MATH4406 (Control Theory)
Unit 6: The Linear Quadratic Regulator (LQR)
and
Model Predictive Control (MPC)

Prepared by Yoni Nazarathy, Artem Pulemotov,

September 12, 2012

Unit Outline

Goal 1: Outline linear quadratic optimal control (LQR).

Goal 2: Introduce Concepts of Model Predictive Control (MPC).

Goal 3: A stability proof for linear quadratic MPC.

Auxiliary items:

- Background on positive (semi) definite matrices.
- Riccati equations.
- Quadratic Programming (convex optimization).

The Linear Quadratic Regulator (LQR)

The "L" part of LQR

The same old **linear** dynamics:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad \text{or} \quad x(k+1) = Ax(k) + Bu(k),$$

$$A \in \mathbb{R}^{n \times n}, \; B \in \mathbb{R}^{n \times m}, \; x(0) = x_0.$$

Assume,

$$\text{rank}\Big(con(A, B) := \big[B, AB, A^2B, \ldots, A^{n-1}B\big]\Big) \quad = \quad n,$$

i.e., the system is controllable (reachable)

So in the continuous time case, we can drive the state from x_0 to any state in any finite time, T. For the discrete time case it can be done in at most n steps.

The "Q" part of LQR

A cost structure:

$$J(u) = \int_0^T \Big(x(t)' \, Q \, x(t) + u(t)' \, R \, u(t) \Big) dt + x(T)' \, Q_f \, x(T),$$

or,

$$J(u) = \sum_{k=0}^{N-1} \Big(x(k)' \, Q \, x(k) + u(k)' \, R \, u(k) \Big) + x(N)' \, Q_f \, x(N).$$

The time horizon, T or N, can be finite or infinite.

The cost matrices satisfy,

$$Q = Q' \geq 0, \quad Q_f = Q_f' > 0, \quad R = R' > 0.$$

The "R" part of LQR

Since cost structure is,

$$J(u) = \int_0^T \Big(x(t)' \, Q \, x(t) + u(t)' \, R \, u(t) \Big) dt + x(T)' \, Q_f \, x(T),$$

or similar for discrete time, we see that we are trying to find a control $u(t), t \in [0, T]$ that will "regulate" the system "at 0". The payment is "quadratic" for both "state" and "control effort".

Typical choices for Q (or Q_f) are,

$$Q = \mathbf{1}\mathbf{1}' \quad \text{or} \quad Q = I \quad \text{or} \quad Q = \text{diag}(q_i),$$

where $q_i \geq 0$.

A typical choice for R is $R = \text{diag}(r_i)$, with $r_i > 0$.

The time horizon, T or N is often taken as ∞.

The LQR Success Story

It turns out that the optimal control is a linear state feedback control law. In the continuous time case,

$$u(t) = \big(- R^{-1} B' P(t) \big) x(t),$$

where the $n \times n$ matrix $P(t)$ is the solution of a Riccati differential equation (to be presented soon).

In the discrete time case,

$$u(k) = \Big(- \big(R + B' P(k+1) B \big)^{-1} B' P(k+1) A \Big) x(k),$$

where the $n \times n$ matrix $P(k)$ is the solution of a Riccati difference equation (to be presented soon).

Further if $T = \infty$ (or $N = \infty$) the terms $P(t)$ (or $P(k)$) are replaced by a constant matrix that is a solution of associated Riccati algebraic equations (different versions for discrete and continuous time).

Regulation of Trajectory Tracking

LQR is often used for tracking some desired trajectory, $\tilde{x}(\cdot)$. In the simplest case this trajectory is a non-zero desired constant set point, $\tilde{x}$.

Finding desired trajectories can be presented as a separate problem: E.g. open-loop optimal control based on calculus of variations.

In this case the objective may be formulated as:

$$J_{\tilde{x}}(u) = \int_0^\infty \Big(\big(x(t) - \tilde{x}(t)\big)' \, Q \, \big(x(t) - \tilde{x}(t)\big) + u(t)' \, R \, u(t) \Big) dt,$$

with similar versions for discrete time and/or finite time horizons.

LQR Results

The Riccati Equation - Continuous Time

This is the Riccati matrix differential equation used to find the state feedback control law of continuous time LQR. Solve it for $\{P(t),\ t \in [0, T]\}$

$$-\dot{P}(t) = A'P(t) + P(t)A - P(t)BR^{-1}B'P(t) + Q, \quad P(T) = Q_f.$$

Observe that it is specified "backward in time".

