Exam Q&As

for the Humanoid Robotics (RO5300) course
by Prof. Dr. Elmar Rueckert

latest updated July 9th 2018

IM FOCUS DAS LEBEN

Agenda

- Lecture:
  - 03.07 RL part 3
  - 10.07 Planning & Exam Q examples
  - 17.07 Exam
  - ~~24.07 free~~
  - 31.07 Debrief & Exam Q&A
- Exercise:
  - 04.07 Exam Q examples
  - 11.04 Assignment 4
  - ~~18.07 free~~
  - ~~25.07 free~~

You can evaluate this course till August, 31st 2018.

Organisation of the course

Organisation

Grading

- Lecture 60 points, at least 30 points have to be achieved!
- Graded assignment 60 points.
- 1-2 points for correct answers during the lecture, at most 20 points!

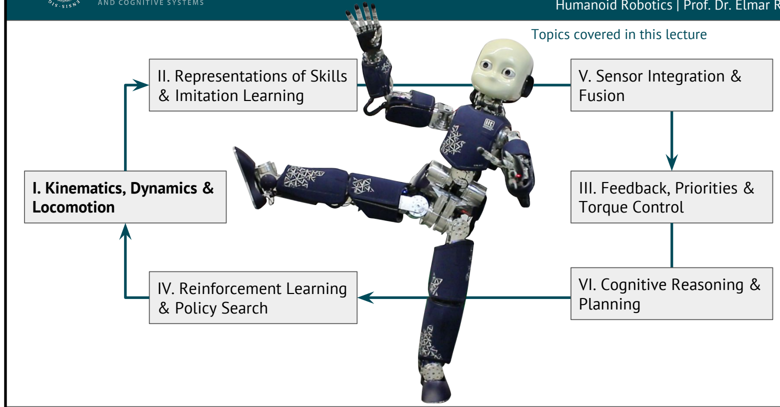
≥ 95 points	1.0	≥ 65 points	3.0
≥ 90 points	1.3	≥ 60 points	3.3
≥ 85 points	1.7	≥ 55 points	3.7
≥ 80 points	2.0	≥ 50 points	4.0
≥ 75 points	2.3	< 50 points	5.0 ☹
≥ 70 points	2.7		

If you fail, you will be automatically registered for the next exam date!

Exam 17.07.2018, 10:15 - 11:45

- 60 Points
- > 30 pts to receive a positive grade
- 90 minutes to answer questions from 5 chapters
  a. Kinematics and Dynamics
  b. Movement Representations
  c. Feedback Control
  d. Reinforcement Learning
  e. Planning
- We will correct your exams and send you an email till 29.07.2018
- 31.07.2018, 10:15 “Nachbesprechung”

Topics covered in this lecture



I. Kinematics, Dynamics & Locomotion

II. Representations of Skills & Imitation Learning

V. Sensor Integration & Fusion

III. Feedback, Priorities & Torque Control

VI. Cognitive Reasoning & Planning

IV. Reinforcement Learning & Policy Search

Learning Objectives of Chapter I (*Lernziele* in german).

In this course you will learn …

- To define and derive forward and inverse kinematic transformations for robotic chains.
- To use these transformations for closed-loop feedback control.

I.1 Classical forward and inverse kinematic.

Can you answer the following questions?

- What is the task space and what is the configuration space in a humanoid?
- What are the forward and the inverse kinematic transformations?
- What is a Jacobian matrix?
- What is the difference between the Newton’s and the gradient based IK control?

I.1 Classical forward and inverse kinematic.

Summary of I.1 Classical forward and inverse kinematics

- The forward kinematic transformation is a **unique function**, whereas the inverse kinematic transformation is a **mapping** with no, one or multiple solutions.
- For feedback control, incremental IK approaches based on **Jacobian inverse** (Newton) or **Jacobian transpose** (gradient descent) operations are used.
- Regularizations and resting poses are needed to handle **singular cases** during control (advanced approaches use the SVD or Levenberg-Marquardt).
- The **Jacobian** can be computed **analytically** (what we did here) or numerically through **finite differences** (exercise) or through **algorithmic differentiation**.

Learning Objectives of Chapter I (*Lernziele* in german).

In today's lecture you will learn ...

- about the difference between kinematics and dynamics in robotics.
- how dynamics equations for mechanical systems can be derived.
- how to use Newton-Euler and Runge Kutta methods for robot dynamics simulation.
- why probabilistic inverse kinematic controller are useful.

I.2-4 Dynamics, Integration and Bayes.

Can you answer the following questions?

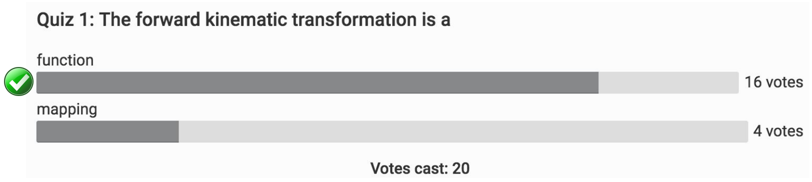
- What is the difference between kinematics and dynamics?
- How can dynamics equations be derived?
- Define the general/vector notation of the rigid body dynamics.
- What is the assumption underlying Euler's integration method?
- What sources of noise challenge robot control?

I.2-4 Dynamics, Integration and Bayes.

Summary of I.2-4 Dynamics, Integration and Bayes

- Robot dynamics models the **dynamic coupling** of joints and full Newtonian physics.
- Should be used for **fast** or **energy efficient** solutions or when kinematic control cannot be applied.
- The dynamics equations can be obtained through **Euler-Lagrange** (for theoretical studies) or **Newton-Euler recursions** (for real time control).
- Numerical integration is needed to **simulate** the behavior of a system given the **forward dynamics**.
- **Probabilistic** control schemes are used to model **sensor noise**, **inaccuracies** in the **mechanical** design, the **numerical** approximations and in the **inputs**.

