Lecture 5: Optimal Control

Underactuated Robotics

Christoffer Sloth

chsl@mmmi.sdu.dk

SDU Robotics
Mærsk Mc-Kinney Møller Instituttet
Syddansk Universitet

Agenda

Knowledge:

- Derive dynamical state-space models of robots as control systems
- Analyze the stability of low dimensional linear and nonlinear systems
- Analyze the observability and controllability of linear control systems
- Use a variety of controllers for underactuated robots

Skills:

- Implement simulations of control systems in software
- Create concise technical reports presenting solutions to proposed problems

Competencies:

- Choose appropriate modern control techniques to solve control problems in robotics
- Apply modern control techniques to control simulated underactuated robots

Introduction

Course Plan

- **Lesson 1**: Newton-Euler Modelling
- **Lesson 2**: Euler-Lagrange Modelling
- **Lesson 3**: Simulation of Robot Dynamics
- **Lesson 4**: Stability Analysis
- **Lesson 5**: Optimal Control
- **Lesson 6**: Energy Shaping Control
- **Lesson 7**: Feedback Linearisation
- **Lesson 8**: Sliding Mode Control
- **Lesson 9**: Simulation and Implementation of Control Systems
- **Lesson 10**: Optimization-Based Control
- **Lesson 11**:
- **Lesson 12**:

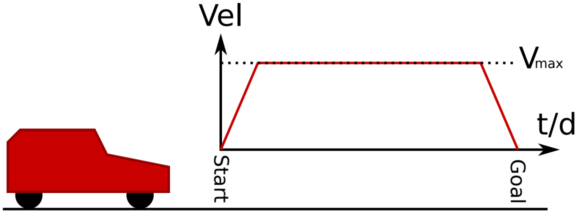
Optimal control is the branch of control which tries to find a control law/input that maximises/minimises some optimality criterion.

Find the velocity profile of a car to go from Odense to Copenhagen in a minimum time, accounting for

- Only one velocity limit (or you get a speeding ticket)
- Acceleration limits (your car engine)
- Acceleration/deceleration limits are the same

The solution is:

1. Maximum acceleration until you reach the velocity limit
2. Maintain velocity to maximum allowed velocity (i.e. zero acceleration if there is no friction)
3. Maximum deceleration when you are arriving



PID control enables to place poles of the closed-loop system.

Pole placement enables to place all the poles of the closed-loop system, but

- Some controllers might not be easy to implement
- Fast response requires good actuators
- Limits on the actions, bounds on the state.
- Sensitivity to system's parameters
- Need full state estimation (effects of noise)

Optimal control enables 'optimal' gain selection

- You must define optimality criterion (maximise or minimise), e.g. time, energy, control effort, error
- You can trade-of different factors (time/energy, error/control effort)
- Optimise as a function of what?
  - Control signal (control program)
  - Controller (e.g. LQR)
- Requires controllability
- Requires full state knowledge (Linear Quadratic Gaussian Controller)

In optimisation problems we need to provide a function to optimise (cost). Given a state evolution $x(t)$ and an input $u(t)$ we can define $L(x, u)$ to minimise

$$\mathcal{J} = h(x(t_f)) + \int_0^{t_f} L(x(t), u(t))dt$$

where $L(x(t), u(t))$ is the cost of being at state $x(t)$ and executing action $u(t)$, $h(x(t_f))$ is the cost of ending at state $x(t_f)$.

In optimisation problems we need to provide a function to optimise (cost). Given a state evolution $x(t)$ and an input $u(t)$ we can define $L(x,u)$ to minimise

$$\mathcal{J} = h(x(t_f)) + \int_0^{t_f} L(x(t), u(t))dt$$

where $L(x(t), u(t))$ is the cost of being at state $x(t)$ and executing action $u(t)$, $h(x(t_f))$ is the cost of ending at state $x(t_f)$.

Problem: Find $x(t)$ and $u(t)$ that minimise the cost (for our system).

