Lecture 14: Summary

Mikael Johansson
KTH - Royal Institute of Technology

Course aims

After the course, you should be able to

- analyze properties of discrete-time linear systems in state-space form
- compute optimal open-loop controls for state transfer via LP and QP
- use dynamic programming to design controllers and observers
- understand the receding-horizon idea
- know how MPC extends LQR to deal with state/control constraints
- design and tune MPC controllers for engeinering systems
- have a basic understanding of stability properties of MPC controllers
- know how MPC can be implemented (explicit/embedded)

A long list, but we covered it all!

Checklist

The basics:

- matrix manipulations, eigenvalue calculations, matrix exponential
- quadratic forms

Discrete-time linear systems

- sampling of continuous-time linear systems
- stability of discrete-time linear systems
- forced and free response, the prediction equations
- reachability, state-transfer and PBH
- observability, initial-state estimation
- state feedback and observers based on pole placement

Checklist

Convex optimization

- convex sets, convex functions and convex optimization problems
- special cases: linear and quadratic programming
- optimality conditions
- least-squares and linearly constrained least-squares solutions

Dynamic programming

- the value function
- the Bellman equation
- recursive updates of value function, from final state and backward

Checklist

Linear-quadratic regulator

- the criterion; impact of weights on solution; cheap/expensive control
- batch solution as least-squares, a linear state feedback
- the dynamic programming solution, a time-varying linear state feedback
- the stationary Riccati equation: solution and interpretation
- reference scaling and integral action

Model-predictive control

- receding-horizon LQR with linear state/control constraints
- representing constraints as linear inequalities
- formulating the quadratic program
- softening constraints, pre-stabilized predictions
- the dual-mode perspective

Checklist

Lyapunov stability

- Lyapunov functions for asymptotic stability of nonlinear systems
- linear systems and quadratic Lyapunov functions
- the Lyapunov equation
- Algebraic Riccati Equation solution as Lyapunov function
- invariant sets, from Lyapunov theory or by direct definition

Stability of MPC

- infinite-horizon LQR cost as Lyapunov function
- predicted cost as Lyapunov function
- the role of the terminal penalty and terminal set

Checklist

Least-squares and Kalman filtering

- forward dynamic programming
- the least-squares and Kalman filters
- the stationary Riccati equation: solution and interpretation

Reference following and disturbance models

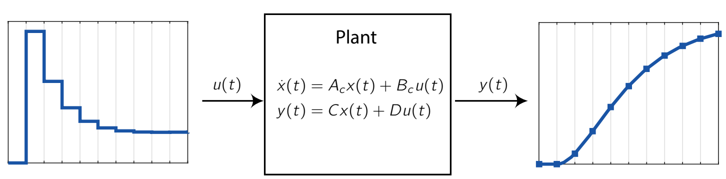
- penalizing deviations from reference states
- computing reference states
- offset-free MPC: limitations on numer of inputs and outputs
- disturbance models, disturbance observers

Lecture 2

Digital control using discrete-time process models.

$$x_{t+1} = Ax_t + Bu_t$$
$$y_t = Cx_t$$

Approximate (Euler) or exact (ZoH) sampling of continuous-time system.



Lecture 2

Autonomous system

$$x_{t+1} = Ax_t$$

asymptotically stable if $\max_i |\lambda_i(A)| < 1$ (all eigenvalues inside unit circle)

When driven by input sequence $\{u_t\}$, output is

$$x_t = A^t x_0 + \sum_{k=0}^{t-1} A^{t-k-1} B u_t := c_t + \mathcal{C}_t U$$

where

$$c_t = A^t x_0, \quad \mathcal{C}_t = \begin{bmatrix} B & AB & \dots & A^{t-1}B \end{bmatrix}, \quad U = \begin{bmatrix} u_{t-1} \\ \vdots \\ u_0 \end{bmatrix}$$

Lecture 2

Reachability: can drive x_t to any desired value

- if $\mathcal{C}_n$ has full rank (n= state dimension).
- can also be checked via the PBH test (useful for a lot of theory)

Observability: can estimate initial state from measurements of $\{y_t\}$

- if observability matrix

$$\mathcal{O}_n = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

  matrix has full rank,
- can also check PBH test for observability

Lecture 2

State feedback controllers

$$u_t = -Lx_t$$

designed by placing poles of $(A - BL)$, as in basic course.

