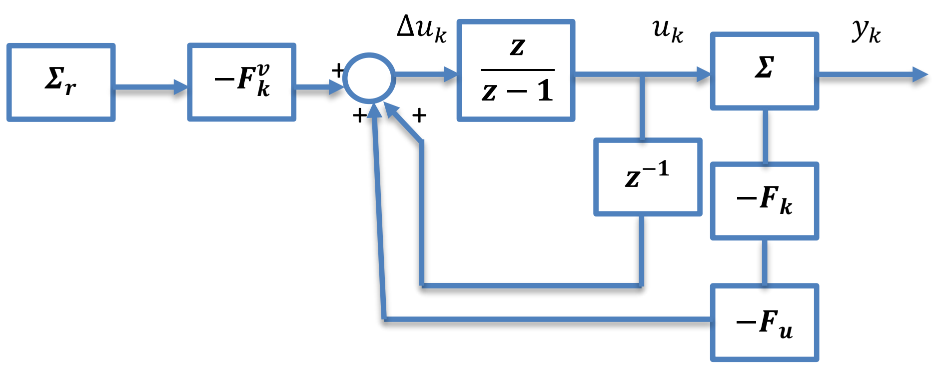


Figure 2.6 Reference tracking with integrator

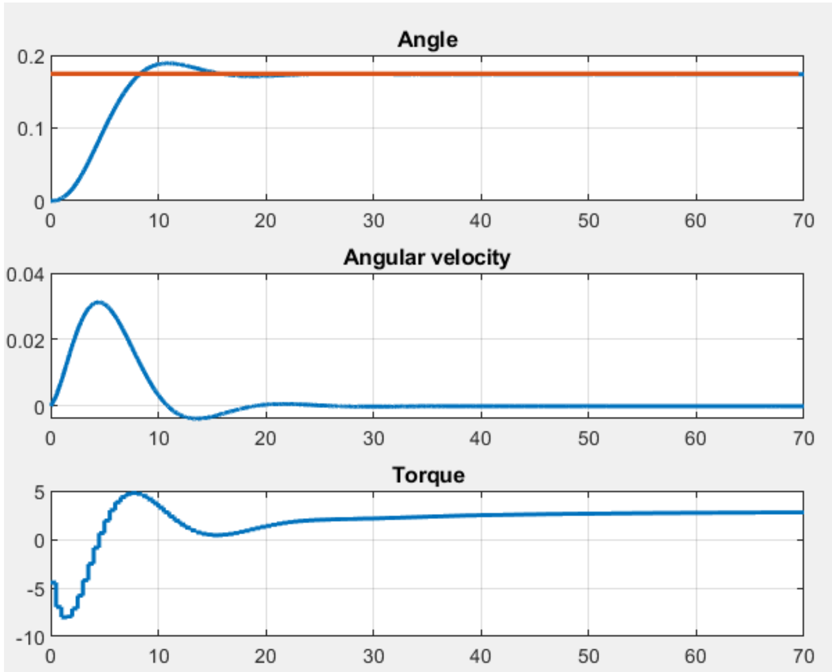


Figure 2.7 Tracking the angle with integrator

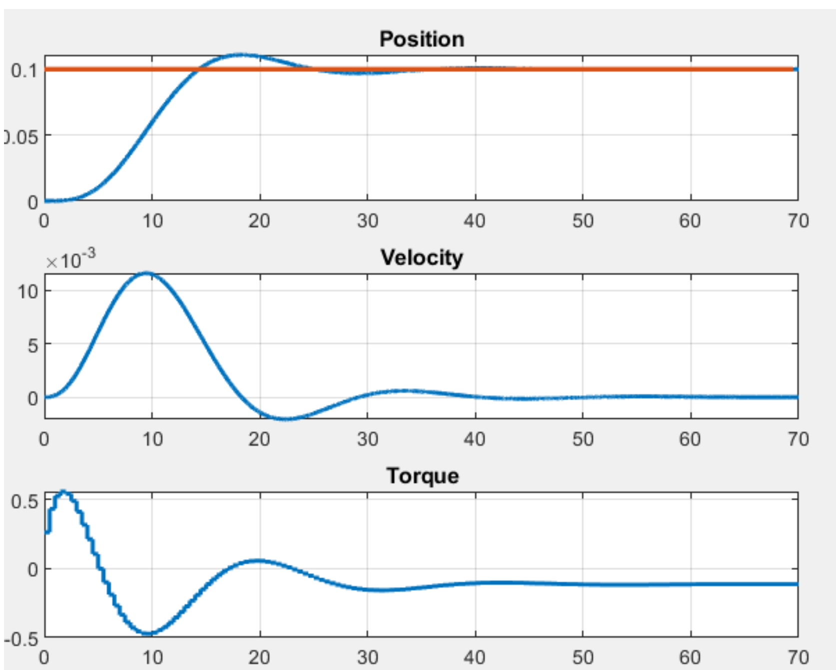


Figure 2.8 Tracking the position with integrator

With this technique, we are able to reach zero steady-state error without tuning ρ to small values. If we tune ρ in this type of strategy, only the transient behaviour is changed.

2.4 LMI formulation of Linear-Quadratic control

Linear Matrix Inequality (LMI) has an important application in control theory by reducing problems such as

1. Asymptotic Stability.
2. Summary of the control law $u = -Fx$
3. Solution of the LQ control problem over an infinite time horizon.

in the computation of the solution to transform it into a convex optimization problem that can be solved numerically through semidefinite programming.
Now we will analyze this formulation in solving LQ optimal control problems.

2.4.1 LMI for discrete time LQ control

Assigned the dynamic system:

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k$$

The goal is to find the law of control:

$$u_k = -F_{opt}x_k$$

such that

$$\sum_{k=0}^{\infty} z_k^T z_k$$

