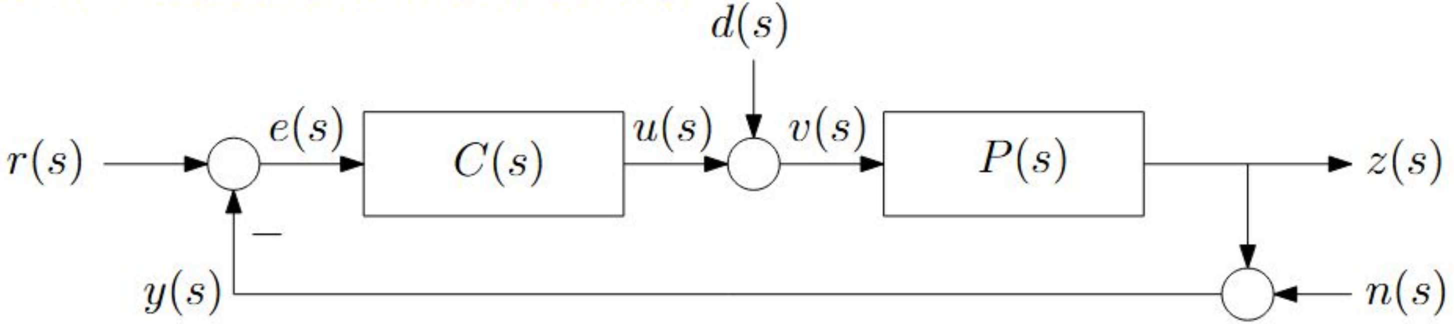
McGill Rocket Team

LQR Control – Background

Patrice Légaré

Control Review

1DOF Feedback Interconnection



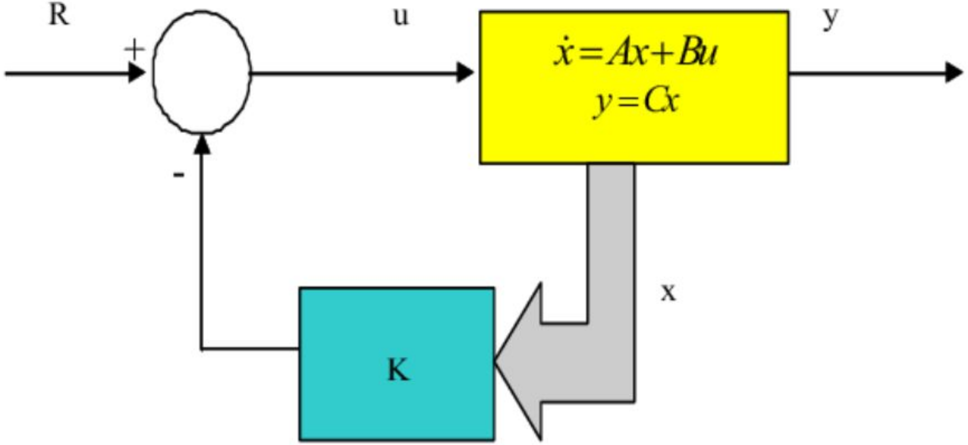
- $r(s)$: command (a.k.a. reference).
- $n(s)$: measurement noise.
- $d(s)$: disturbance.
- $P(s)$ or $z(s) = P(s)v(s)$: plant TF (a.k.a. process TF).
  - $z(s)$: plant output.
  - $v(s) = d(s) + u(s)$: plant input.
- $C(s)$ or $u(s) = C(s)e(s)$: controller (a.k.a. compensator) to be designed.
  - $u(s)$: controller output.
  - $e(s) = r(s) - y(s) = r(s) - z(s) - n(s)$: control input.
  - $y(s)$: measurement.

Satellite Model

State column matrix (x) : position, velocity, angular rate, etc.

Actuator column matrix (u): magnetorquer

LQR Specific Overview



- Yellow block is our satellite model
- LQR is the K matrix
- K is removed from reference R and fed back to model

LQR Optimization

Posing the Control problem as an optimization problem to find where to place our poles (OLHP).In control theory, we are always trading-off between performance and robustness. We therefore set up a cost function to weight each of these criterias.

penalize bad performance penalize actuator effort

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

x is the state vector (what we are measuring), ex:angular position, velocity

Q is nxn to match dimension of state. Positive-semidefinite matrix

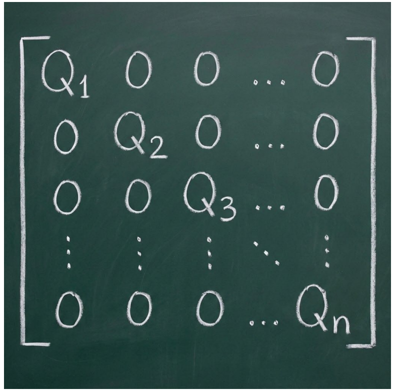
R matches dimension of input. Positive definite matrix. If we have one input (ex: 1 thruster), R is a scalar.

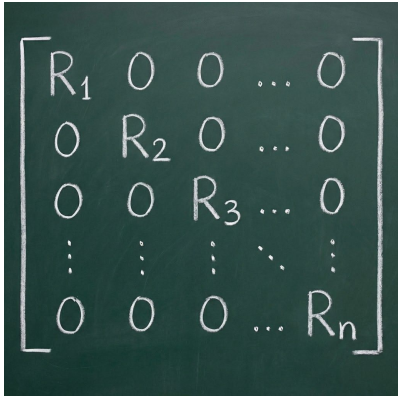
LQR Optimization

R and Q weighted matrices are diagonal matrices.

R will penalize high efforts in the magnetorquer which will use up a lot of the battery. (Larger R means the satellite will smoothly slew to its desired position and therefore be more energy efficient)

Q will penalize large amounts of time spent NOT at our reference. (Larger Q means the satellite will move more aggressively to get to desired position)





The difference between Q1, Q2, etc is based on which state they are mapped to.

Example: I want to prioritize accuracy of the angular position of my spacecraft over any other state. If we chose angular position to be x1 and z position to be x2, I should choose Q1 to be larger than Q2,Q3,etc in order to penalize errors in angular position heavily

**R and Q are not necessarily the same dimension

Expanded Representation Cost Function

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} Q_1 & & & 0 \\ & Q_2 & & \\ & & \ddots & \\ 0 & & & Q_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix} \begin{bmatrix} R_1 & & & 0 \\ & R_2 & & \\ & & \ddots & \\ 0 & & & R_m \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}$$

Q1: How much does it cost to have errors in my state x1

R1: How much does it cost to use and actuate the controller u1

Implementation - MATLAB

1) Once we have a linearized model of our satellite in the form:

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

2) We can adjust the diagonal terms of Q and R based on our design decisions. A good starting point is Identity for both matrices.
3) In MATLAB, use the function *k=lqr(A,B,Q,R)* to find the optimal gains.
4) Simulate the system with *sys = ss((A - B*K), B, C, D)*
5) if changes need to be done, adjust Q and R and rerun sims.

References

1) Brian Douglas, MATLAB, What is Linear Quadratic Regulator (LQR) Optimal Control, https://youtu.be/E_RDCFOlJx4?si=8mG4fGft02hM0yzS
2) Christopher Lum, Introduction to Linear Quadratic Regulator (LQR) Control, https://youtu.be/wEevt2a4SKI?si=UB8ifx64q_Lhgoqh