Discrete-time observers

$$\hat{x}_{t+1} = A\hat{x}_t + Bu_t + K(y_t - \hat{y}_t)$$
$$\hat{y}_t = C\hat{x}_t$$

Gain matrix K designed by pole placement for $A - KC$ (error dynamics).

Examples

Exercise 1.1 (system response), 1.2, 1.3 (sampling)

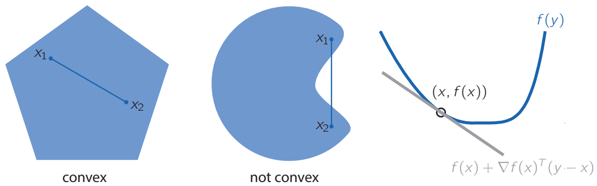
Exercises 1.5 (stability), 1.6, 1.7 (reachability)

Exercise 1.8 (observability)

Exercise 1.11, 1.12 (properties of closed-loop system)

Lecture 3

Convex sets and convex functions.



Convex optimization problems ($f_0, f_1, \ldots$ are convex functions):

$$\begin{array}{lll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0, & i = 1, \ldots, m \\ & g_i^T x = h_i, & i = 1, \ldots, p \end{array}$$

Local optimum is also global. Strong and useful theory. Reliable solvers.

Lecture 3

If $f(x)$ is convex, then any $x^\star$ that satisfies

$$\nabla f(x^\star) = 0$$

is global minimizer.

Important special case: least squares

$$f(x) = x^T P x + 2q^T x + r$$

with $P \succ 0$. Then $x^\star = -P^{-1} q$ (used to compute batch-LQ control).

Can also derive optimality conditions (KKT) for constrained optimization.

Some constrained problems also admit explicit solutions:
– e.g. least-norm solution to system of linear equations

Lecture 3

Modeling finite-time control problems as LPs or QPs:
- the prediction equations give linear equality constraints
- constraints on states and controls expressed as linear inequalities
- objective can be linear or (convex) quadratic

Minimum-energy state transfer, input-constrained state transfer, ...

Example

Exercises 3.1, 3.2 (convex functions, convex sets)

Exercise 3.7a-c (modeling with LPs and QPs; see also MPC lectures)

Lecture 4

Dynamic programming: technique for sequential decision-making problems

$$\begin{array}{ll} \text{minimize} & \sum_{t=0}^{N-1} g_t(x_t, u_t) + g_N(x_N) \\ \text{subject to} & x_{t+1} = f_t(x_t, u_t), \\ & x_t \in X_t,\ u_t \in U_t \end{array}$$

Form value function $V_t(x)$ characterizing (optimal) cost-to-go from $x_t = x$.

Compute V_t recursively, for $t = N, N-1, \dots$ using the Bellman equation

$$V_t(z) = \min_{u: u \in U_t \wedge f(z,u) \in X_{t+1}} g_t(z,u) + V_{t+1}(f_t(z,u))$$

Challenge: represent and update V_t in an efficient way.

Lecture 4

The linear-quadratic regulator: linear dynamics, quadratic cost.

$$\begin{array}{ll} \text{minimize} & \sum_{t=0}^{N-1} x_t^T Q_1 x_t + u_t^T Q_2 u_t + x_N^T Q_N x_N \\ \text{subject to} & x_{t+1} = Ax_t + Bu_t \qquad t = 0, 1, \dots, N-1 \end{array}$$

Value function is quadratic, $V_t(x) = x_t^T P_t x_t$, computed via Riccati recursion

$$P_{t-1} = Q_1 + A^T P_t A - A^T P_t B (Q_2 + B^T P_t B)^{-1} B^T P_t A$$

Solution is a sequence of postitive definite matrices $\{P_t\}$.

Example

Exercise 3.9, 3.10, 3.15 (dynamic programming)

Exercise 3.8, 4.1, 4.13 (LQR, Riccati recursion)

Lecture 5

Infinite-horizon LQR problem: find u which minimizes cost

$$\sum_{t=0}^{\infty} x_t^T Q_1 x_t + u_t^T Q_2 u_t$$

Optimal control is a linear time-invariant feedback

$$u_t = -Lx_t \qquad L = (Q_2 + B^T PB)^{-1} B^T PA$$

characterized by solution P to Algebraic Riccati Equation (ARE):

$$P = Q_1 + A^T PA - A^T PB(Q_2 + B^T PB)^{-1} B^T PA.$$

ARE solution P describes infinite-horizon cost-to-go $x^T Px$.