is minimal. Obviously, we must consider the knowledge of the initial state and hypothesize that the system is stabilizable and detectable and the states measurable.
Then, the optimization problem that numerically determines the solution to this LQ control problem on an infinite time horizon in discrete time has the form:

$$\begin{cases} \min\limits_{Y,k,\alpha} \alpha \\ x_0^T Y^{-1} x_0 \leq 1 \\ Y > 0 \\ \begin{bmatrix} Y & YA^T - K^T B^T & YC_z^T - K^T D_{zu}^T \\ AY - BK & Y & 0 \\ C_z Y - D_{zu} K & 0 & \alpha I_{n_z} \end{bmatrix} \geq 0 \end{cases}$$

with $K = FY$, X= αY^{-1} and called slack variable, which is reached after using variable substitutions, Schur's complements and congruence transformations that render the constraints in LMI.

Once the LMI variables are defined α,K,Y as well as the alleys of the upper bound, the positivity on the and Riccati inequality in discrete time, we proceed to optimize using the YALMIP toolbox with the lmilab solver, checking that the problem is feasible and ultimately obtaining the gain:

$$F = KY^{-1} = [46.6940 \ \ -240.8937 \ \ 4.6862 \ \ -577.3760]$$

This result actually coincides with the one obtained through the Riccati iterations in closed form, verifying the equivalence between the two methods.

