

Robot Learning and Control #3

Linear Quadratic Gaussian Control

Kazuki Shibata

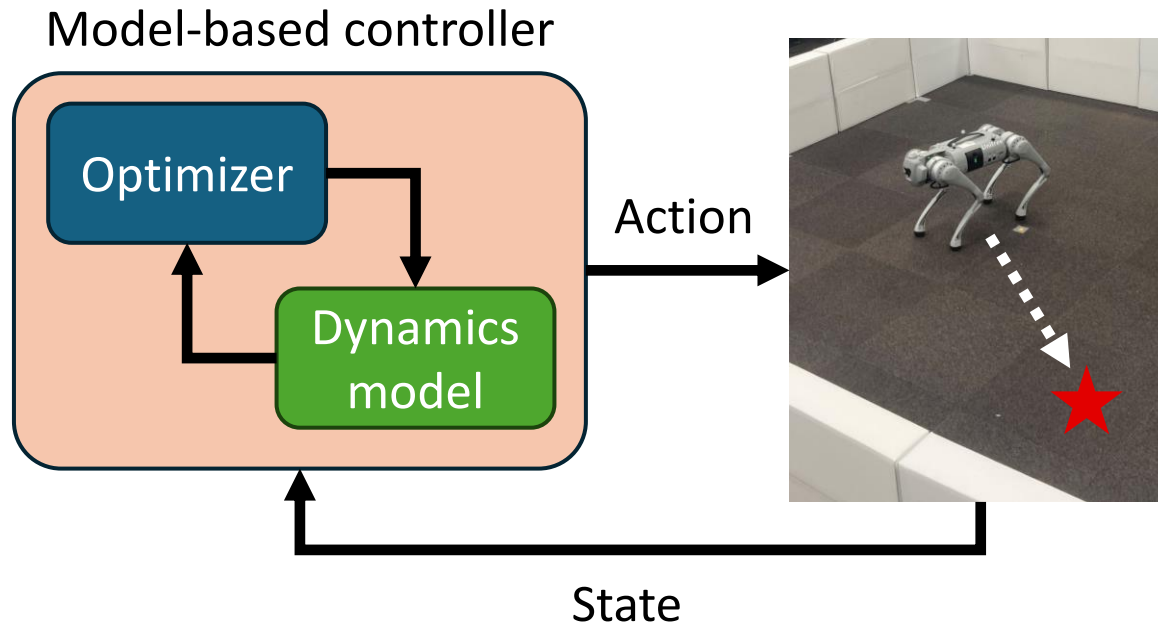
Robot Learning Lab

Nov. 22, 2024

General announcement

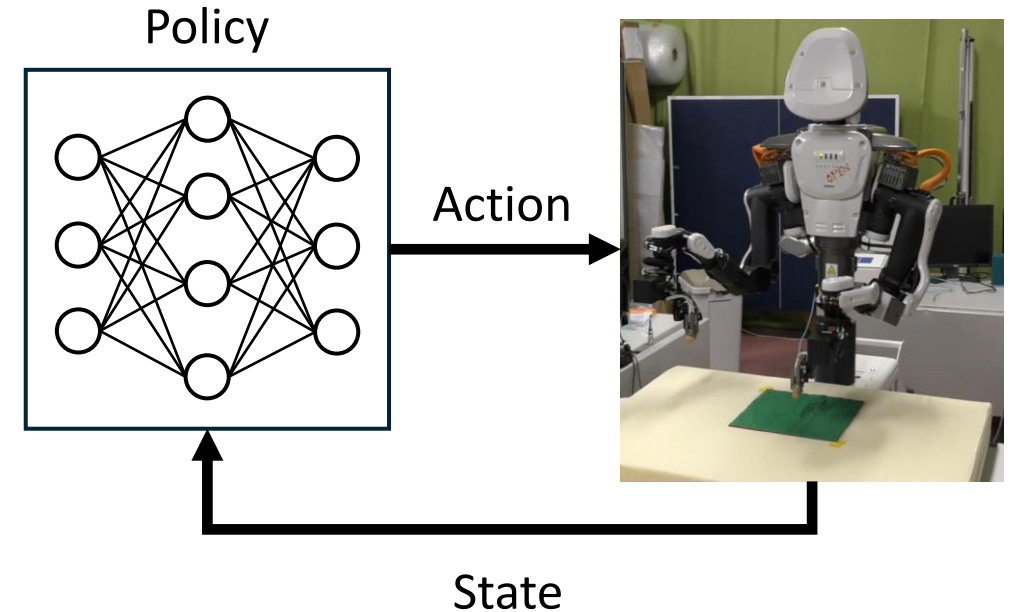
- The attendance is evaluated by **submitting the report paper.**
- The report is evaluated by **the exercises during the course.**
- Never give up!