Recap of the previous lecture

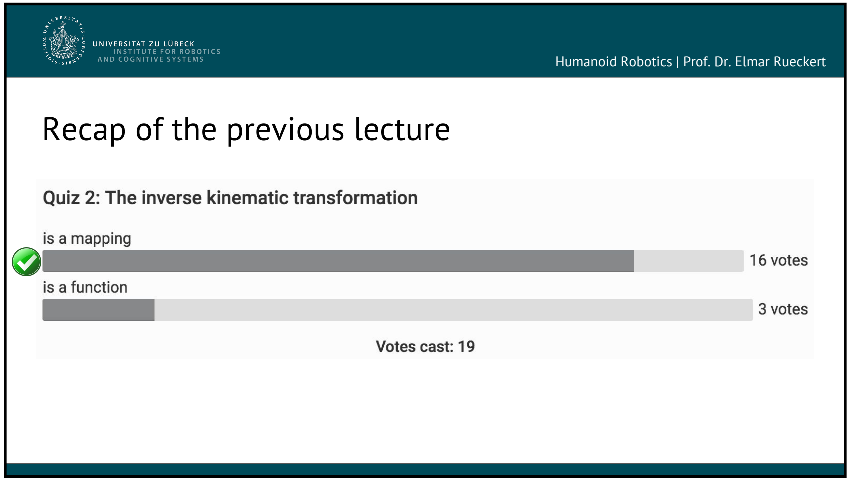


Quiz 1: The forward kinematic transformation is a

✓ function — 16 votes

mapping — 4 votes

Votes cast: 20



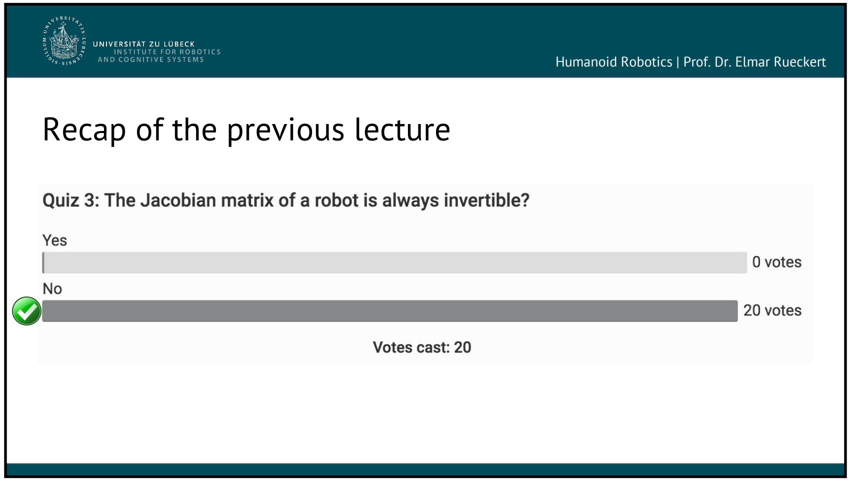
Recap of the previous lecture

Quiz 2: The inverse kinematic transformation

✓ is a mapping — 16 votes

is a function — 3 votes

Votes cast: 19



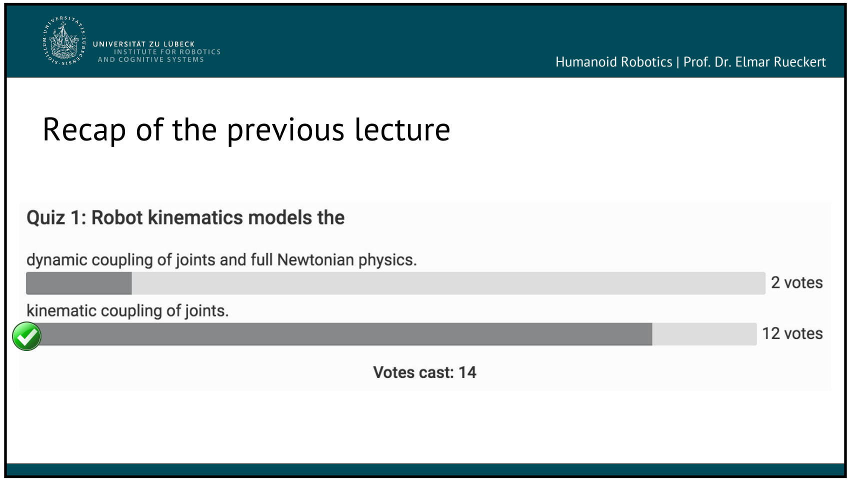
Recap of the previous lecture

Quiz 3: The Jacobian matrix of a robot is always invertible?