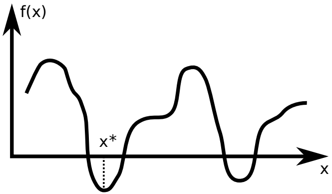
Optimize

$$\mathcal{J} = h(x(t_f)) + \int_0^{t_f} L(x(t), u(t))dt$$

Classes of problems

- Regulation problem: $x(t_f) = 0$ (or $x_e(t_f) = x(t_f) - x_d$)
- Finite vs infinite time optimal control $t_f = \infty$
- Mathematically $\mathcal{J}$ is called functional (a function of a function)
- Related to Calculus of Variations (like Lagrange equations)

Find $x^* = \arg\min_x f(x)$ (i.e. find x^* such that $f(x) > f(x^*)$)



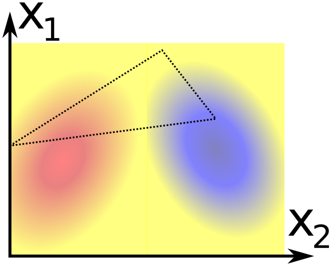
This is a **fundamental** problem.

Many problems can be stated as optimisation problems, e.g. training a neural network, Linear regression, Gaussian Processes.

There are many optimisation algorithms:

- Gradient Based
  - Find local solutions
  - Fast convergence
  - Conjugate Gradient, Quasi-Newton Methods
- Gradient Free
  - Can find global solutions
  - Slow convergence
  - Genetic algorithms, Stochastic annealing, Particle Swam Optimisation

The graph of a cost function is illustrated below, with constraint indicated by triangle.



A constrained optimization problem is given by

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g(x) = 0 \\ & h(x) \geq 0 \end{aligned}$$

A ***Linear Programming Problem*** (LP) can be expressed in the ***standard form***

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \le b \\ & x \in \mathbb{R}^n \\ & c \in \mathbb{R}^n,\ A \in \mathbb{R}^{m \times n},\ b \in \mathbb{R}^m \end{aligned}$$

Solve the following linear programming problem

$$\begin{aligned} \text{minimize} \quad & 225x + 200y \\ \text{subject to} \quad & y \geq 25 \\ & x \geq 40 \\ & x + y \leq 150 \end{aligned}$$

Use YALMIP in MATLAB to formulate the optimization problem. Download it here: https://yalmip.github.io/download/

A ***Quadratic Programming Problem*** (QP) is given by

$$\begin{array}{rl} \text{minimize} & \frac{1}{2}x^T Bx - x^T b \\ \text{subject to} & A_1 x = c \\ & A_2 x \leq d \end{array}$$

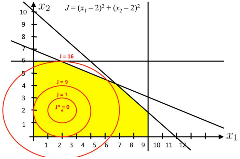
where $B = B^T \in \mathbb{R}^{n\times n}$, $A_1 \in \mathbb{R}^{m\times n}$, $A_2 \in \mathbb{R}^{p\times n}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$, $d \in \mathbb{R}^p$, and the minimization is over the decision variable $x \in \mathbb{R}^n$. The inequality $A_2 x \leq d$ is interpreted componentwise.

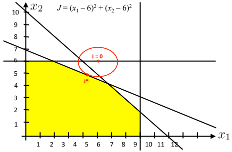
Solve the following optimization problem graphically

$$
\begin{aligned}
\text{minimize} \quad & (x_1 - 2)^2 + (x_2 - 2)^2 \\
\text{subject to} \quad & 2x_1 + 4x_2 \le 28 \\
& 5x_1 + 5x_2 \le 50 \\
& x_1 \le 8 \\
& x_2 \le 6 \\
& x_1 \ge 0 \\
& x_2 \ge 0
\end{aligned}
$$

Solve the following optimization problem graphically

$$
\begin{array}{rl}
\text{minimize} & (x_1 - 2)^2 + (x_2 - 2)^2 \\
\text{subject to} & 2x_1 + 4x_2 \leq 28 \\
 & 5x_1 + 5x_2 \leq 50 \\
 & x_1 \leq 8 \\
 & x_2 \leq 6 \\
 & x_1 \geq 0 \\
 & x_2 \geq 0
\end{array}
$$





Optimizations problems are often nonlinear and given by

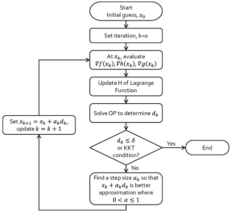
$$
\begin{aligned}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & g(x) = 0 \\
& h(x) \geq 0
\end{aligned}
$$

Can a QP be used for solving such optimization problem?