If $T = \infty$ the steady state solution P of the Riccati differential equation replaces $P(t)$ in the optimal control law. This P is the unique positive definite solution of the algebraic Riccati equation,

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

The optimal control is:

$$u(t) = \big(-R^{-1}B'P(t)\big)x(t) \quad \text{or} \quad u(t) = \big(-R^{-1}B'P\big)x(t).$$

The Riccati Equation - Discrete Time

This is the Riccati matrix- difference equation. Solve it for $\{P(k),\ k \in \{0, \ldots, N\}\}$

$$
\begin{aligned}
P(k) &= Q + A'P(k+1)A \\
&\qquad - A'P(k+1)B\big(R + B'P(k+1)B\big)^{-1}B'P(k+1)A, \\
P(N) &= Q_f.
\end{aligned}
$$

If $N = \infty$ the steady state solution P replaces $P(k)$. This P is the unique positive define solution found by the algebraic Riccati equation,

$$P = Q + A'PA - A'PB(R + B'PB)^{-1}B'PA.$$

The optimal control is:

$$u(k) = \Big(-\big(R+B'P(k+1)B\big)^{-1}B'P(k+1)A\Big)x(k), \text{or} \quad u(k) = \Big(-\big(R+B'PB\big)^{-1}B'PA\Big)x(k).$$

Stability

When $T = \infty$ (or $N = \infty$) we see that the resulting system is simply a state-feedback controller,

$$u(t) = Fx(t) \quad \text{or} \quad u(k) = Fx(k).$$

with $F = \left(- R^{-1} B' P \right)$ or $F = \left(- \left(R + B' P B \right)^{-1} B' P A \right)$.

Is it stable? In the continuous time case, the algebraic Riccati equation is,

$$0 = A' P + P A - P B R^{-1} B' P + Q.$$

Recall the formula $-(A'P + PA) > 0$ from Yoni's lecture on Lyapunov functions. The presence of this term suggests that we can hope to have stability. Indeed, we do under assumptions.

For the discrete time system, the analog of $A'P + PA$ is $P - A'PA$.

A Solved Example

Consider the continuous time (A, B, C, D) system as studied previously with,

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = 0.$$

We wish to find $u(\cdot)$ that minimizes,

$$J = \int_0^\infty \left(y^2(t) + \rho u^2(t)\right) dt.$$

Then this can be formulated as an LQR problem with,

$$Q = C'C, \quad R = \rho.$$

The algebraic Riccati equation turns out to be a system of equations for the elements of $P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}$,

$$-\frac{1}{\rho}p_2^2 + 1 = 0, \quad p_1 - \frac{1}{\rho}p_2 p_3 = 0, \quad 2p_2 - \frac{1}{\rho}p_3^2 = 0.$$

A Solved Example - cont.

$$P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}$$

$$-\frac{1}{\rho}p_2^2 + 1 = 0, \quad p_1 - \frac{1}{\rho}p_2 p_3 = 0, \quad 2p_2 - \frac{1}{\rho}p_3^2 = 0.$$

We will have $P > 0$ if and only if $p_1 > 0$ and $p_1 p_3 - p_2^2 > 0$.... We get,

$$P = \begin{bmatrix} \sqrt{2\sqrt{\rho}} & \sqrt{\rho} \\ \sqrt{\rho} & \sqrt{2\rho\sqrt{\rho}}. \end{bmatrix}$$

And the optimal feedback control law is,

$$u(t) = -\frac{1}{\rho}[\sqrt{\rho} \quad \sqrt{2\rho\sqrt{\rho}}]x(t).$$

LQR in MATLAB

Very simple:

$$[K, S, e] = lqr(SYS, Q, R, N)$$

N is an additional type of cost term,

$$2x(t)'Nu(t).$$

The return values:

$-K$ is the state feedback gain matrix.

S is the solution of the algebraic Riccati equation

e are the resulting closed loop eigenvalues (i.e. the eigenvalues of $A - BK$).

In practice this is often the preferred way of deriving an initial controller before making finer refinements (based on simulations and tests).

Model Predictive Control

MPC Overview

Model Predictive Control (MPC), also called "receding horizon control", works as follows:
For a plant modeled as, $x(k+1) = f\big(x(k), u(k)\big)$ an input,

$$u_{\cdot|k} = \Big(u(k|k), u(k+1|k), \ldots, u(k+N-1|k)\Big)$$

is determined at each time slot k based on $x(k)$. The input is selected as to minimize predicted costs over the "planning horizon" $k, k+1, \ldots, k+N$. Here N is the length of the planning horizon. Once $u_{|k}$ is determined, the control $u(k|k)$ is applied and at time $k+1$ the process is repeated.