Yes — 0 votes

✓ No — 20 votes

Votes cast: 20



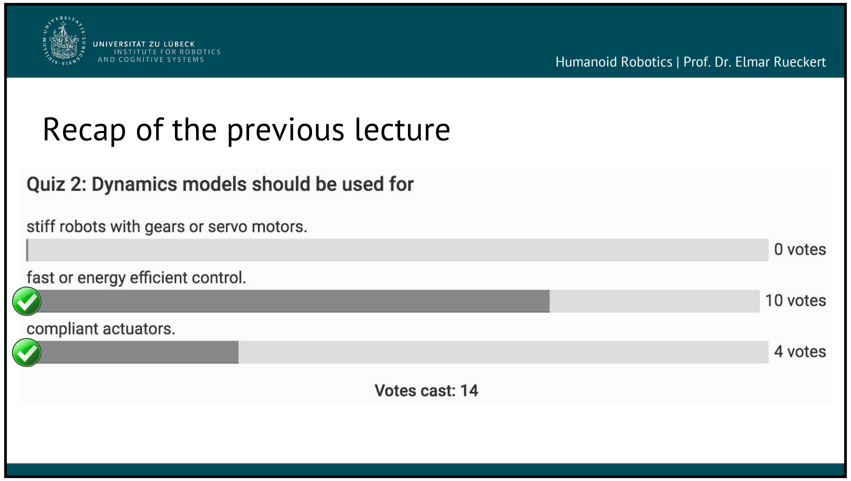
Recap of the previous lecture

Quiz 1: Robot kinematics models the

dynamic coupling of joints and full Newtonian physics. — 2 votes

✓ kinematic coupling of joints. — 12 votes

Votes cast: 14



Recap of the previous lecture

Quiz 2: Dynamics models should be used for

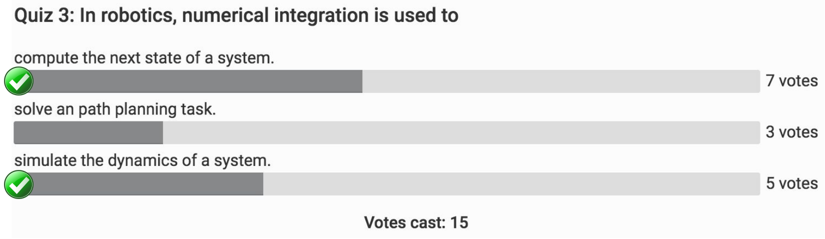
stiff robots with gears or servo motors. — 0 votes

✓ fast or energy efficient control. — 10 votes

✓ compliant actuators. — 4 votes

Votes cast: 14

Recap of the previous lecture



Quiz 3: In robotics, numerical integration is used to

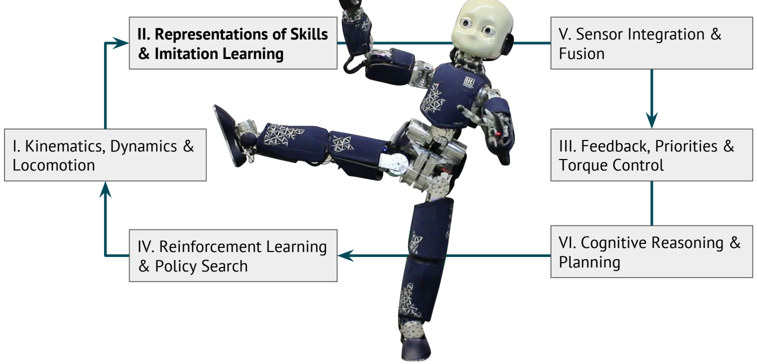
✓ compute the next state of a system. — 7 votes

solve an path planning task. — 3 votes

✓ simulate the dynamics of a system. — 5 votes

Votes cast: 15

Topics covered in this lecture


II. Representations of Skills & Imitation Learning
V. Sensor Integration & Fusion
I. Kinematics, Dynamics & Locomotion
III. Feedback, Priorities & Torque Control
IV. Reinforcement Learning & Policy Search
VI. Cognitive Reasoning & Planning

Learning Objectives of Chapter I (*Lernziele* in german).

In today's lecture you will learn …

- how movement primitives (MPs) can represent complex motions.
- how robot skills are learnt from demonstrations with dynamical systems MPs.

II.1-2 Movement primitives and DMPs.

Can you answer the following questions?

- Which movement representations were discussed?
- How do these representations scale with the data?
- What are the desired features of movement representations?
- What are dynamical systems movement primitives?
- Which types of imitation learning were discussed?

Summary of II.1 Movement primitives

- Discussed movement representations: **matrix, via-points, splines, dynamical systems movement primitives (DMPs)**.
- DMPs are **decoupled model** of a multi-dim. system.
- The goal **attractor state** ensures stable and straight traj. prior to any learning (when the weights are zero).
- DMPs can be learned from demonstrations via (regularized) **least squares regression**.
- Types of learning from demonstrations: **kinesthetic teaching** (or teach-in)**, teleoperation, visual observation, sensor suits**.

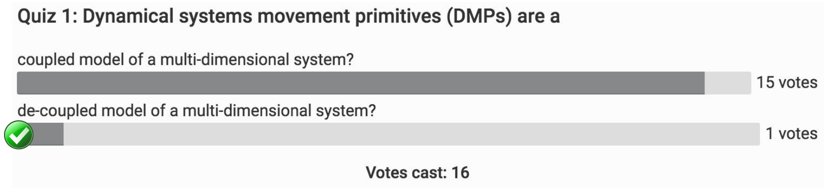
Can you answer the following questions?

- What is Bernstein’s motor equivalence problem?
- Which typical steps are needed to classify motor skills given EMG signals?
- What are time-invariant and what are time-varying muscle synergies?
- How can muscle synergies be embedded in dynamical attractor systems?
- What are probabilistic trajectory models?

Summary of II.2-3 Movement primitives

- Our **musculoskeletal system** as well as modern robots have **more degrees of freedom than needed** to perform a specific skill [Bernstein, 1967].
- The **key difference** between time-varying and time-invariant muscle synergies are that the **temporal time-shift parameters** per task and per synergy results in simpler synergies compared to the complex shapes of the temporal mixing coefficients.
- Probabilistic trajectory models (PTMs) model a **distribution over trajectories**, whereas DMPs encode a single trajectory.
- PTMs are **coupled model** of multi-dimensional data.

