MCE/EEC 647/747: Robot Dynamics and Control

Lecture 7: Overview of State-Space Methods

Reading: SHV Chapter 6

Mechanical Engineering

Hanz Richter, PhD

General Definition of Control System

A contol system can be understood in its more general form as a collection (T, X, U, ϕ), where

- T is the time set (indexing the transition of states)

- X is the state space (where the states belong)

- U is the control input space

- ϕ is the transition mapping, a rule giving new states as a function of time, old states and input.

General Definition of Control System...

Example: mass m pushed by a force $[F_x, F_y]^T$ in the $x - y$ plane. Here, the time set is $T = \mathbb{R}$ (it could be further restricted to non-negative real numbers), the state space is $\mathbb{R}^4$, the control input space is the plane (real vectors with two components) and the transition mapping can be found by integrating Newton's Law in each dimension:

$$
\begin{aligned}
x(t) &= x_0 + \dot{x}_0 t + \int_0^t \int_0^t F_x(\tau) d\tau \\
y(t) &= y_0 + \dot{y}_0 t + \int_0^t \int_0^t F_y(\tau) d\tau
\end{aligned}
$$

The transition mapping is

$$[x(t), \dot{x}(t), y(t), \dot{y}(t)]^T = \phi(t, [x_0, \dot{x}_0, y_0, \dot{y}_0]^T, [F_x(t), F_y(t)]^T)$$

a function defined on $[0, \infty) \times \mathbb{R}^4 \times \mathbb{R}^2$.

General Definition of Control System...

A number of properties are attached to the definition. For details, see Sontag, Mathematical Control Theory.

For a second example, consider a savings account with interest rate i. Here, the time set is $\mathbb{N}$ (natural numbers 0,1,2....). The state space is $\mathbb{R}$ (including the negatives to represent overdrafting), the control input space is also $\mathbb{R}$ (including the negatives to represent withdrawals). The transition mapping would be

$$x(k) = \phi(k, x(0), u(k)) = (i+1)^k x(0) + (i+1)^{k-1} u(0) + (i+1)^{k-2} u(1) + ... + u(k-1)$$

General Definition of Control System...

A third example: consider a wedge controlled by a force, as in the figure.
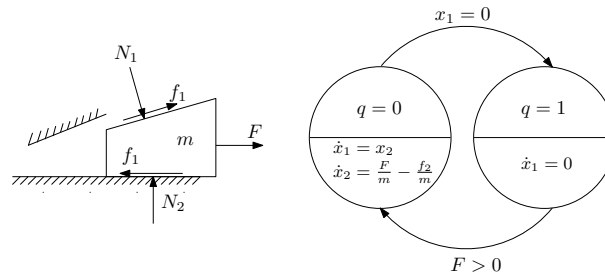
This is an example of a *hybrid dynamic system*, because the states are a mix of continuous variables and discrete variables. The time set is still $\mathbb{R}$. The state space is $\{0, 1\} \times \mathbb{R}$. The input space is $\mathbb{R}$ and the transition mapping can be represented in the chart of the figure. Can you write an expression for $\phi(t, [q_0, x_0, \dot{x}_0]^T, F(t))$?

Systems of Nonlinear O.D.E.s as Dynamic Systems

A system of nonlinear first-order ordinary differential equations has the form:

$$
\begin{aligned}
\dot{x}_1 &= f_1(x_1, x_2, ..., x_n, u_1, u_2, ...u_n, t) \\
\dot{x}_2 &= f_2(x_1, x_2, ..., x_n, u_1, u_2, ...u_n, t) \\
&\vdots \\
\dot{x}_n &= f_n(x_1, x_2, ..., x_n, u_1, u_2, ...u_n, t)
\end{aligned}
$$

If we define $x \triangleq [x_1, x_2, ...x_n]^T$ (state vector), $u \triangleq [u_1, u_2, ..., u_m]^T$ (control vector) and $f \triangleq [f_1(.), f_2(.), ..., f_n(.)]^T$ (vector function), we can simply write

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

Often, systems are *autonomous*, that is, f is not an explicit function of time. Then we write $\dot{x} = f(x, u)$.

