
Lecture notes on Control Systems and Reinforcement Learning

Written by
Manuel Hinz

mh@mssh.dev or s6mlhinz@uni-bonn.de

Lecturer

Prof. Dr. Jochen Garcke
[garcke\[at\]math.uni-bonn.de](mailto:garcke[at]math.uni-bonn.de)



University of Bonn
Summer semester 2025
Last update: April 10, 2025

Contents

Chapter 0	Manuel’s notes	2
0.1	Organization	2
Chapter 1	Control Problems	3
1.1	State Space Models	4
1.1.1	Linear State Space Model	5
1.1.2	State Space Models in continuous Time	5
Journal		8
Bibliography		8

Chapter 0:

Manuel's notes

Warning

These are unofficial lecture notes written by a student. They are messy, will almost surely contain errors, typos and misunderstandings and may not be kept up to date! I do however try my best and use these notes to prepare for my exams. Feel free to email me any corrections to mh@mssh.dev or s6mlhinz@uni-bonn.de.
Happy learning!

General Information

- Basis: [Basis](#)
- Website: <https://ins.uni-bonn.de/teachings/ss-2025-467-v5e1-advanced-topics/>
- Time slot(s): **Tuesday: 14-16** SR 2.035 and **Thursdays: 16-18** SR 2.035
- Exams: ?
- Deadlines: ?

0.1 Organization

- Focused on ingredients, won't get to the current state of the art
- Some algorithmic / numerical background (Euler method is fine)
- Control Problems (Steering the bike / car)

Start of lecture 01
(10.4.2025)

Chapter 1:

Control Problems

1. u is the control (input / action)
2. y observations (outputs)
3. $\phi : Y \rightarrow U$ policy
4. ff feed forward control (plan we had)

Interactions with the outside world might be hidden in the observations. Typically ff is in regard to some reference state. There might be some disturbances (holes in the road, ...).

The overall aim is to find a policy ϕ that sticks close to $r(k), k \geq 0$.

t is continuous, k is step
by step / iterative

$$u(k) = u_{ff}(k) + U_{fb}(k)$$

where u_{ff} is the planing to reach the overall goal and u_{fb} actual steering, updated "all the time".
Some examples from the book:

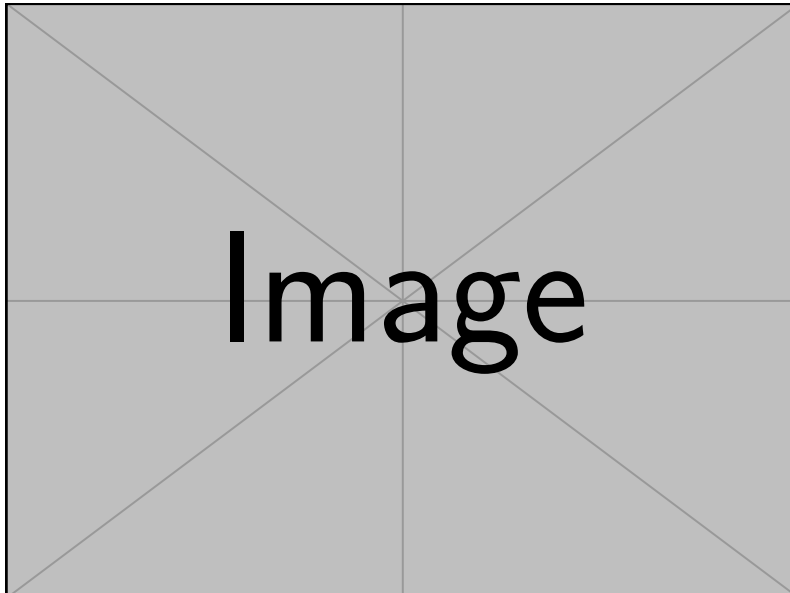


Figure 1.1: Sketch 1.01

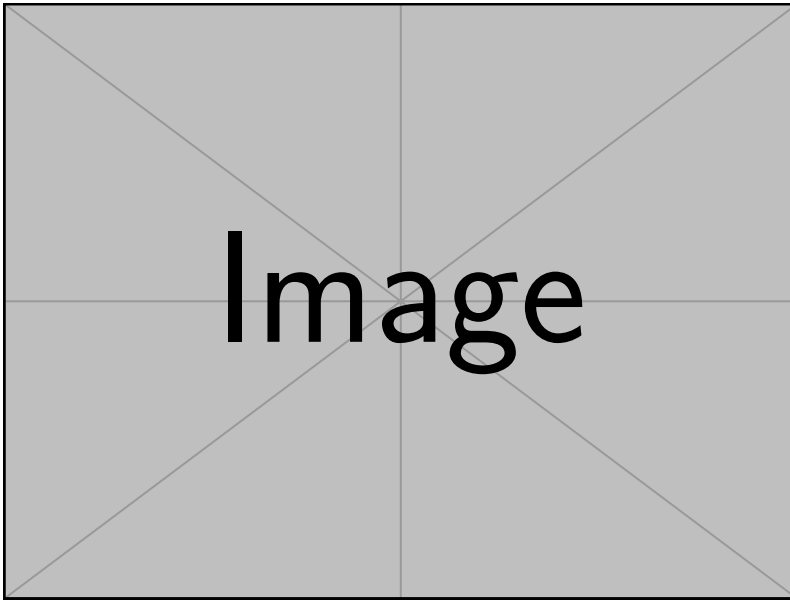


Figure 1.2: Sketch 1.02: Mountain car

Difference: In Reinforcement learning, we don't start with a model / ode.
Some part of reinforcement learning works model-free (i.e. assumes the model only implicitly)

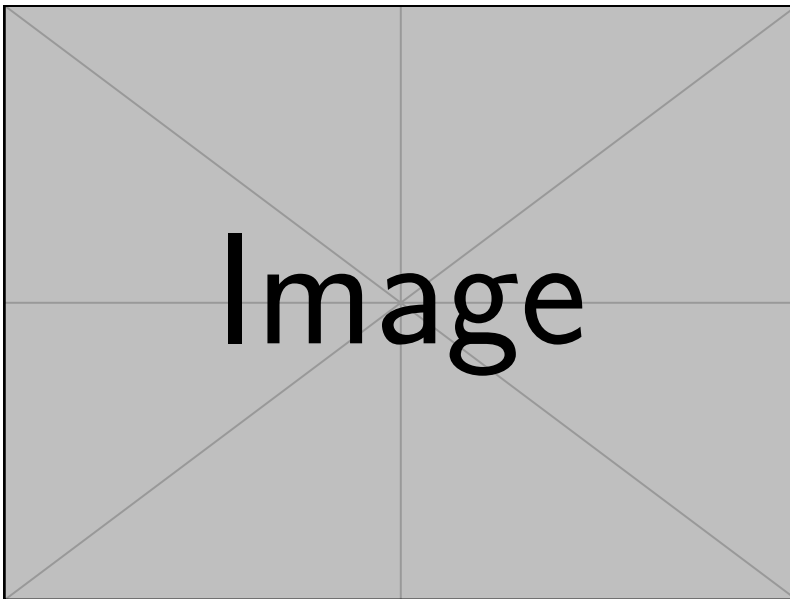


Figure 1.3: Sketch 1.03: cart pole / inverted pendulum

Next example: Acrobot (more than one equilibrium)

1.1 State Space Models

We have some

- state space $X, x \in X$
- action space $U, u \in U$
- action at step $k : u(k) \in U(k)$, i.e. we might have some constraints

- observation space $Y, y \in Y$

Definition 1.1. Given state, action and observation spaces X, U, Y , a state space model is defined by

$$x(k+1) = \mathcal{F}(x(k), u(k)) \quad (1)$$

$$y(k) = \mathcal{C}(x(k), u(k)) \quad (2)$$

$x(k)$ might include the past, might be useful for the stock trading problem

Remark. Overcomplicating problems by loading lots of information into the state space, might make the problem harder!

1.1.1 Linear State Space Model

$$x(k+1) = Fx(k) + Gu(k) \quad (3)$$

$$y(k) = Cx(k) + Du(k) \quad (4)$$

Remark. The representations (in terms of the matrices) might not be unique!

Common scenario for (3) is to keep $x(k)$ near the origin. You have to think about robustness of the system. Disturbances should be handled by the system.

$$u(k) = -Kx(k).$$

Consider a disturbance under the same control:

$$u(k) = -Kx(k) + v(k)$$

inserting this into (3) yields

$$x(k+1) = (F - GK)x(k) - Gv(k)$$

$$y(k) = (C - DK)x(k) + Dv(k)$$

Closed vs open loop: In closed loops we don't change our course based on observations, while in open loop systems we do.