Recap of the previous lecture



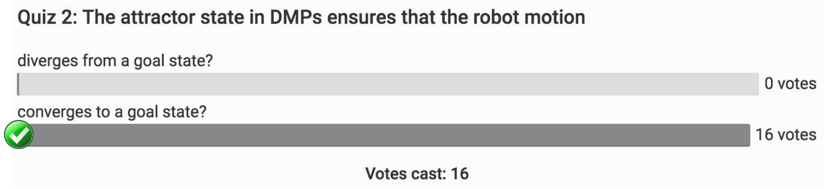
Quiz 1: Dynamical systems movement primitives (DMPs) are a

coupled model of a multi-dimensional system? — 15 votes

✓ de-coupled model of a multi-dimensional system? — 1 votes

Votes cast: 16

Recap of the previous lecture



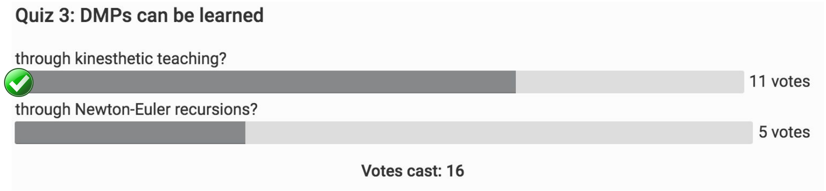
Quiz 2: The attractor state in DMPs ensures that the robot motion

diverges from a goal state? 0 votes

✓ converges to a goal state? 16 votes

Votes cast: 16

Recap of the previous lecture



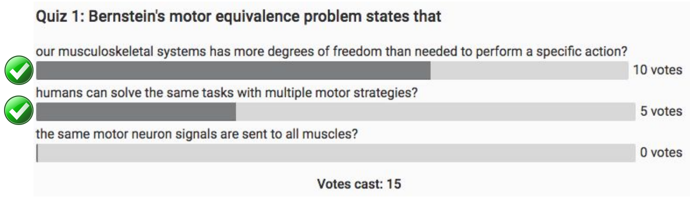
Quiz 3: DMPs can be learned

✓ through kinesthetic teaching? 11 votes

through Newton-Euler recursions? 5 votes

Votes cast: 16

Recap of the previous lecture



Quiz 1: Bernstein's motor equivalence problem states that

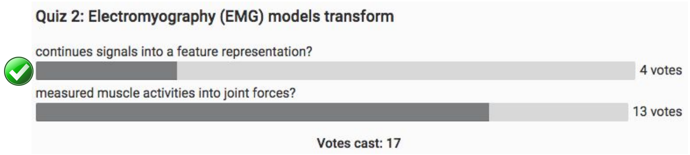
✓ our musculoskeletal systems has more degrees of freedom than needed to perform a specific action? 10 votes

✓ humans can solve the same tasks with multiple motor strategies? 5 votes

the same motor neuron signals are sent to all muscles? 0 votes

Votes cast: 15

Recap of the previous lecture



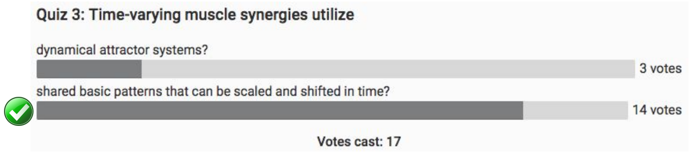
Quiz 2: Electromyography (EMG) models transform

✓ continues signals into a feature representation? 4 votes

measured muscle activities into joint forces? 13 votes

Votes cast: 17

Recap of the previous lecture



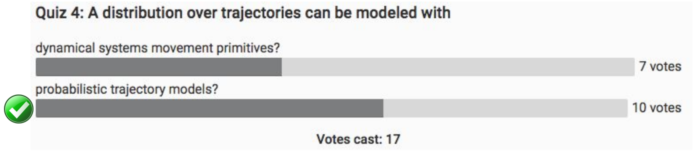
Quiz 3: Time-varying muscle synergies utilize

dynamical attractor systems? 3 votes

✓ shared basic patterns that can be scaled and shifted in time? 14 votes

Votes cast: 17

Recap of the previous lecture



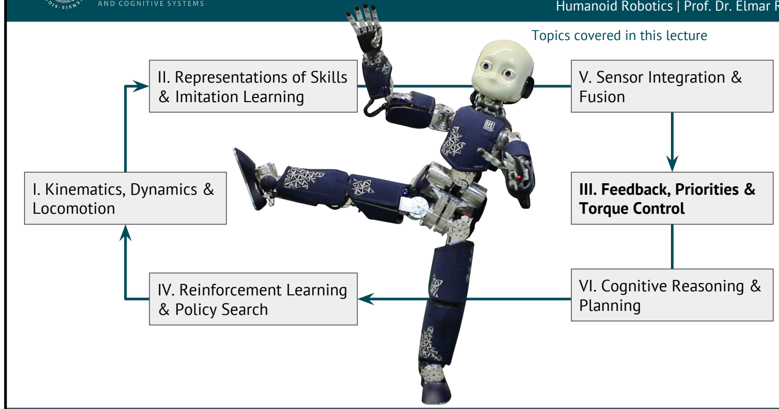
Quiz 4: A distribution over trajectories can be modeled with

dynamical systems movement primitives? 7 votes

✓ probabilistic trajectory models? 10 votes

Votes cast: 17

Topics covered in this lecture



I. Kinematics, Dynamics & Locomotion

II. Representations of Skills & Imitation Learning

III. Feedback, Priorities & Torque Control

IV. Reinforcement Learning & Policy Search

V. Sensor Integration & Fusion

VI. Cognitive Reasoning & Planning

Learning Objectives of Chapter II (*Lernziele* in german).