Systems of Linear O.D.E.s as Dynamic Systems

In a system of linear first-order ordinary differential equations, the f_i have the form

$$f_i(x, u) = a_{i1}x_1 + a_{i2}x_2 + ...a_{in}x_n + b_{i1}u_1 + b_{i2}u_2 + ...b_{im}u_m$$

This can be conveniently represented in matrix form as

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

where the state vector is n-by-1, the input vector is m-by-1, $A = a_{ij}$ is n-by-n and $B = b_{ij}$ is n-by-m. When viewed as a dynamic system, the transition mapping of the above control system is

$$x(t) = e^{A(t-t_0)}x(0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau$$

Special methods (EEC510,MCE603/703) are needed to compute the matrix exponentials.

Poles and Zeroes of Linear State-Space Systems

Sometimes a set of outputs is (arbitrarily) defined for a state-space system. In the case of linear systems, we write

$$y = Cx + Du$$

where y is a p-by-1 vector, C is p-by-n and D (called *direct transmission term*) is p-by-m. As shown in basic courses, the input-output transfer function corresponding to the state-space system between u and y is

$$G(s) = C(sI - A)^{-1}B + D$$

Note that when $m > 1$ or $p > 1$ a *transfer matrix* is obtained which contains mp individual transfer functions.

It can be shown that the poles of $G(s)$ are the same as the eigenvalues of A.

Also, the zeroes of $G(s)$ can be found from (A, B, C, D). In this course, we use

Matlab for both needs: `eig(A)` and `tzero(A,B,C,D)`

Overview Topics

For the purposes of this section of the course, we need to be able to:

1. Linearize a nonlinear system about a point to obtain a linearized system in state-space form (A, B, C, D)

2. Convert between state-space and transfer function and viceversa

3. Determine the controllability/observability and open loop stability of a linear system

4. Obtain linear feedback gains to stabilize a linear system: pole placement and optimal methods

Fortunately, all of the above, except for 1) can be reduced to one or a few

Matlab commands.

Linearization - Equilibrium Points

An *equilibrium point* or *fixed point* of a control system under constant control u_0 is defined by the property

$$\dot{x} = f(x, u_0) = 0$$

In nonlinear systems, multiple equilibrium points can exist. For example, the system

$$\dot{x}_1 = x_1^2 - u$$
$$\dot{x}_2 = x_1 x_2^2 - x_2 u$$

has 4 equilibrium points for $u_0 = 1$. **Find them.**

We may linearize a nonlinear system about an equilibrium point by using

the Jacobian of f. In doing so, we obtain a linear state-space system for

state and control deviations: $\dot{\Delta}x = A\Delta x + B\Delta u$.

Linearization - Equilibrium Points

As an example, we linearize the above system about $u_0 = 1$ and $x_0 = [1, 0]^T$

$$\dot{\Delta} x_1 = \left.\frac{\partial f_1}{\partial x_1}\right|_{u_0,x_0} \Delta x_1 + \left.\frac{\partial f_1}{\partial x_2}\right|_{u_0,x_0} \Delta x_2 + \left.\frac{\partial f_1}{\partial u}\right|_{u_0,x_0} \Delta u$$

$$\dot{\Delta} x_2 = \left.\frac{\partial f_2}{\partial x_1}\right|_{u_0,x_0} \Delta x_1 + \left.\frac{\partial f_2}{\partial x_2}\right|_{u_0,x_0} \Delta x_2 + \left.\frac{\partial f_2}{\partial u}\right|_{u_0,x_0} \Delta u$$

The matrix A is obtained by taking all partial derivatives of f with respect to all variables ($A = \{a_{ij}\} = \frac{\partial f_i}{\partial x_j}$), evaluated at (u_0, x_0). The matrix B is obtained by taking partial derivatives w.r.t. all control components: $B = \{b_i\} = \frac{\partial f_i}{\partial u_i}$.

Note that the linearized system is valid only in close proximity to (u_0, x_0).

State-Space/Transfer Function Conversions

it is sufficient to test for $x_f = 0$. A Since a state-space system has a well-defined input/output relationship, a unique transfer function corresponds to the state-space system. To obtain it, we use the Laplace-based formula:

$$G(s) = C(sI - A)^{-1}B + D$$

In Matlab, we use the old syntax: `ss2tf` or the newer object-oriented one: `tf(my_ss_sys)`
The opposite conversion does not have a unique answer, since states can be defined in multiple ways. For example, the second-order system

$$G(s) = \frac{1}{s^2 + as + b} = \frac{Y(s)}{U(s)}$$

corresponds to the differential equation

$$\ddot{y} + a\dot{y} + by = u$$

State-Space/Transfer Function Conversions...