For the calculation made during time k, denote the "predicted state" (due to $u_{\cdot|k}$) by $x(k+1|k), x(k+2|k), \ldots, x(k+N|k)$. Observe that in general if $N < \infty$, $u(k+1|k) \neq u(k+1|k+1)$ even though (without disturbances/noise),

$$x(k+1) = f\Big(x(k), u(k|k)\Big) = x(k+1|k).$$

MPC Notes

Model Predictive Control (MPC) is a sub-optimal control method that "makes sense". If you think about it, this is in a sense how we (individuals) sometimes make decisions.

It originated from the chemical process control industry in the 80's. There each time step is in the order of a few hours. With the advent of cheap fast computers - it is now often the method of choice for real-time controllers also (e.g. time step every 10 milliseconds). The challenge is to solve the optimization problem for $u_{\cdot|k}$ quickly.

It is not always the case that increasing the time horizon "N" is better.

The stability of systems controlled by MPC is in generality not trivial. We will show a stability proof for certain cases.

Linear Quadratic MPC

Model:

$$x(k+1) = Ax(k) + Bu(k).$$

Cost:

$$J(u) = \sum_{k=0}^{N-1} x(k)' \, Q \, x(k) + u'(k) \, R \, u(k) + x'(N) \, Q_f \, x(N).$$

so far... exactly LQR. But... add constraints:

$$F \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \leq b.$$

In practice there are often hard constraints on the state and the control. Hence linear quadratic MPC is in practice very useful.

A Simple Illustrative MPC Example

Consider, $x(k+1) = x(k) + u(k)$ with state and control constraints,

$$|x(k)| \leq \frac{3}{2}, \quad |u(k)| \leq 1.$$

Let $Q = R = 1$ and consider a planning horizon of $N = 2$.

At state $x(0)$ the MPC chooses $(u(0), u(1))$ as to minimize,

$$x(0)^2 + u(0)^2 + (x(0) + u(0))^2 + u(1)^2.$$

subject to the constraints,

$$|u(0)|, |u(1)| \leq 1, |x(0) + u(0)| \leq \frac{3}{2}.$$

Find the controller $u(0) = g\Big(x(0)\Big)$. Is it stable?

Formulation as a Quadratic Program (QP)

At time k (taken to be 0 for simplicity), given a measured (or estimated) state $x(k)$ we need to solve,

$$min_{u(0),u(1),\ldots,u(N-1)} \sum_{k=0}^{N-1} x(k)'\, Q\, x(k) + u'(k)\, R\, u(k) + x'(N)\, Q_f\, x(N)$$

s.t. $x(k+1) = Ax(k) + Bu(k)$ and,

$$F \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \leq b.$$

How can we pose this as an optimization problem (over finite vectors) just in the mN variables $u(0), \ldots, u(N-1)$?

Formulation as a Quadratic Program (QP)

Use

$$\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} x(0) + \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & & \vdots \\ \vdots & & \ddots & \\ A^{N-1}B & & \cdots & B \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}$$

This converts the optimization problem of the MPC controller to one that simply depends on the mN dimensional vector $u(0), \ldots, u(N-1)$.

The general form of a quadratic program (QP) is:

$$\begin{aligned} & min_z z' \tilde{Q} z + \tilde{P} z, \\ s.t. \quad & \tilde{F} z \leq \tilde{b}. \end{aligned}$$

With a bit of (tedious rearranging) the MPC controller can then be presented as a convex QP in mN decision variables. QPs where $\tilde{Q} > 0$ have a unique solution and are quite efficiently solvable!!!

The Closed Loop System is Non-Linear

MPC generates a "feedback" control law $u(k) = g\big(x(k)\big)$, where the function $g(\cdot)$ is implicitly defined by the unique solution of the QP. The resulting controlled system,

$$x(k+1) = Ax(k) + Bg\big(x(k)\big),$$

is in general non-linear (it is linear if there are no-constraints because then the problem is simply LQR).

Nevertheless, as will be seen in the HW (looking at parts of Bemporad, Morari, Dua and Pistikopoulus, 2002, "The explicit linear quadratic regulator for constrained systems"), the resulting system is piece-wise linear.

Stability of MPC

Stability of MPC

A system controlled by MPC is generally not guaranteed to be stable.

It is thus important to see how to "modify" the optimization problem in the controller so that the resulting system is stable.

We now show one such method based on adding an "end-point constraint" that forces the optimized $u_{\cdot|k}$ to drive the predicted system to state 0.