In today's lecture you will learn …

- how the P-, the I- and the D-term in a PID controller works.
- how classical robot feedback control is implemented and tuned.
- how optimal feedback control laws can be derived.
- how task priorities can be modelled.

Summary of III.1 Feedback control

- PID control consists of a proportional, an integral and a derivative term.
- Tuning the gains is challenging in robotics.
- Instead of the integral term, a gravity model should be used if available.

Link to Marc Toussaint's slides on Robotics, For PID control review p. 37 - 42

PID control explained for car control.

In today's lecture you will learn ...

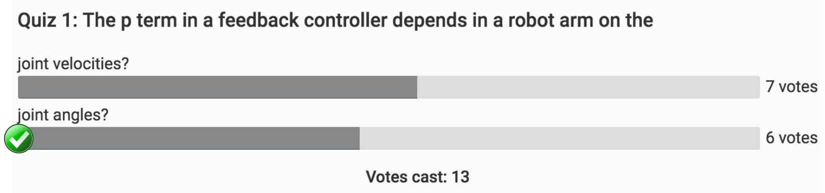
- how the P-, the I- and the D-term in a PID controller works.
- how classical robot feedback control is implemented and tuned.
- how optimal feedback control laws can be derived.
- how task priorities can be modelled.

Summary of III.2 Feedback control

- A Linear Quadratic Gaussian Regulator assumes linear dynamics, quadratic costs and Gaussian noise.
- The control law is the result of an optimization process where the energy of the controlled output and the inputs (the control signals) is minimized.
- In a PID controller usually only the diagonals are used whereas a LQR controller all elements in the gain matrix are learned.
- Extensions of the LQR consider more complex cost functions, drift terms and non-linear dynamics through iterative linearizations.

Link to more details on how to derive the most simple LQR controller

Recap of the previous lecture



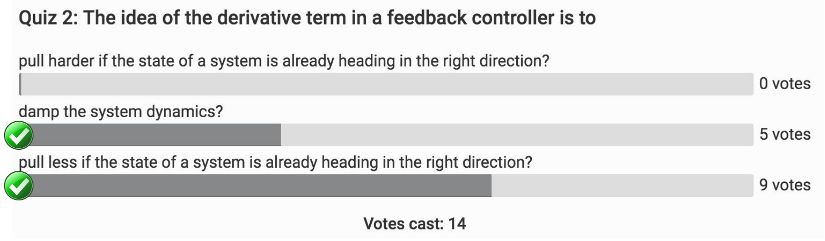
Quiz 1: The p term in a feedback controller depends in a robot arm on the

joint velocities? 7 votes

✓ joint angles? 6 votes

Votes cast: 13

Recap of the previous lecture



Quiz 2: The idea of the derivative term in a feedback controller is to

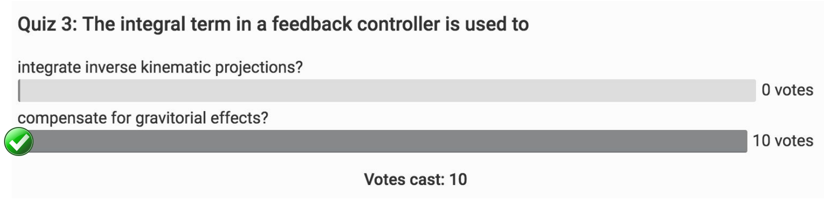
pull harder if the state of a system is already heading in the right direction? 0 votes

✓ damp the system dynamics? 5 votes

✓ pull less if the state of a system is already heading in the right direction? 9 votes

Votes cast: 14

Recap of the previous lecture



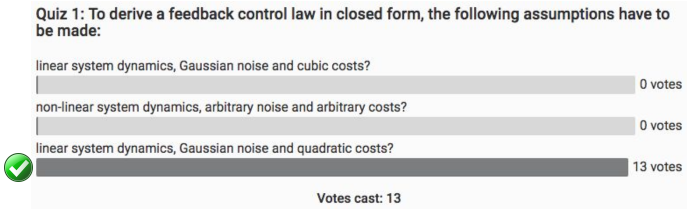
Quiz 3: The integral term in a feedback controller is used to

integrate inverse kinematic projections? 0 votes

✓ compensate for gravitorial effects? 10 votes

Votes cast: 10

Recap of the previous lecture



Quiz 1: To derive a feedback control law in closed form, the following assumptions have to be made:

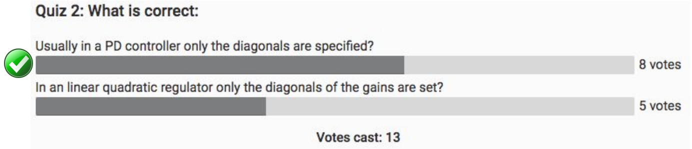
linear system dynamics, Gaussian noise and cubic costs? 0 votes

non-linear system dynamics, arbitrary noise and arbitrary costs? 0 votes

✓ linear system dynamics, Gaussian noise and quadratic costs? 13 votes

Votes cast: 13

Recap of the previous lecture



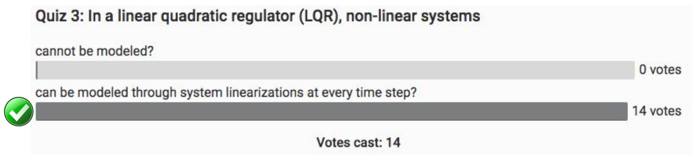
Quiz 2: What is correct:

✓ Usually in a PD controller only the diagonals are specified? 8 votes

In an linear quadratic regulator only the diagonals of the gains are set? 5 votes