Lecture 5

How to choose weights Q_1 and Q_2 in

$$\sum_{k=0}^{\infty} x_k^T Q_1 x_k + u_k^T Q_2 u_k$$

to get desired closed-loop properties?

Basic idea from Bryson's rule.

Additional insight from considering

$$\sum_k y_k^T y_k + \rho u_k^T u_k$$

Closed-loop bandwidth vs control energy. Cheap and expensive control.

Lecture 5

LQR: regulates to zero, does not consider disturbances. Need additions.

Standard to include reference feed-forward and integral action

$$u_t = -Lx_t - l_i i_t + l_r r_t$$

where

- l_r is chosen so static gain from r to y is one
- i_t is an integral state, maintained by the controller

$$i_{t+1} = i_t + (r_t - y_t)$$

- L and l_i designed via LQ on an extended system

Example

Exercise 4.4a (Riccati equations)

Exercise 4.5 (recognize impact of weights on performance)

Although no exercises in compendium, you should also be able to compute l_r to get offset-free tracking, and be able to form the extended system for LQR design with integral action.

Lecture 6

Linear system with state and measurement disturbances

$$\begin{aligned} x_{t+1} &= Ax_t + w_t \\ y_t &= Cx_t + v_t \end{aligned} \qquad (1)$$

Denote our best prior guess on x_t by $\bar{x}_0$.

Least-squares principle: find smallest uncertainty that explain observations:

$$J_N = (x_t - \bar{x}_0)^T R_0 (x_t - \bar{x}_0) + \sum_{t=0}^{N-1} w_t^T R_1 w_t + v_t^T R_2 v_t$$

Recursive solution via forward dynamic programming.

Lecture 6

The Kalman filter: another parameterization of the same estimator.

Measurement update:

$$\begin{aligned}
\bar{K}_t &= S_t^{-1} C^T (R_2^{-1} + C^T S_t^{-1} C) = P_t C^T (\Sigma_v + C^T P_t C) \\
\hat{x}_{t|t} &= \hat{x}_{t|t-1} + \bar{K}_t \Delta y_t \\
P_t^+ &= P_t - P_t C^T (\Sigma_v + C P_t C^T)^{-1} C P_t
\end{aligned}$$

Prediction step:

$$\begin{aligned}
\hat{x}_{t+1|t} &= A \hat{x}_{t|t} \\
P_{t+1} &= \Sigma_w + A P_t^+ A^T
\end{aligned}$$

Lecture 6

Can eliminate intermediate variable P_t^+ and obtain Riccati recursion

$$P_{t+1} = \Sigma_w + A P_t A^T - A P_t C^T (\Sigma_v + C P_t C^T)^{-1} C P_t A^T$$

Tends to stationary solution as $t \to \infty$

$$P = APA^T + \Sigma_w - APC^T (\Sigma_v + CPC^T)^{-1} CPA^T$$

an algebraic Riccati equation; gives time-invariant observer (constant $\bar{K}_t$)

Example

Exercise 4.1 (Riccati equation for scalar system)

In addition, you should be able to solve Kalman filter ARE, and form the corresponding estimator.

Lecture 7

Lyapunov functions: ensure that "system energy" decreases with time.

Allows to analyze stability of nonlinear systems $x_{t+1} = f(x_t)$.

Basic result: look for positive definite Lyapunov function $V(x)$ such that

$$V(f(x)) - V(x) \leq -l(x) \qquad \forall x$$

Then, if $l(x)$ is positive definite, $x_t \to 0$ as $t \to \infty$ (details in Lecture 8)

Proofs reason about level sets of $V(x)$, and indirectly about system states.

Lecture 7

An autonomous linear system

$$x_{t+1} = Ax_t$$

is asymptotically stable iff it admits a quadratic Lyapunov function

$$V(x) = x^T P x$$

Solve Lyapunov equation

$$A^T P A - P + Q = 0$$

for some positive definite matrix $Q \succ 0$, check that $P \succ 0$.

Lecture 7

Positively invariant sets:

- a subset of state-space,
- if initial state in set, then all future states will also be in the set
- level sets of Lyapunov functions are invariant

Invariant set satisfying constraints $x(t) \in X = \{x \mid Hx \leq h\}$ for all $t \geq 0$:

- finding a Lyapunov function level set fully contained in X, or
- use the definition

  $$\{x \mid Hx \leq h \wedge HAx \leq h \wedge HA^2x \leq h, \dots\}$$

  a polyhedron; finite number of hyperplanes under certain conditions.