Robot control



Model-based approach

- ✓ Provably guarantee the convergence
- ✗ Inapplicable without dynamics model

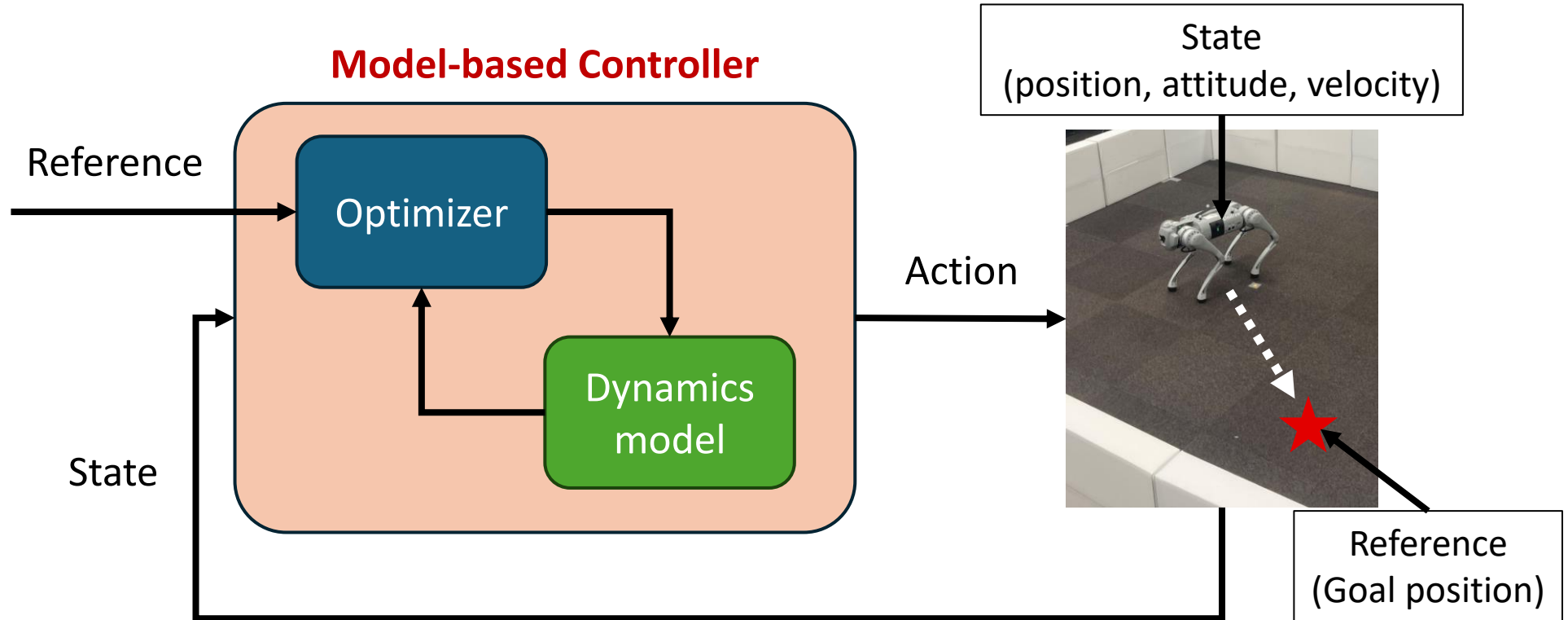


Model-free approach

- ✓ Applicable without dynamics model
- ✗ Unable to guarantee the convergence

This lecture focuses on the model-based approach based on the dynamics model.

Model-based Robot Control



How can we design robot controller by using dynamics model ?

Today's notebook

This lecture utilizes Google Colab to run example codes developed by AP Sasaki.