Votes cast: 13

Recap of the previous lecture



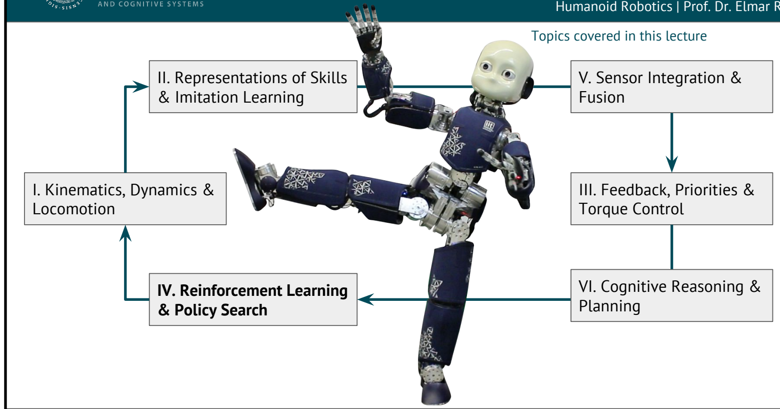
Quiz 3: In a linear quadratic regulator (LQR), non-linear systems

cannot be modeled? — 0 votes

✓ can be modeled through system linearizations at every time step? — 14 votes

Votes cast: 14

Topics covered in this lecture



I. Kinematics, Dynamics & Locomotion

II. Representations of Skills & Imitation Learning

V. Sensor Integration & Fusion

III. Feedback, Priorities & Torque Control

VI. Cognitive Reasoning & Planning

IV. Reinforcement Learning & Policy Search

Learning Objectives of Chapter IV (*Lernziele* in german).

In today's lecture you will learn ...

- about the underlying principles in reinforcement learning (RL).
- to describe Markov Processes and Markov Reward Processes.

Summary of the lessons learnt so far

- In RL, we assume that all goals can be achieved through maximizing a scalar reward.
- For sequential decision making we define a state instead of using a full history.
- The state definition needs to be sufficiently informative to represent the task.

Let's use the state and the reward for decision making!

Summary of the lessons learnt so far

- A Markov process describes the transitions between states which have the Markov property.
- By adding a reward we can quantify the “quality” of generated sequences.
- The total cumulative reward is called return.
- Using the return we can compute the Value function for every state.

We can now use the value function in a Markov Reward Process!

Summary of IV.1 Optimality principles in RL

- The learning from rewards concept is a powerful approach used in many research disciplines.
- The idea is to learn from interactions with the environment.
- The challenge is to define a sufficiently complete state variable and a reward function that encapsulates all goals.
- By using the Markov property closed form solutions as well as iterative algorithms can be derived.

In today’s lecture you will learn …

- to describe ~~Markov Processes, Markov Reward Processes~~ and Markov Decision Processes.
- how to implement a simple reinforcement learning algorithm called Value Iteration.
- about Q-learning and policy iteration.

Summary of the lessons learnt so far

- The value function has two parts: the **immediate reward** and the (discounted) **value of the successor state.**
- It is also called the Bellman equation.
- For small MRPs the Value function can be computed in closed form in O(3) (matrix inverse).
- However, in practice iterative approaches are used.

What about actions?

Summary of the lessons learnt

- For any MPD there exists at least one optimal policy.
- All optimal policies achieve optimal V or Q-functions.
- The value iteration algorithm iteratively updates the values of all states.
- The q-learning algorithm iteratively updates the q-function using the optimal future value.
- The policy improvement algorithm iterates between a policy evaluation phase (the learning of the V- and Q-functions) and a policy improvement step.

Summary of IV.1 Optimality principles in RL

- By using the Markov property closed form solutions as well as iterative algorithms can be derived.
- We discussed Value Iteration, Q-Learning and Policy Iteration.
- RL algorithms can be
  - model-free (Q-learning)
  - or model-based and use $\mathcal{P}^a_{ss'}$ (Value Iteration, Policy Iteration)

- Many sophisticated extensions based on function approximations and neural networks are used in practice. The last two lectures were just a brief introduction to RL!

Learning Objectives of Chapter IV (*Lernziele* in german).

In today's lecture you will learn ...

- about the goal definition of policy search methods.
- to understand gradient descent and how to apply it for policy search.
- where to find more resources to learn about policy search.

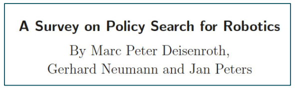
Finite Differences Policy Search

- Directly perturbs the policy parameters (*episode based exploration* scheme vs. step based per time step).
- The policy gradient is estimated numerically through finite differences (model-free approach).
- **Pros:** Black box method that does not make any assumptions on the reward function or policy model.
- **Cons:** Very sensitive to noise in the parameter space. Scales badly with the number of policy parameters. Exploration can get stuck in local minima or harm the robot.

If you want to learn how to do this correctly, visit my lecture on **Probabilistic Learning for Robotics (RO5601)**

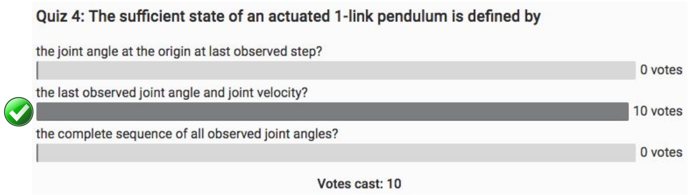
Summary of IV.3 Policy Gradient Methods

- Gradient descent is a powerful optimization technique.
- It can be used to learn parameterized policies.
- The challenges are to determine the step size (e.g., adaptive or via a Hessian) and to compute the gradient.
- Many algorithms make no assumption about the policy, so called **black box methods**, and numerically estimate the gradient.
- **White box methods** exploit this knowledge but are not discussed here.