If we choose the states as $x_1 \triangleq y, \; x_2 =\triangleq \dot{y}$ we obtain certain (A, B, C, D) matrices. If we make another choice, for instance $x_1 \triangleq y + \dot{y}, \; x_2 \triangleq y - \dot{y}$, we obtain different matrices. Of course, if we go back to transfer function, there's only one answer. To obtain a state-space realization in Matlab, use `tf2ss` (old way) or `ss(my_tf_sys)` (newer way).

We illustrate these conversions using our linearized system.

Controllability

A dynamic system is *controllable* if given any two points x_0 and x_f and an initial time t_0 there exists a control input function $u(t)$ and a finite final time t_f so that the state is transferred from x_0 at t_0 to x_f at time t_f under the action of u. In terms of the transition mapping:

$$x_f = \phi(t_0, x_0, u(t))$$

If the system is linear,a simple computational test can be done using (A, B). The system is controllable if

$$\text{rank } [B|AB|A^2 B|...A^{n-1} B] = n$$

The matrix in brackets is known as the *controllability matrix*. In Matlab simply type `rank(ctrb(A,B))`

Is our linearized system controllable?

Observability

A dynamic system is *observable* if the initial state x_0 can be determined by observing the output y for a finite period of time. If the system is linear, a simple computational test can be done using (A, C). The system is observable if

$$\text{rank } [C^T | (CA)^T | (CA^2)^T | .... (CA^{n-1})^T]^T = n$$

The matrix in brackets is known as the *observability matrix*. In Matlab simply type `rank(obsv(A,C))`

Is our linearized system controllable w.r.t the output defined by $C = [10]$?

Stability

The stability of a linear system can be judged by inspecting the eigenvalues of A. In the next chapter, a more detailed treatment of stability will be offered.

Is our linearized system stable? Find the eigenvalues and determine the poles of the transfer function to check that they are the same.

Regulation vs. Tracking

From the state-space point of view, control problems can be divided in two: regulation of the state to an equilibrium point or output tracking.

In the regulation problem, we wish to determine a feedback control law that will keep the state as close as possible to a fixed point (usually zero) despite external disturbances and model errors.

In the tracking problem, the controller must be designed so that the output follows a reference trajectory as close as possible, despite external disturbances and model errors.

In linear systems, both problems can be tackled using *linear state feedback*, which is a control law of the form

$$u = Pr - Kx$$

where P is a feedforward term and K is the *state feedback gain*. For regulation, we take $P = 0$.

Regulation by Linear State Feedback

When $u = -Kx$, we obtain the closed-loop system:

$$\dot{x} = (A - BK)x$$

This system will be stable if $(A - BK)$ is Hurwitz (eigenvalues in the l.h.p.). The design problem is then to find K so that the closed-loop matrix is Hurwitz.

Two main methods exist: *pole placement* (incl. *eigenstructure assignment*) and *linear quadratic optimal control*. In the first, the designer selects desirable pole locations. Pole locations are relatively simple to decide only for low-order systems. Beyond third order, one doesn't know where to place the poles.

Poles may be placed arbitrarily only when the system is controllable. In this case, we use Matlab's `place` function.

As an example, take the linearized system and find a state-feedback gain achieving closed-loop poles with a frequency of 12 and a damping ratio of 0.75.

Regulation by Linear State Feedback....

A better way to determine the feedback gain is to use linear quadratic regulator (LQR) theory. Here, given the system (A, B), we find the control function u that minimizes a performance index

$$J = \int_0^\infty (x^T Q x + u^T R u) dt$$

The solution to this problem is precisely to use a linear state feedback control of the form $u = -Kx$ where K is obtained from the following *algebraic Riccati equation (ARE):*

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

taking $K = R^{-1}B^T P$, where P is a symmetric, *positive-definite* matrix which solves the ARE. In this course, we use Matlab's `lqr` command to generate the desired feedback gain.

Example

We take our linearized system and design a linear quadratic regulator. Then we illustrate the implementation of the nonlinear system, the linearized system and the feedback control in Simulink. We observe that the controller fails when the system is operated "too far" from the linearization point.

As a homework project, you will take all of the above steps with another nonlinear system.