Our proof is for linear-quadratic MPC, yet this type of result exists for general MPC applied to non-linear systems.

Generalizations of the "end-point constraint method" also exist.

The Constrained Controllability Assumption

Based on the constraints $F \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \leq b$, denote by $\mathbb{X} \subset \mathbb{R}^n$ the state-space and by $\mathbb{U}(x)$ the allowable controls for every $x \in \mathbb{X}$.

For our stability result, we assume:

1. $Q > 0$
2. $0 \in \mathbb{U}(0)$
3. **Constrained Controllability Assumption:** Assume $\exists N_0$ such that for every initial condition $x(0) \in \mathbb{X}$ satisfying the constraint set, $\exists u(0), u(1), \ldots, u(N_0 - 1)$ that when applied as input, results in $x(N_0) = 0$.

In practice, verifying the constrained controllability assumption is not much harder than verifying that the system is controllable. For controllable unconstrained linear systems, $N_0 \leq n$.

The Modified Linear-Quadratic MPC

Add now an additional **end-point constraint** to the constraints of the optimization problem in the MPC controller:

$$x(N_0) = 0.$$

The problem can again be solved by a quadratic program, yet at any time k (for any measured state $x(k)$) will result in $u_{\cdot|k}$ that has a predicted state of $x(k + N_0) = 0$.

Observe that this does not imply that the system will actually be at state 0 after N_0 steps. Why?

Simple Example Revisited

Consider again, $x(k+1) = x(k) + u(k)$ with constraints, $|x(k)| \leq \frac{3}{2}$ and $|u(k)| \leq 1$, and with $Q = R = 1$ and a planning horizon of $N = 2$. Add now an **end point constraint** at $N = 2$.

At state $x(0)$ the MPC chooses $(u(0), u(1))$ as to minimize,

$$x(0)^2 + u(0)^2 + (x(0) + u(0))^2 + u(1)^2.$$

subject to the constraints,$|u(0)|$, $|u(1)| \leq 1$, $|x(0) + u(0)| \leq 3/2$, as well as the new end point constraint:

$$|x(0) + u(0) + u(1)| = 0$$

The minimization now requires $u(1) = -(x(0) + u(0))$ and this implies that $u(0) = -\frac{2}{3}x(0)$. The controlled system is now:

$$x(k+1) = \frac{1}{3}x(k).$$

Observe that it is asymp' stable but does not reach 0 in finite time.

A Stability Proof

To show that linear-quadratic MPC systems satisfying the constrained controllability assumptions are stable, we use a Lyapounov function constructed from the cost objective in the optimization problem. Denote,

$$V_N\Big(x\Big) = \min_{u} \sum_{k=0}^{N-1} \ell\big(x(k), u(k)\big) = \min_{u} J_N\Big(x, u\Big),$$
$$s.t. \;\; x(0) = x$$

where $\ell(\cdot, \cdot)$ is the cost per stage (quadratic in our case) and $J_N(x, u)$ is the cost function starting in state x with control u over the time horizon.

A Stability Proof - cont.

Then a feasible trajectory (one satisfying the constraints) for horizon $N-1$ based on feasible controls $u(0), \ldots, u(N-2)$, can be "prolonged" with no cost by setting $u(N-1) = 0$. This is because with the new end point constraint, a feasible trajectory for horizon $N-1$ has $x(N-1) = 0$ and $\ell\big(x(N-1), u(N-1)\big) = \ell(0,0) = 0$.

Thus all feasible u's for the $N-1$ horizon problem are feasible for the problem with horizon N and further,

$$J_{N-1}\Big(x(0), u\Big) = J_N\Big(x(0), [u, 0]\Big).$$

Hence,

$$V_N(x) \leq V_{N-1}(x).$$

A Stability Proof - cont. 2

Let now u_0^* be the control applied with a time horizon of N. The optimal cost to go based on Bellman's Dynamic Programming Principle (focus of the next unit) is,

$$V_N(x) = \ell(x, u_0^*) + V_{N-1}(Ax + Bu_0^*).$$

But due to the endpoint constraint (as shown on previous slide),

$$V_N(Ax + Bu^*) \leq V_{N-1}(Ax + Bu^*),$$

hence,

$$V_N(x) \geq \ell(x, u_0^*) + V_N(Ax + Bu_0^*).$$

or,

$$V_N(Ax + Bu_0^*) - V_N(x) \leq -\ell(x, u_0^*) < 0.$$

Where the last inequality holds for all $x \neq 0$ due to the cost structure.

Further it can be shown that $V_N(x)$ is continuous in x and hence it is a (discrete time system) Lyapounov function as required.