1.1.2 State Space Models in continuous Time

$$\frac{d}{dt}x = f(x, u)$$

for $x \in \mathbb{R}^n, u \in \mathbb{R}^m$. We often write u_t, x_t for u, x at time t . If f is linear we get

$$\frac{d}{dt}x = Ax + Bu$$

$$y = Cx + Du$$



Figure 1.4: Sketch 1.04

To discretize we use the forward Euler method. Given time interval Δ

$$x(k+1) = x(k) + \Delta f(x(k), u(k))$$

so in (1) $\mathcal{F}(x, u) = x + \Delta f(x, u)$. Using Taylor

$$x_{t+\Delta} = x_t \Delta f(x, u) + O(\Delta^2)$$

For the linear model we get $F = I + \Delta A$

$$x(k+1) = x(k) + \Delta A x(k) + \underbrace{\Delta B}_{=:G} u(k)$$

For now fix some policy ϕ , so $u(k) = \phi(x(k))$:

$$x(k+1) = \mathcal{F}(x(k))$$

Assumption 2:

The state space X is equal to \mathbb{R}^n or a closed subset of \mathbb{R}^n .

Definition 1.2. An equilibrium x^e is a state at which is system is frozen:

$$x^e = \mathcal{F}(x^e).$$

Definition 1.3. Given a cost function $C : X \rightarrow \mathbb{R}_+$ and a policy ϕ we define

$$J_\phi(x) = J(x) = \sum_{k=0}^{\infty} C(x(k)), \quad x(0) = x$$

This is called total cost or value function of the policy ϕ .

Given x^e , we usually assume $C(x^e) = 0$. Generally, we consider a discount factor γ^k in front of $C(x(k))$.

Definition 1.4. Denote by $\mathcal{X}(k; x_0)$ the state step k with initial condition x_0 and following fixed policy ϕ . The equilibrium x^e is stable in the sense of Lyapunov if for all $\epsilon > 0 \exists \delta > 0$ s.t. $\|x_0 - x^e\| < \delta$, then

$$\|\mathcal{X}(k; x_0) - \mathcal{X}(k; x_0)\| < \epsilon \forall k \geq 0$$

The same concept with a different sign comes up in RL under the term reward

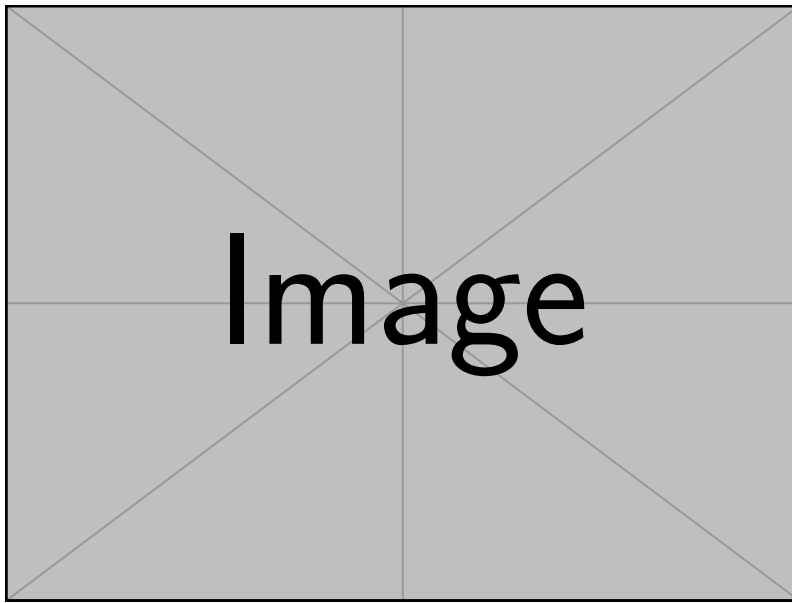


Figure 1.5: Sketch about Lyapunov stability

Definition 1.5. An equilibrium is said to be asymptotically stable if x^e is stable in the sense of Lyapunov and for some $\delta_0 > 0$, whenever $\|x_0 - x^e\| < \delta_0$, it follows

$$\lim_{k \rightarrow \infty} \mathcal{X}(k, x_0) = x^e.$$

The set of x_0 for which this holds is the region of attraction for x^e . An equilibrium is globally asymptotically stable if the region of attraction is X .

Journal

- Lecture 01: Covering:

.
Starting in ‘[Organization](#)’ on page [2](#) and ending in ‘[State Space Models in continuous Time](#)’ on page [7](#). Spanning 5 pages