The screenshot shows a Google Colab interface. At the top, the notebook is titled "Lecture3.ipynb" with a star icon. Below the title, there are tabs for "ファイル", "編集", "表示", "挿入", "ランタイム", "ツール", and "ヘルプ", followed by the text "変更は保存されません". The main area of the notebook is divided into two sections: a left sidebar with icons for file management and a main code editor. The code editor contains a single code cell with the following Python code:

```
[ ] warnings.simplefilter('ignore')
env = gnwrapper.Monitor(gym.make('PosiEnv-v0'), directory="./random_action_video", video_callable=lambda a: True)
# env = gym.make('PosiEnv-v0')
env.name_prefix = "random_action"

obs = env.reset()
done = False

while not done:
    env.render()
    a = env.action_space.sample()
    obs, r, done, _ = env.step(a)

env.display()
```

URL: <https://colab.research.google.com/drive/1jIUhqFEJ8mDMW-IF2GkCkytKvuDjvf7Y?usp=sharing>

Table of Contents

- 1 Preliminary
- 2 Optimal Control Problem
- 3 Dynamic Programming
- 4 Linear Quadratic Gaussian Control (LQG)

Linear Gaussian Dynamics with Control

Consider a state-space model that involves zero-mean Gaussian noise:

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{C}\mathbf{w}_t \quad (1)$$

where, \mathbf{x} is state, \mathbf{u} is control input, and \mathbf{w} is noise that follows $w \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$

It can also be written as follows:

$$p(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1}; \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t, \sigma^2 \mathbf{C}\mathbf{C}^\top) \quad (2)$$

Let's confirm how to derive (2).

$$E[\mathbf{x}_{t+1}] = \int \mathbf{x}_{t+1} p(\mathbf{w}_t) d\mathbf{w}_t \quad (3)$$

$$= \int (\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{C}\mathbf{w}_t) p(\mathbf{w}_t) d\mathbf{w}_t \quad (4)$$

Since $\int p(\mathbf{w}_t) d\mathbf{w}_t = 1$ and $\int \mathbf{w}_t p(\mathbf{w}_t) d\mathbf{w}_t = E[\mathbf{w}_t] = 0$,

$$E[\mathbf{x}_{t+1}] = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t \quad (5)$$

Exercise 3-1: Linear Gaussian Dynamics with Control

Derive

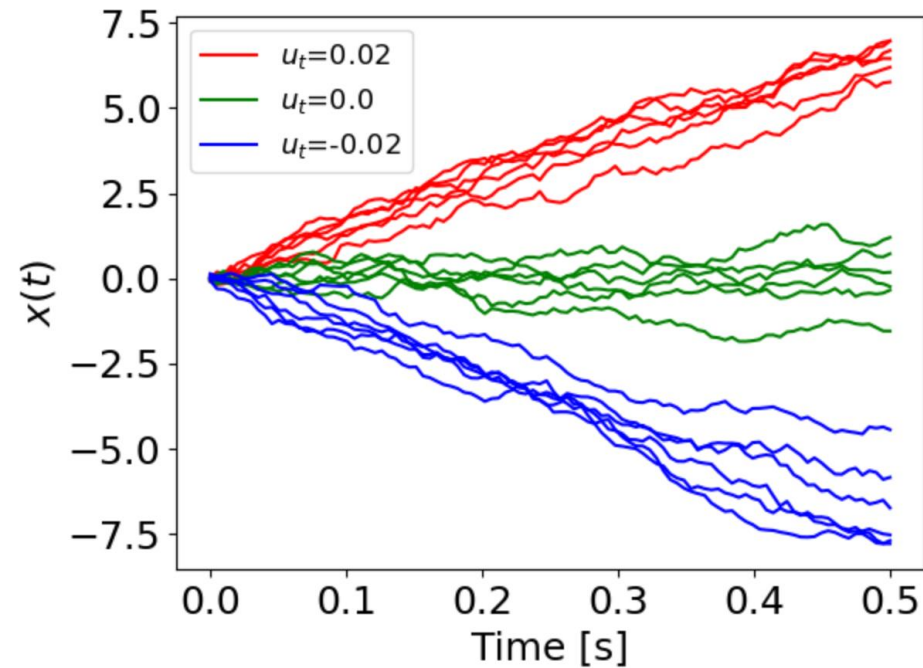
$$V(\mathbf{x}_{t+1}) = \sigma^2 \mathbf{C} \mathbf{C}^\top$$

using $V(\mathbf{x}_{t+1}) = \int (\mathbf{x}_{t+1} - E[\mathbf{x}_{t+1}])(\mathbf{x}_{t+1} - E[\mathbf{x}_{t+1}])^\top p(\mathbf{w}_t) d\mathbf{w}_t$

- You now have 5 minutes to write up your report
- Write your name and student number in your report.