Sequential Quadratic Programming Problem (SQP) solves an quadratic approximation to the nonlinear problem at each iteration x_k, i.e.

$$\begin{array}{rl} \text{minimize} & f(x_k) + \nabla f(x_k)^T\delta + \frac{1}{2}\delta^T H\mathcal{L}(x_k, \lambda_k, \sigma_k)\delta \\ \text{subject to} & g(x_k) + \nabla g(x_k)^T\delta \geq 0 \\ & h(x_k) + \nabla h(x_k)^T\delta = 0 \end{array}$$

where H is the Hessian of f and $\mathcal{L}$ is the Lagrangian $\mathcal{L} = f(x) - \lambda g(x) - \sigma h(x)$.


Start
Initial guess, x_0
Set iteration, k=o
At x_k, evaluate
$\nabla f(x_k), \nabla h(x_k), \nabla g(x_k)$
Update H of Lagrange Function
Solve QP to determine d_k
$d_k \leq \delta$
or KKT condition?
Yes
End
No
Find a step size α_k so that $x_k + \alpha_k d_k$ is better approximation where $0 < \alpha \leq 1$
Set $x_{k+1} = x_k + \alpha_k d_k$,
update $k = k + 1$

Linear Quadratic Regulation

We consider a linear control system of the form:

$$\begin{array}{rcl} \dot{x} & = & Ax + Bu\,, \quad x(0) = x_0 \\ y & = & Cx \end{array}$$

A control law for such a system is said to be *optimal*, if it minimizes the cost functional:

$$\mathcal{J} = \int_0^\infty x^T Q x + u^T R u \, dt$$

where Q is a positive semi-definite matrix ($Q = Q^T \succeq 0$) and R is a positive definite matrix ($R = R^T \succ 0$).

An *Algebraic Riccati Equation* (ARE) is a second order matrix equation in an indeterminate $P = P^T \in \mathbb{R}^{n \times n}$ of the form:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ are matrices, $R = R^T \in \mathbb{R}^{m \times m}$ is a positive definite matrix, and $Q = Q^T \in \mathbb{R}^{n \times n}$ is a positive semidefinite matrix.

An *Algebraic Riccati Equation* (ARE) is a second order matrix equation in an indeterminate $P = P^T \in \mathbb{R}^{n \times n}$ of the form:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ are matrices, $R = R^T \in \mathbb{R}^{m \times m}$ is a positive definite matrix, and $Q = Q^T \in \mathbb{R}^{n \times n}$ is a positive semidefinite matrix.

The matrix P is called a *stabilizing solution* to the ARE, if it satisfies the equation, and further satisfies that the eigenvalues of $A - BR^{-1}B^T P$ are in the open left half plane.

THEOREM. Consider a linear system of the form:

$$\begin{aligned} \dot{x} &= Ax + Bu\,, \quad x(0) = x_0 \\ y &= Cx \end{aligned}$$

Let P be a stabilizing solution to the ARE:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

Then the optimal state feedback law is given by:

$$u = Fx \quad \text{where} \quad F = -R^{-1}B^T P$$

By introducing $y = Cx$ into a cost functional of the type

$$\mathcal{J} = \int_0^\infty \rho y^T y + u^T u \, dt \,, \quad \rho \in \mathbb{R}$$

the optimal control problem can be written as

$$\begin{aligned} \mathcal{J} &= \int_0^\infty \rho y^T y + u^T u \, dt \\ &= \int_0^\infty \rho x^T C^T C x + u^T u \, dt \\ &= \int_0^\infty x^T Q x + u^T R u \, dt \,, \quad Q = \rho C^T C \,, R = I \end{aligned}$$

Alternatively, use a cost functional of the type

$$\mathcal{J} = \int_0^\infty x^T Q x + u^T R u \, dt$$

where Q and R are diagonal matrices with this can be written as an optimal control problem

$$\begin{aligned} Q_{ii} &= \frac{1}{\text{maximum acceptable value of } x_i^2} \\ R_{jj} &= \frac{1}{\text{maximum acceptable value of } u_j^2}. \end{aligned}$$

We consider once again the system

$$
\begin{aligned}
\dot{x} &= \begin{bmatrix} 2 & -3 \\ 4 & -5 \end{bmatrix} x + \begin{bmatrix} 2 \\ 3 \end{bmatrix} u \\
y &= \begin{bmatrix} -3 & 2 \end{bmatrix} x
\end{aligned}
$$