Control-invariant sets: there exists a control such that x remains in set.

Example

Exercise 2.1, 2.2, 2.5 (Lyapunov stability),

Exercise 2.8 (Lyapunov equation)

Exercise 2.10, 2.11, 2.12, 2.18 (invariant sets)

Lecture 8

Model predictive control: receding-hozion LQ with constraints.

Plan trajectory over horizon of N future samples by solving

$$\begin{array}{lll} \text{minimize} & \sum_{k=0}^{N-1} x_k^T Q_1 x_k + u_k^T Q_2 u_k + x_N^T Q_f x_N & \\ \text{subject to} & x_{k+1} = Ax_k + Bu_k & k = 0, \dots, N-1 \\ & x_k \in X, & k = 0, \dots, N \\ & u_k \in U & k = 0, \dots, N-1 \\ & x_N \in X_f, x_0 = x(t) & \end{array}$$

apply first control $u_0^\star$ in optimal sequence, repeat at next sampling instant.

Planning problem is a QP when X, U, X_f are polyhedra.

Lecture 8

Modeling constraints as linear inequalities

- magnitude and rate limitations on output, monotonicity constraints etc.

QP can be formulated as either

- extensive form, where both x and u are decision variables, or
- condensed form, where x is eliminated using prediction equations

Automated by software such as YALMIP, cvx, etc.

Lecture 8

Motivation and understanding via infinite-horizon constrained LQR

$$\begin{aligned}
\sum_{k=0}^{\infty} x_k^T Q_1 x_k + u_k^T Q_2 u_k &= \sum_{k=0}^{N-1} x_k^T Q_1 x_k + u_k^T Q_2 u_k + \sum_{k=N}^{\infty} x_k^T Q_1 x_k + u_k^T Q_2 u_k \\
&= \sum_{k=0}^{N-1} x_k^T Q_1 x_k + u_k^T Q_2 u_k + v(x_N) \\
&\approx \sum_{k=0}^{N-1} x_k^T Q_1 x_k + u_k^T Q_2 u_k + x_N^T Q_f x_N.
\end{aligned}$$

Insight:

- $x_N^T Q_f x_N$ should approximate infinite-horizon cost-to-go.
- terminal set should ensure approximation is valid (e.g. linear operation)
- approximate approach, unless horizon is sufficiently large

Examples

5.1, 5.5 (influence of tuning parameters on time responses)

5.2 (MPC optimization problem as QP)

5.3 (Modeling constraints as linear inequalities)

Lecture 9

Stability of receding-horizon LQR

- uses infinite-horizon cost-to-go as Lyapunov function
- basic proof relies on ARE solution for defining terminal const

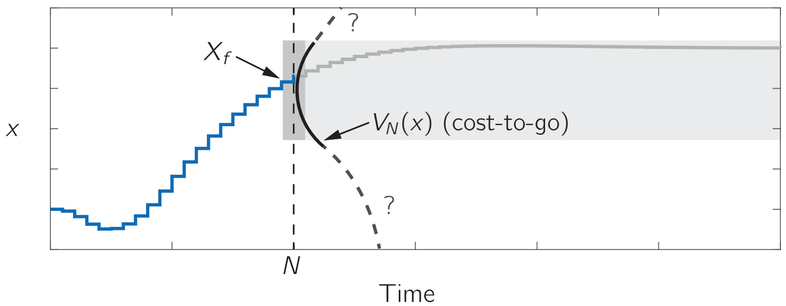
MPC stability proof: use predicted cost as Lyapunov function;

$$\begin{aligned}
J(x(t)) \quad = \quad & \min_{u_0, u_1, \ldots, u_{N-1}} && \textstyle\sum_{k=0}^{N-1} x_k^T Q_1 x_k + u_k^T Q_2 u_k + x_N^T Q_f x_N \\
& \text{subject to} && x_{k+1} = Ax_k + Bu_k \\
& && x_0 = x(t)
\end{aligned}$$

Proof argues using tail, control-invariance, and Lyapunov function decay.

Control invariance of terminal set also ensures recursive feasibility.

Lecture 9



Terminal state X_f:

- is control invariant
- allows us to compute quadratic cost-to-go, use as terminal penalty

Examples

Exercise 5.9 (Stability of receding-horizon LQR)

Exercise 5.6 (Terminal sets)

Exercises 5.4 (MPC stability)

Note that more advanced MPC problems have appeared in old exams.