Simulation: Linear Gaussian Dynamics with Control

Let's see Google Colab to confirm the behavior of linear Gaussian dynamics.



How can we design actions to control the state to the desired position?

Table of Contents

- 1 Preliminary
- 2 Optimal Control Problem
- 3 Dynamic Programming
- 4 Linear Quadratic Gaussian Control (LQG)

Optimal Control Problem

Optimal control problem involves finding a control law for a dynamical system over a given period such that an objective function is optimized.

Derive a control sequence $\mathbf{u}_{1:T} := (u_1, \dots, u_T)$ that minimizes

$$J_0(\mathbf{u}_{1:T}) = E \left[\sum_{t=1}^T c(\mathbf{x}_t, \mathbf{u}_t) \right] \quad (11)$$

where $c(\mathbf{x}_t, \mathbf{u}_t)$ is immediate cost function and the state follows $p(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t)$.

How can we solve the optimal control problem?

Naive Solutions

- **Random search:** generate random candidates of $\mathbf{u}_{1:T}$ and evaluate J_0 for each trial, and find the best one.
- **Direct search:** take the partial derivative of $\frac{\partial J_0}{\partial \mathbf{u}_{1:T}}$ and find $\mathbf{u}_{1:T}$ that makes it zero.

Both methods are not scalable for a large length of horizon T ...

Table of Contents

- 1 Preliminary
- 2 Optimal Control Problem
- 3 Dynamic Programming**
- 4 Linear Quadratic Gaussian Control (LQG)

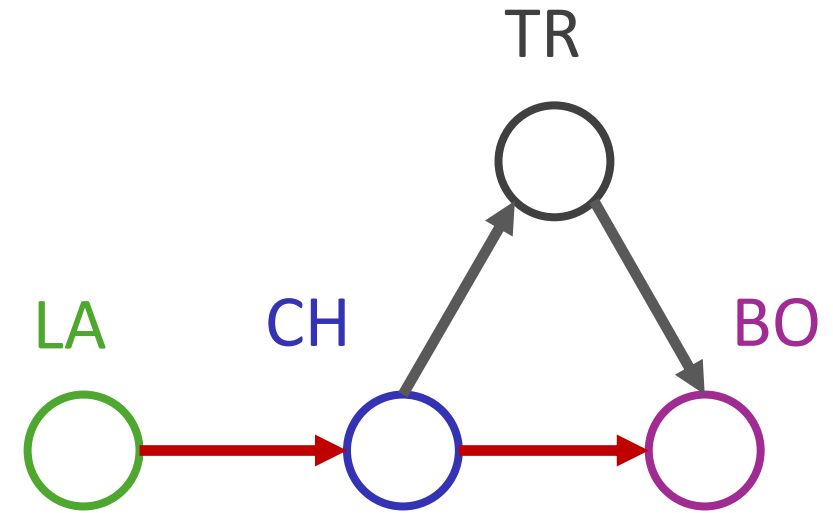
Principle of Optimality

Sub-solutions of a whole optimal solution of the problem are themselves optimal solutions for their Sub-problems.

- Intuitive Example:

Let's consider the optimal route from LA to BO.

- Whole optimal solution is $LA \rightarrow CH \rightarrow BO$
- If we know the shortest path from CH to BO, we could more easily find the shortest path from LA to BO.



We can solve the optimal problem more easily by considering it backward in time!

Dynamic Programming Algorithm

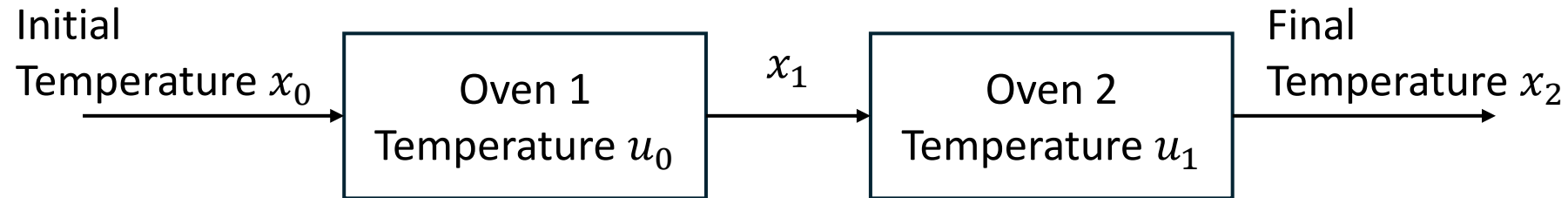
- Dynamic Programming:
 - if a problem can be solved optimally by breaking it into sub-problems, an efficient solution method is **recursively** finding the optimal solutions to the sub-problems
- Bellman equation:
 - consider a value function (total cost)

$$v_t(\mathbf{x}_t) = \min_{\mathbf{u}_{t:T}} E \left[\sum_{k=t}^T c(\mathbf{x}_k, \mathbf{u}_k) \right] \quad (8)$$