Computing an optimal state feedback for the cost functional:

$$\mathcal{J} = \int_0^\infty \rho y^T y + u^T u \, dt$$

with $\rho = 800$ can be done with the MATLAB command

```
Fopt = -lqr(A,B,rho*C'*C,1)
```

```
%% System Definition
A = [2 -3;4 -5];
B = [2; 3];
C = [-3 2];
m = size(B,2);
sys = ss(A,B,C,0);
%% Linear Quadratic Regulation
rho = 800;
Fopt = -lqr(sys.A,sys.B,rho*sys.C'*sys.C,eye(m))
```

This yields the result:

$$F_{\text{opt}} = \begin{bmatrix} 69.3536 & -47.8542 \end{bmatrix}$$

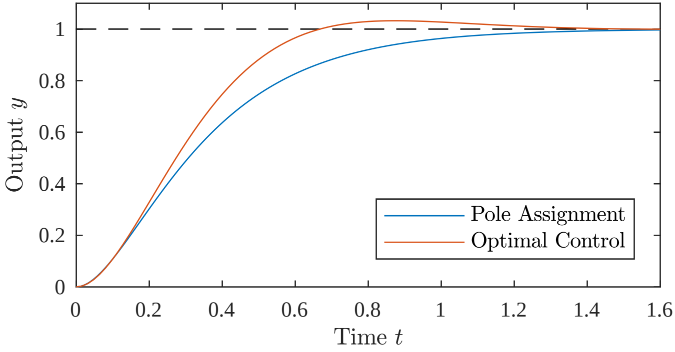
In comparison, a pole assignment with the poles $\{-4, -8\}$ leads to the gain:

$$F = \begin{bmatrix} 72 & -51 \end{bmatrix}$$

A first glance would suggest that the pole assignment with its larger gains would have faster dynamics. However, the optimal feedback assigns complex poles, giving a better rise-time.

Optimal Control

Example (4)



Consider the system model

$$\dot{x} = Ax + Bu$$

where

$$A = \begin{bmatrix} -13 & -6 & 6 \\ -6 & -16 & -5 \\ 6 & -5 & -8 \end{bmatrix}$$

$$B = \begin{bmatrix} -1 & 0 \\ 1 & 1 \\ 2 & 1 \end{bmatrix}$$

Design a linear quadratic regulator for the system, where $Q = I$ and $R = I$.

Consider a linear control system of the form

$$\begin{aligned} \dot{x} &= Ax + Bu\,, \quad x(0) = x_0 \\ y &= Cx \end{aligned}$$

A control law for such a system is said to be *optimal*, if it minimizes the cost functional:

$$\mathcal{J} = x_f^T Q_f x_f + \int_0^{t_f} x^T Q x + u^T R u \; dt$$

where Q and Q_f are positive semi-definite matrices ($Q = Q^T \succeq 0, Q_f = Q_f^T \succeq 0$) and R is a positive definite matrix ($R = R^T \succ 0$).

The continuous-time *Differential Riccati Equation* is a first order differential equation depending on second order matrix expressions in an indeterminate $S(t) = S^T(t) \in \mathbb{R}^{n\times n}$ of the form:

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

with terminal constraint

$$S(t_f) = Q_f$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ are matrices, $R = R^T \in \mathbb{R}^{m\times m}$ is a positive definite matrix, and $Q = Q^T \in \mathbb{R}^{n\times n}$ is a positive semidefinite matrix.

The continuous-time *Differential Riccati Equation* is a first order differential equation depending on second order matrix expressions in an indeterminate $S(t) = S^T(t) \in \mathbb{R}^{n\times n}$ of the form:

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

with terminal constraint

$$S(t_f) = Q_f$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ are matrices, $R = R^T \in \mathbb{R}^{m\times m}$ is a positive definite matrix, and $Q = Q^T \in \mathbb{R}^{n\times n}$ is a positive semidefinite matrix.

The matrix function $S(t)$ is called a *stabilizing solution* to the differential Riccati equation, if it satisfies the equation, and further satisfies that the eigenvalues of $A - BR^{-1}B^T S(t)$ are in the open left half plane for all t.