Lecture 10

Softening of constraints to ensure feasibility

- at the expense of extra variables in the QP
- linear or quadratic slack penalty gives different properties

Pre-stabilized predictions for better numerical conditioning:

- re-parameterize control as $u_t = -Lx_t + \nu_t$ in prediction equations
- ν_t is free variable, $(A - BL)$ better conditioned than A.

Alternative to terminal set: explicit constraint checking

- optimizes control moves over horizon $0, 1, \ldots, N-1$.
- uses fixed strategy, e.g.

$$u_t = -Lx_t$$

  over horizon $N, N+1, \ldots, N + N_c - 1$.
- equivalent if N_c large enough, but flexible when constraints change

Lecture 10

Reference tracking: penalize deviations from target states, controls.

$$\begin{array}{ll} \text{minimize} & \sum_{k=0}^{N-1} q(x_k - x^{\mathrm{eq}}, u_k - u^{\mathrm{eq}}) + q_f(x_N - x^{\mathrm{eq}}) \\ \text{subject to} & x_{k+1} = Ax_k + Bu_k \\ & x_k \in X,\ u_k \in U,\ x_N \in X_f \end{array}$$

Target states and controls must satisfy

$$\begin{bmatrix} A - I & B \\ C & D \end{bmatrix} \begin{bmatrix} x^{\mathrm{eq}} \\ u^{\mathrm{eq}} \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}$$

Can be pre-computed if r is constant.

Lecture 10

If reference risks to be infeasible, soften reference tracking constraint

$$
\begin{array}{ll}
\text{minimize} & \sum_{k=0}^{N-1} q(x_k - \hat{x}_k, u_k - \hat{u}_k) + q_f(x_N - \hat{x}_N) + \varphi(s) \\
\text{subject to} & x_{k+1} = Ax_k + Bu_k \\
& x_k \in X \\
& u_k \in U \\
& x_N - \hat{x}_N \in X_f \\
& \begin{bmatrix} A - I & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{u}_k \end{bmatrix} = \begin{bmatrix} 0 \\ r + s \end{bmatrix}
\end{array}
$$

Lecture 11

Offset-free MPC: modeling, estimating and compensating for disturbances

- model disturbances as outputs of linear systems
- estimate the states of the disturbance models using an observer
- compensate for these in the MPC design

New limitations on target states, controls. (cf. reference following)

In this framework, constant disturbances yield integral action in controller.

Lecture 12

Getting the model

- from physics, data, or both

Tuning the controller

- impact of tuning performance, rules-of-thumb

Solving the MPC planning problem

- embedded vs explicit solvers

Example

Exercise 1 in offset-free MPC exercise.

In addition, we expect you to be able to be able to compute target states for a given system, and to be able to modify the MPC problem accordingly.

We also expect you to be able to form the augmented system for a given disturbance model, and to validate the conditions for offset-free tracking.

Material from lectures 10 and 12 have mostly appeared as "general knowledge questions" on past exams.

About the exam

When? Friday 18/10 2019, at 14.00-19.00

About the exam

You may bring the following items to the exam

- Print outs from course book, notes
- The book from the basic course
- Copies of lecture slides
- Mathematical handbook
- Calculator

You *may not* bring exercise materials or old exams.

About the exam

Expect an exam in the same spirit as last year's exam

Total of 5 problems, each valued at 10 points.

Rough idea of distribution of problem type, difficulty:

- 1-2 simple problems: basic insight and understanding of key concepts, effect of tuning parameters, etc; simple and straightforward calculations
- 1-2 intermediate problems: apply more advanced results (DP, Riccati and Lyapunov equations, invariant sets, . . .) in standard way
- 1-2 more difficult problems: apply techniques from class on new problems (often with hints to keep things "reasonable")

Ask if things are unclear on the exam; we will tell you if we cannot answer!

Beyond the exam

Learn more in our advanced courses!

- Modeling of dynamical systems
- Nonlinear control
- Reinforcement learning
- Hybrid and embedded systems

Beyond the exam

Put your skills to the test: make your master thesis with us!

- see the list on the web, or come talk to us!

Contribute to frontline research: enroll in our PhD program!

- an exciting career – come and talk to us!

Course evaluation and feedback

1. What is the strongest point(s) of the course?
2. What is the weakest point(s) of the course?
3. Which are the most important changes that we should do next year? (comment on lectures, labs, exercises)
4. Which part has been hardest to understand?

Be honest, but be constructive!

Course evaluation opens today. Help us to improve the course!