- recursion formula of value function backward in time is as follows:

$$v_t(\mathbf{x}_t) = \min_{\mathbf{u}_t} [c(\mathbf{x}_t, \mathbf{u}_t) + E[v_{t+1}(\mathbf{x}_{t+1})]] \quad (9)$$

Exercise 3-2: Two-ovens Problem (deterministic)



The temperature of the material evolves according to:

$$x_{t+1} = (1 - a)x_t + au_t \quad (10)$$

where a is a scalar in $0 < a < 1$, and $t = 0, 1$.

The objective is to get the final temperature x_2 close to a given target T , while reducing energy.

This is expressed by a cost function of the form

$$c_2(x_2, u_2) = c_2(x_2) = r(x_2 - T)^2, c_1(x_1, u_1) = c_1(u_1) = u_1^2, c_0 = u_0^2 \quad (11)$$

where r is a coefficient.

Please find optimal u_0 and u_1 by Bellman equation (10min)

Dimitri P. Bertsekas 2005

Table of Contents

- 1 Preliminary
- 2 Optimal Control Problem
- 3 Dynamic Programming
- 4 Linear Quadratic Gaussian Control (LQG)

Linear Quadratic Gaussian Control (LQG)

Linear Quadratic Gaussian Control is one of the most fundamental optimal control methods. It concerns

- linear Gaussian dynamics

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{C}\mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{I}) \quad (19)$$

- quadratic cost

$$c(\mathbf{x}_t, \mathbf{u}_t) = \mathbf{x}_t^\top \mathbf{Q}\mathbf{x}_t + \mathbf{u}_t^\top \mathbf{R}\mathbf{u}_t \quad (20)$$

Derive a control sequence $\mathbf{u}_{1:T}$ that minimizes

$$J_0(\mathbf{x}) = E \left[\sum_{t=1}^T c(\mathbf{x}_t, \mathbf{u}_t) \right] \quad (21)$$

Exercise 3-3: LQG

Assume the optimal value function is a second-order polynomial function of state:

$$v_{t+1}(\mathbf{x}_{t+1}) = \mathbf{x}_{t+1}^\top \mathbf{S}_{t+1} \mathbf{x}_{t+1} + s_{t+1} \quad (22)$$

where $\mathbf{S}_{t+1}^\top = \mathbf{S}_{t+1}$.

Derive the optimal control input \mathbf{u}_t^* using Bellman equation.

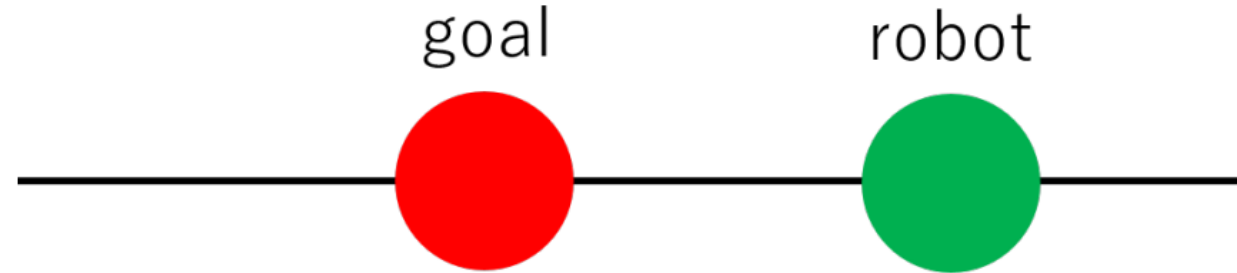
- You now have 10 minutes to write up your report

Use the following formula:

$$E[\mathbf{w}_t^\top \mathbf{M} \mathbf{w}_t] = \text{tr}(\mathbf{M} \Sigma_w)$$

Simulation: A point-mass control problem

- $\mathbf{x} = \begin{bmatrix} pos \\ vel \end{bmatrix}$
- $\mathbf{u} = [acc]$
- $\mathbf{A} = \begin{bmatrix} 1 & 0.3 \\ 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ 0.3 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.01 \end{bmatrix}$
- $\mathbf{Q} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.08 \end{bmatrix}, R = 0.02$



Let's see Google Colab to confirm the control result using LQG.