THEOREM. Consider a linear system of the form:

$$\begin{aligned} \dot{x} &= Ax + Bu\,, \quad x(0) = x_0 \\ y &= Cx \end{aligned}$$

Let $S(t)$ be a stabilizing solution to the differential Riccati equation:

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

Then the optimal state feedback law is given by:

$$u = F(t)x \quad \text{where} \quad F(t) = -R^{-1}B^T S(t)$$

To impose constraints at the final time t_f of the type

$$x(t_f) = x_f$$

the terminal weight Q_f can be set to infinite. This requires a change of variables where $P(t) = S(t)^{-1}$ is used.
One may now find a solution to the differential Riccati equation

$$-\dot{P}(t) = -P(t)A^T - AP(t) + BR^{-1}B^T - P(t)QP(t)$$

with terminal constraint

$$P(t_f) = 0$$

This equation can be solved backwards in time.

Consider a system of the form

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

and a nominal trajectory $(x_d(t), u_d(t))$.
A control law for such a system is said to be *optimal*, if it minimizes the cost functional:

$$\mathcal{J} = (x - x_d(t_f))^T Q_f (x - x_d(t_f)) + \int_0^{t_f} (x - x_d)^T Q (x - x_d) + (u - u_d)^T R (u - u_d) \, dt$$

where Q and Q_f are positive semi-definite matrices ($Q = Q^T \succeq 0, Q_f = Q_f^T \succeq 0$) and R is a positive definite matrix ($R = R^T \succ 0$).

Consider a system of the form

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

and a nominal trajectory $(x_d(t), u_d(t))$.
A linearization of the system about the nominal trajectory is given by

$$\dot{\hat{x}} = \dot{x} - x_d = f(x, u) - f(x_d, u_d)$$

This implies that

$$\begin{aligned}\dot{\hat{x}} &\approx f(x_d, u_d) + \frac{\partial f}{\partial x}\hat{x} + \frac{\partial f}{\partial u}\hat{u} - f(x_d, u_d)\\ &\approx A(t)\hat{x} + B(t)\hat{u}\end{aligned}$$

where

$$A(t) = \frac{\partial f}{\partial x}(x_d, u_d) \qquad B(t) = \frac{\partial f}{\partial u}(x_d, u_d)$$

THEOREM. Consider a time-varying linear system of the form:

$$\begin{aligned} \dot{x} &= A(t)x + B(t)u\,, \quad x(0) = x_0 \\ y &= Cx \end{aligned}$$

Then the optimal tracking state feedback control law is given by:

$$u = u_d(t) - R^{-1}B^T[S_{xx}(t)x + s_x(t)]$$

where

$$\begin{aligned} -\dot{S}_{xx}(t) &= A^T S_{xx}(t) + S_{xx}(t)A - S_{xx}(t)BR^{-1}B^T S_{xx}(t) + Q \\ -\dot{s}_x(t) &= -Qx_d(t) + [A^T - S_{xx}BR^{-1}B^T]s_x(t) + S_{xx}(t)Bu_d(t) \\ -\dot{s}_0(t) &= x_d^T(t)Qx_d(t) - s_x^T(t)BR^{-1}B^T s_x(t) + 2s_x(t)^T Bu_d(t) \end{aligned}$$

with boundary conditions $S_{xx}(t_f) = Q_f$, $s_x(t_f) = -Q_f x_d(t_f)$, $s_0(t_f) = x_d^T(t_f)Q_f x_d(t_f)$.

We consider a discrete-time linear control system of the form:

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k]\,, \quad x[0] = x_0 \\ y[k] &= Cx[k] \end{aligned}$$

A control law for such a system is said to be *optimal*, if it minimizes the cost functional:

$$\mathcal{J} = \sum_{k=0}^{N-1} x^T[k]Qx[k] + u^T[k]Ru[k]$$

where Q is a positive semi-definite matrix ($Q = Q^T \succeq 0$) and R is a positive definite matrix ($R = R^T \succ 0$).

The *Riccati Difference Equation* is a second order matrix equation in an indeterminate $P = P^T \in \mathbb{R}^{n\times n}$ of the form:

$$A^T S[k] A - (A^T S[k] B)(R + B^T S[n] B)^{-1} (A^T S[k] B)^T + Q = S[k-1]$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ are matrices, $R = R^T \in \mathbb{R}^{m\times m}$ is a positive definite matrix, and $Q = Q^T \in \mathbb{R}^{n\times n}$ is a positive semidefinite matrix.