A Survey on Policy Search for Robotics
By Marc Peter Deisenroth, Gerhard Neumann and Jan Peters

The definition of the state is crucial



Quiz 4: The sufficient state of an actuated 1-link pendulum is defined by

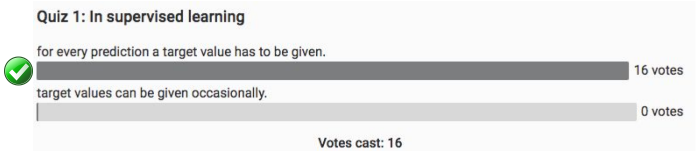
the joint angle at the origin at last observed step? 0 votes

✓ the last observed joint angle and joint velocity? 10 votes

the complete sequence of all observed joint angles? 0 votes

Votes cast: 10

Recap of the previous lecture



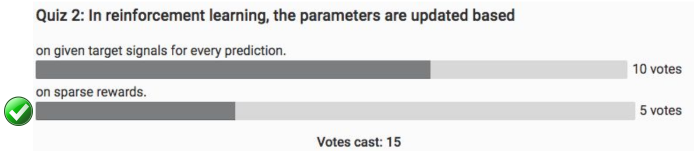
Quiz 1: In supervised learning

✓ for every prediction a target value has to be given. 16 votes

target values can be given occasionally. 0 votes

Votes cast: 16

Recap of the previous lecture



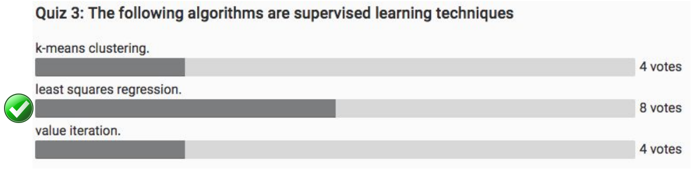
Quiz 2: In reinforcement learning, the parameters are updated based

on given target signals for every prediction. 10 votes

✓ on sparse rewards. 5 votes

Votes cast: 15

Recap of the previous lecture



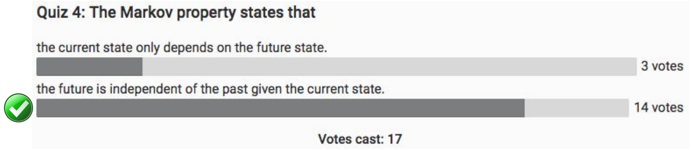
Quiz 3: The following algorithms are supervised learning techniques

k-means clustering. 4 votes

✓ least squares regression. 8 votes

value iteration. 4 votes

Recap of the previous lecture



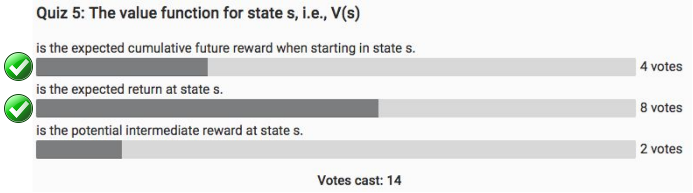
Quiz 4: The Markov property states that

the current state only depends on the future state. 3 votes

✓ the future is independent of the past given the current state. 14 votes

Votes cast: 17

Recap: What is the Value function?



Quiz 5: The value function for state s, i.e., V(s)

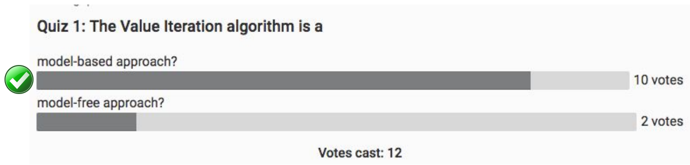
✓ is the expected cumulative future reward when starting in state s. 4 votes

✓ is the expected return at state s. 8 votes

is the potential intermediate reward at state s. 2 votes

Votes cast: 14

Recap of the previous lecture



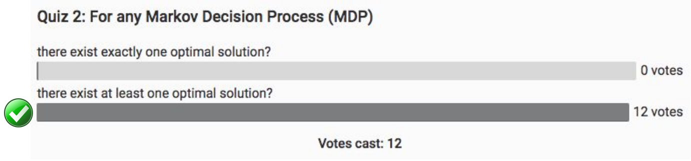
Quiz 1: The Value Iteration algorithm is a

✓ model-based approach? 10 votes

model-free approach? 2 votes

Votes cast: 12

Recap of the previous lecture



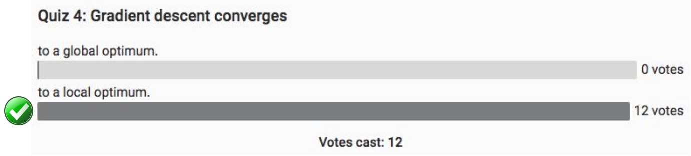
Quiz 2: For any Markov Decision Process (MDP)

there exist exactly one optimal solution? — 0 votes

✓ there exist at least one optimal solution? — 12 votes

Votes cast: 12

Recap of the previous lecture



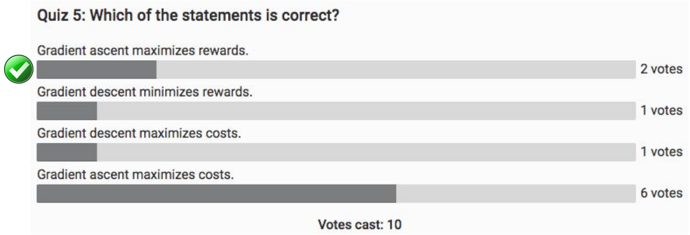
Quiz 4: Gradient descent converges

to a global optimum. — 0 votes

✓ to a local optimum. — 12 votes

Votes cast: 12

Recap of what we have learned so far



Quiz 5: Which of the statements is correct?