Linear to Nonlinear?

Robot dynamics model often follows nonlinear dynamics.

Consider the following extended model:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{C}\mathbf{w}_t \quad (32)$$

where $\mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) \neq \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t$

Can we solve the optimal control for such nonlinear systems?

→ it is mostly impossible in exact, but possible in approximation

Iterative Linear Quadratic Gaussian Control (iLQG)

The algorithm begins with an open-loop control sequences $\bar{\mathbf{u}}_{1:T}$, and the corresponding “zero-noise” trajectories $\bar{\mathbf{x}}_{1:T}$ obtained by $\bar{\mathbf{x}}_{t+1} = \mathbf{f}(\bar{\mathbf{x}}_t, \bar{\mathbf{u}}_t)$.

Some preparations with deviation state and action $\delta \mathbf{x}_t$ and $\delta \mathbf{u}_t$:

$$\mathbf{x}_t = \bar{\mathbf{x}}_t + \delta \mathbf{x}_t, \quad \mathbf{u}_t = \bar{\mathbf{u}}_t + \delta \mathbf{u}_t \quad (33)$$

$$\mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) = \mathbf{f}(\bar{\mathbf{x}}_t + \delta \mathbf{x}_t, \bar{\mathbf{u}}_t + \delta \mathbf{u}_t) \text{ (Taylor expansion)} \quad (34)$$

$$\approx \mathbf{f}(\bar{\mathbf{x}}_t, \bar{\mathbf{u}}_t) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}_t} \delta \mathbf{x}_t + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\bar{\mathbf{u}}_t} \delta \mathbf{u}_t \quad (35)$$

$$= \bar{\mathbf{x}}_{t+1} + \mathbf{A}_t \delta \mathbf{x}_t + \mathbf{B}_t \delta \mathbf{u}_t \quad (36)$$

Then, we can derive a linear system dynamics of deviation state and action at around $\bar{\mathbf{x}}, \bar{\mathbf{u}}$ as a pseudo linear as follows:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{C} \mathbf{w}_t \quad (37)$$

$$\bar{\mathbf{x}}_{t+1} + \delta \mathbf{x}_{t+1} \approx \bar{\mathbf{x}}_{t+1} + \mathbf{A}_t \delta \mathbf{x}_t + \mathbf{B}_t \delta \mathbf{u}_t + \mathbf{C} \mathbf{w}_t \quad (38)$$

$$\delta \mathbf{x}_{t+1} = \mathbf{A}_t \delta \mathbf{x}_t + \mathbf{B}_t \delta \mathbf{u}_t + \mathbf{C} \mathbf{w}_t \quad (39)$$

Todorov and Li, ACC2005

Exercise 3-4: iLQG

The cost function is also expanded as follows:

$$c(\mathbf{x}_t, \mathbf{u}_t) = \mathbf{x}_t^\top \mathbf{Q} \mathbf{x}_t + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t \quad (40)$$

$$= (\bar{\mathbf{x}}_t + \delta \mathbf{x}_t)^\top \mathbf{Q} (\bar{\mathbf{x}}_t + \delta \mathbf{x}_t) + (\bar{\mathbf{u}}_t + \delta \mathbf{u}_t)^\top \mathbf{R} (\bar{\mathbf{u}}_t + \delta \mathbf{u}_t) \quad (41)$$

$$\begin{aligned} &= \bar{\mathbf{x}}_t^\top \mathbf{Q} \bar{\mathbf{x}}_t + 2\bar{\mathbf{x}}_t^\top \mathbf{Q} \delta \mathbf{x}_t + \delta \mathbf{x}_t^\top \mathbf{Q} \delta \mathbf{x}_t + \bar{\mathbf{u}}_t^\top \mathbf{R} \bar{\mathbf{u}}_t \\ &\quad + 2\bar{\mathbf{u}}_t^\top \mathbf{R} \delta \mathbf{u}_t + \delta \mathbf{u}_t^\top \mathbf{R} \delta \mathbf{u}_t \end{aligned} \quad (42)$$