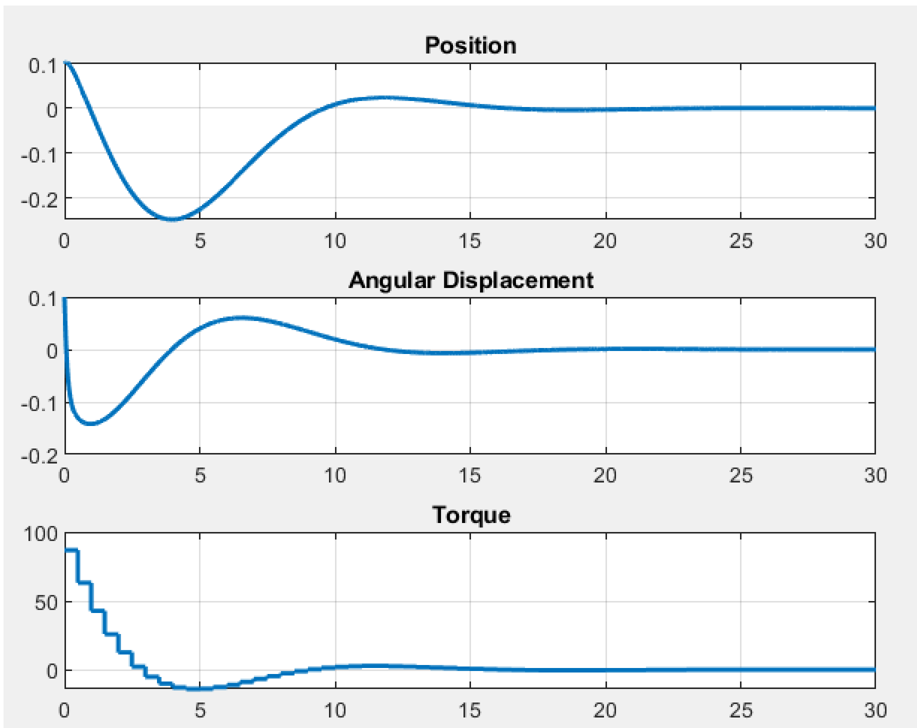


Figure 2.9 Response of the system with LMI

2.5 Constant disturbance rejection with integral control

In this section we want to reject a constant disturbance F_d on the first state variable (position/linear displacement). We have linearized model Σ and a standard state-feedback law is used to find steady state gain $-F$, described by the following diagram

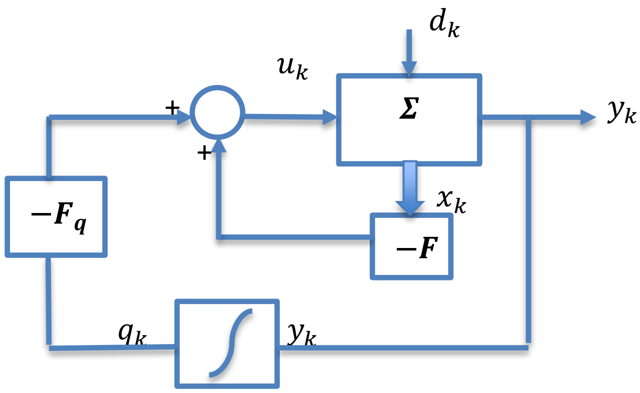


Figure 2.10 Disturbance rejection with integral control

where d_k is discretized version of step with amplitude D.
In general, $\lim_{k\to\infty} y_k = 0$, but due to constant exogenous signal d_k, y_k converges to constant quality, which represents error $y_k \to (y_d \neq 0)$. We want to cope indirectly with this type of error by reducing this quantity as much as possible by performing feedback with integrator such that the following system is asymptotically stable:

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + B_k d_k \\ q_{k+1} = q_k + y_k \end{cases}$$

Control law is: $u_k = -[F \quad F_q] \begin{bmatrix} x_k \\ q_k \end{bmatrix}$, where F_q is component of integrator numerically modifying the control law in order to achieve a disturbance rejection in steady-state condition.
When the transient is over every state and additional variable d_k become constant,

$$q_{k+1} \to \bar{q}$$
$$\bar{q} = \bar{q} + \bar{y}$$
$$\bar{y} = 0$$

which means we are able to obtain desired value $\bar{y} = 0$, rejecting the disturbance.
LMI method is used to implement this task.
Since we want to reject disturbance on linear displacement: $C_y = [\,1\;0\;0\;0]$.
Matrix C_z for LMI is built as follows:

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

The first row represents the output to be monitored and 1 in second row refers to the state of integrator, which is highlighted in order the regulation to be effective.

Initial state is taken as $dx = [0; 0; 0; 0; 0.5]$, where 0.5 is the state of integrator.
After time instant t=15s, we introduce a constant disturbance with amplitude 1.