✓ Gradient ascent maximizes rewards. — 2 votes

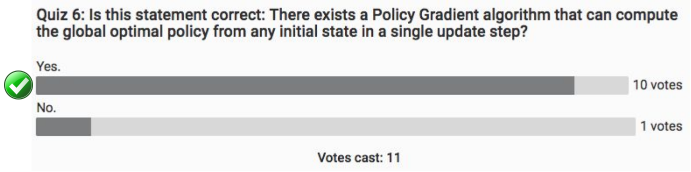
Gradient descent minimizes rewards. — 1 votes

Gradient descent maximizes costs. — 1 votes

Gradient ascent maximizes costs. — 6 votes

Votes cast: 10

Recap of what we have learned so far



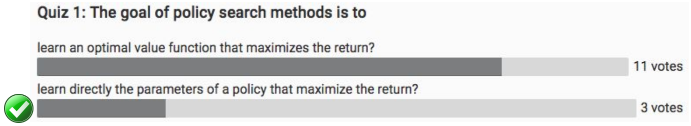
Quiz 6: Is this statement correct: There exists a Policy Gradient algorithm that can compute the global optimal policy from any initial state in a single update step?

✓ Yes. — 10 votes

No. — 1 votes

Votes cast: 11

Recap of what we have learned so far

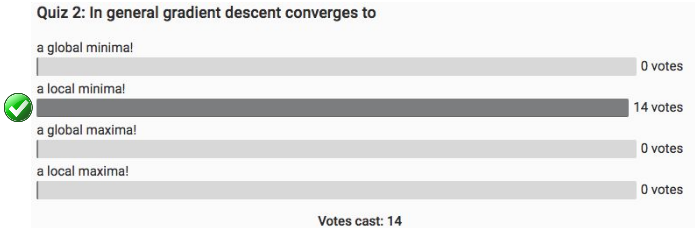


Quiz 1: The goal of policy search methods is to

learn an optimal value function that maximizes the return? 11 votes

✓ learn directly the parameters of a policy that maximize the return? 3 votes

Recap of what we have learned so far



Quiz 2: In general gradient descent converges to

a global minima! 0 votes

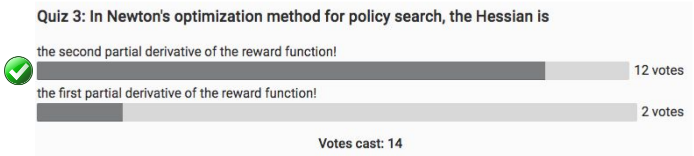
✓ a local minima! 14 votes

a global maxima! 0 votes

a local maxima! 0 votes

Votes cast: 14

Recap of what we have learned so far



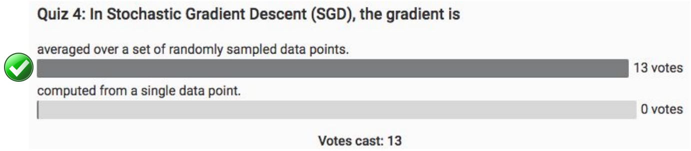
Quiz 3: In Newton's optimization method for policy search, the Hessian is

✓ the second partial derivative of the reward function! 12 votes

the first partial derivative of the reward function! 2 votes

Votes cast: 14

Recap of what we have learned so far



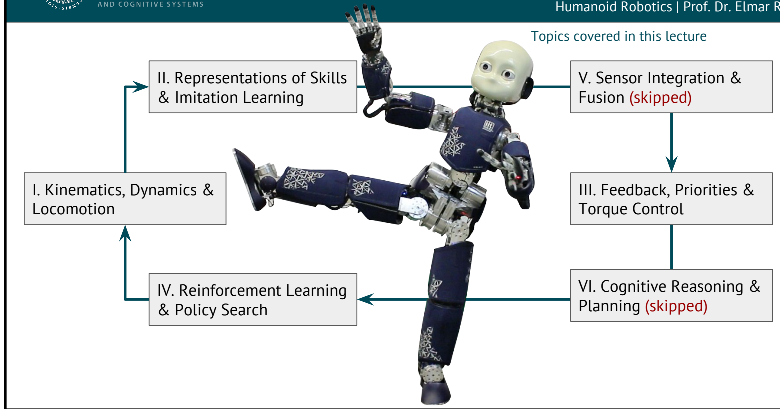
Quiz 4: In Stochastic Gradient Descent (SGD), the gradient is

✓ averaged over a set of randomly sampled data points. 13 votes

computed from a single data point. 0 votes

Votes cast: 13

Topics covered in this lecture


I. Kinematics, Dynamics & Locomotion

II. Representations of Skills & Imitation Learning

III. Feedback, Priorities & Torque Control

IV. Reinforcement Learning & Policy Search

V. Sensor Integration & Fusion (skipped)

VI. Cognitive Reasoning & Planning (skipped)

How to contact me

Thank you for your attention!

Contact:

Universität zu Lübeck
Institute for Robotics and Cognitive Systems
Ratzeburger Allee 160
Building 64, Room 94
23538 Lübeck, Deutschland

Telefon: +49 (0) 451 3101 5209
E-Mail: rueckert@rob.uni-luebeck.de

Disclaimer:

The lecture notes posted on this website are for personal use only. The material is intended for educational purposes only. Reproduction of the material for any purposes other than what is intended is prohibited. The content is to be used for educational and non-commercial purposes only and is not to be changed, altered, or used for any commercial endeavor without the express written permission of Professor Rueckert.