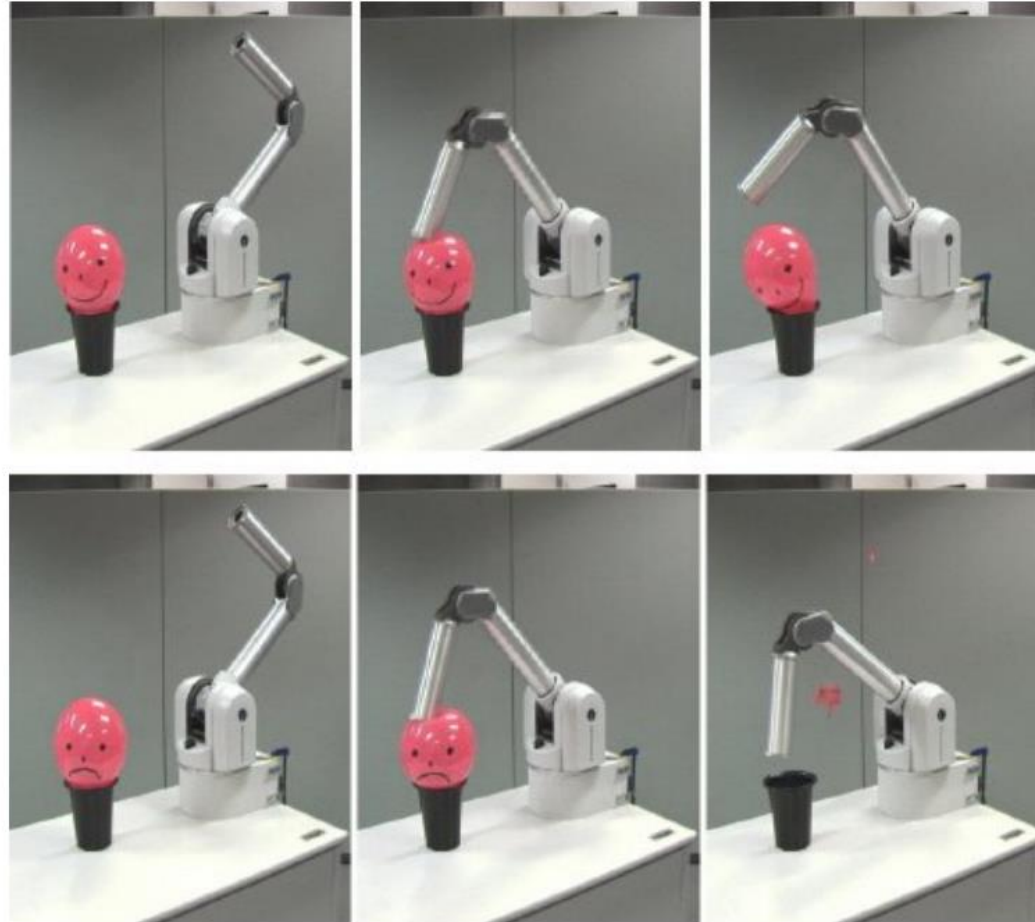
Assume the following value function:

$$v_t(\delta \mathbf{x}_t) = s_t + \delta \mathbf{x}_t^\top \mathbf{s}_t + \delta \mathbf{x}_t^\top \mathbf{S}_t \delta \mathbf{x}_t \quad (43)$$

Derive the optimal control input $\delta \mathbf{u}_t^*$ using Bellman equation.

- You now have 10 minutes to write up your report

Robot Application



Mitrovic+, ICRA2010

Let's see youtube video

References

- [1] Y. Tsurumine, Y. Cui, E. Uchibe and T. Matsubara, "Deep dynamic policy programming for robot control with raw images," IROS2017, pp. 1545-1550.
- [2] Dimitri P. Bertsekas: Dynamic programming and optimal control, 3rd Edition. Athena Scientific 2005, ISBN 1886529264, pp. I-XV, 1-543
- [3] Emanuel Todorov: Optimal Control Theory, December 2006, DOI: 10.7551/mitpress/9780262042383.003.0012
- [4] Emanuel Todorov, Weiwei Li: A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. ACC 2005: 300-306vol.1
- [5] Djordje Mitrovic, Sho Nagashima, Stefan Klanke, Takamitsu Matsubara, Sethu Vijayakumar: Optimal Feedback Control for anthropomorphic manipulators. ICRA 2010: 4143-4150