The *Riccati Difference Equation* is a second order matrix equation in an indeterminate $P = P^T \in \mathbb{R}^{n \times n}$ of the form:

$$A^T S[k] A - (A^T S[k] B)(R + B^T S[n] B)^{-1}(A^T S[k] B)^T + Q = S[k-1]$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ are matrices, $R = R^T \in \mathbb{R}^{m \times m}$ is a positive definite matrix, and $Q = Q^T \in \mathbb{R}^{n \times n}$ is a positive semidefinite matrix.

The matrix S is a *fixed-point* of the Riccati difference equation if $S[k] = S[k-1]$, i.e.

$$A^T S A - (A^T S B)(R + B^T S B)^{-1}(A^T S B)^T + Q = S$$

THEOREM. Consider a linear system of the form:

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k]\,, \quad x[0] = x_0 \\ y[k] &= Cx[k] \end{aligned}$$

Let S be a stabilizing solution to the DARE:

$$A^T SA - (A^T SB)(R + B^T SB)^{-1}(A^T SB)^T + Q = S, \qquad S[N] = 0$$

Then the optimal state feedback law (LQR) is given by:

$$u[k] = F[k]x[k] \quad \text{where} \quad F = (R + B^T S[k]B)^{-1} B^T S[k]Ax[k]$$

The optimal control problem in finite time for linear systems can be discretised (over time) and solved as an optimisation problem.

Given an initial state x_0 the actions $u(t)$ $(u(t_k))$, we can determine the whole state space trajectory.

The cost function becomes a function of $u(t)$ $(u(t_k))$ and the initial position x_0.

New problem: Find $u(t_k)$ that optimises the cost function $\mathcal{J}$.

Discretized Optimal Control

Reformulation of Optimal Control Problem

Consider a discrete time linear state space model

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

Discretized Optimal Control
Reformulation of Optimal Control Problem

Consider a discrete time linear state space model

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

Problem: Find a control sequence u_k $k = 0, 1, \ldots, N-1$ that drives the system state from x_0 to x_N in N steps while minimizing the cost function

$$\mathcal{J} = \frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right)$$

Note that we can write

$$
\begin{aligned}
x_1 &= Ax_0 + Bu_0 \\
x_2 &= A^2 x_0 + ABu_0 + Bu_1 \\
x_3 &= A^3 x_0 + A^2 Bu_0 + ABu_1 + Bu_2 \\
&\vdots
\end{aligned}
$$

$$
x_N = A^N x_0 + \sum_{i=0}^{N-1} A^i B u_{N-1-i}
$$

Discretized Optimal Control
Time Evolution of State (Matrix Form)

Define vectors $X = [x_1, x_2, \ldots, x_N]^T$ and $U = [u_0, u_1, \ldots, u_{N-1}]^T$ and define matrices

$$\Gamma = \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & A^{N-3}B & \cdots & B \end{bmatrix}, \qquad \Omega = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}$$

Then we can write

$$X = \Omega X_0 + \Gamma U$$

where $X_0 = [x_0, x_0, \ldots, x_0]^T$.

The quadratic cost function from LQR can be written as

$$\mathcal{J} = b + U^T C + \frac{1}{2} U^T A U$$

where

$$\begin{aligned} b &= \frac{1}{2}(X_0 \Omega)^T \boldsymbol{Q} \Omega X_0 \\ C &= \Gamma^T \boldsymbol{Q} \Omega X_0 \\ A &= \Gamma^T \boldsymbol{Q} \Gamma + \boldsymbol{R} \end{aligned}$$

and $\boldsymbol{Q} = \text{diag}(Q, Q, \ldots, Q)$, $\boldsymbol{R} = \text{diag}(R, R, \ldots, R)$.

Download OptimTraj – Trajectory Optimization Library that is available here (or via search in Add-On Explorer): https://se.mathworks.com/matlabcentral/fileexchange/54386-optimtraj-trajectory-optimization-library

Run the demo on the acrobot and do the following modifications

- Change the cost function to include the angular velocities of the joints (try different weights and see their effect).
- Change the final time to 4 s.
- Change the torque limit to 10 Nm.
- Change the torque limit to 10 Nm and use final time of 2 s.
- Change the torque limit to 20 Nm and change the objective to be minimum time.