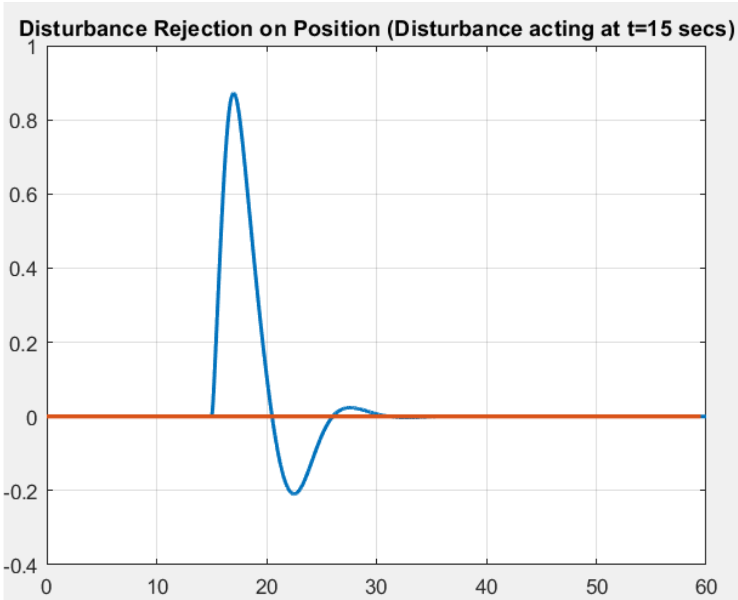


Figure 2.11 Disturbance rejection
As you can see, the integrator was able to recover the original output signal.

2.5.1 Position Tracking and Disturbance Rejection

By a slight modification, it is possible to track the reference. Let's add a constant signal $r = 0.1$ and keep everything from the previous section. The result is following:

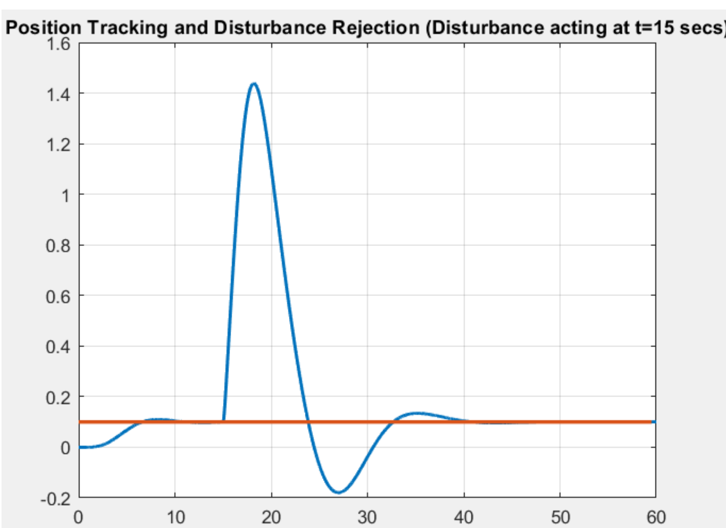


Figure 2.12 Position tracking and disturbance rejection
In this case, integrator successfully rejects the disturbance, recovering the output signal when the transient is over and tracks the given reference.

Conclusion

In the work presented, the different optimal control techniques learned in class were applied to a non-linear system whose mathematical model and parameters are known physical, which is formed by platform able to rotate around a horizontal axis on which a mass slide, where the controlled variable were the displacement of the mass and angle of platform with respect to horizontal axis. To do this, first of all, the model was linearized around given equilibrium point, which makes it possible to use more robust methodologies designed for this type of system, after that, its stability and structural properties were analyzed. Having done this, the methodologies were implemented and the response of the system was analyzed subject to different disorders in the states in which it was possible to verify the effectiveness of algorithms in determining the optimal control moves.

Bibliography

1. Casavola, A. (2021). Notes from the System Theory course. University of Calabria, Department of Computer Engineering, Modeling, Electronics and Systems.
2. Famularo, D. (2022). Notes from the course of Optimal Control. University of Calabria, Department of Computer Engineering, Modeling, Electronics and Systems.