KIT-Department of Informatics
Prof. Dr.-Ing. Tamim Asfour

Exam Solution Sheet

Robotics II: Humanoid Robotics

am September 24, 2020, 09:00 – 10:00

Family name:	Given name:	Matriculation number:
David	Marr	978-0262514620

Exercise 1	7 out of 7 points
Exercise 2	8 out of 8 points
Exercise 3	10 out of 10 points
Exercise 4	8 out of 8 points
Exercise 5	12 out of 12 points

Total:	45 out of 45 points

	Grade: **1.0**

Exercise 1 *Humanoid Robots*

1. Humanoid robots:
   ARMAR robots, Atlas, iCub, Sofia, HRP series, Toro, Justin, Robonaut, Valkyrie, P2, Petman, Ocean One, Momaro, Nextage, Wabot series, Wabian, Partner Robots, Asimo, Toyota T-HR3, Sarcos, Nao, Pepper, KHR, Hubo, ...

2. Advantages/Disadvantages:

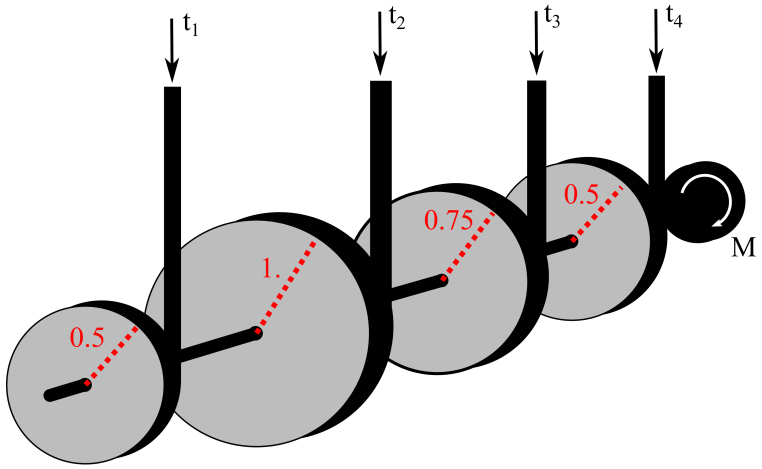
   - Advantages: Versatility (can perform a wide variety of tasks in human environments); Predictability (better prediction of robot actions); Acceptance (human-like appearance may support acceptance and intuitive human-robot interaction)
   - Disadvantages: Very complex; Uncanny Valley (human-like appearance may result into a negative emotional response)

3. Models:

   - Kinematic model including joints and segment lengths
   - Dynamic model (e.g. with mass, CoMs and moments of inertia)
   - Statistic/Anthropomorphic model with segment properties (e.g. length, mass) defined as a function of global parameters (e.g. body height, body weight)

Exercise 2 *Grasping Synergies and Eigengrasps*

1. Mechanism for the first eigengrasp:



2. Mechanism:

   Adds two scalar values so that:
   $z_{i,j} = \frac{1}{2}(y_{i,1,j} + y_{i,2,j})$

3. Amplitude vector **a**

   Approach: $a_1 \cdot \mathbf{e}_1 + a_2 \cdot \mathbf{e}_2 = \mathbf{p}$, with $\mathbf{a} = \begin{bmatrix} a_1 & a_2 \end{bmatrix}^T$.

   Solving for **a** yields (one possible way):

$$\begin{cases} \frac{1}{2}a_1 + \frac{3}{5}a_2 = \frac{3}{4} \;\Rightarrow\; a_1 = \frac{3}{2} - \frac{6}{5}a_2 \\ a_1 + \frac{7}{10}a_2 = \frac{5}{4} \;\Rightarrow\; a_1 = \frac{5}{4} - \frac{7}{10}a_2 \end{cases}$$

$$\Rightarrow a_1 = \frac{3}{2} - \frac{6}{5}a_2 = \frac{5}{4} - \frac{7}{10}a_2$$

$$\Rightarrow \frac{1}{2}a_2 = \frac{1}{4} \;\Rightarrow\; a_2 = \frac{1}{2}$$

$$\Rightarrow a_1 = \frac{9}{10}$$

$$\Rightarrow \mathbf{a} = \begin{bmatrix} 0.9 & 0.5 \end{bmatrix}^T.$$

4. Soft Synergy Model:

   - It uses a combination of two force fields to control the physical hand.
   - One field is attracting the physical hand towards a virtual hand (which is moving on the synergy manifold).
   - The other field is repelling the hand from penetrating the object.
   - The dynamical equilibrium between those two fields is found depending on the stiffness of the hand actuation and control system.

   Solved actuation problem:
   Using the soft synergy model a stable grasp is reachable even if one finger is reaching contact before the other fingers.

Exercise 3 *Grasping*

1. (a) Two other object classes:
   - Known objects
   - Familiar objects

   (b) Object knowledge of each class:
   - Known objects: Known object geometry, i.e. we have a complete geometric object model (mesh) for each object
   - Familiar objects: The class of objects is known (e.g. bottle). [This grasp knowledge from known class members can be used for new objects]

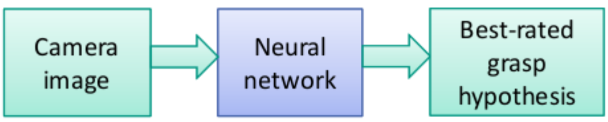
   (c) Approach to grasp an object (for each class):
   - Necessary steps to grasp **known objects** with a database:
     – Scene segmentation
     – Object recognition
     – Pose estimation
     – Grasp selection and reachability filtering
   - Necessary steps to grasp **familiar objects** with a database:
     – Scene segmentation
     – Feature extraction
     – Comparison to known examples (to retrieve possible grasp candidates)
     – Grasp selection and reachability filtering

2. Steps required:

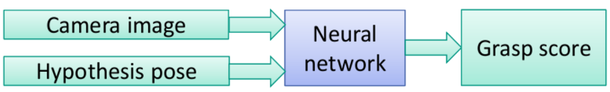
   - Given the segmented point cloud, shape fitting/approximation is conducted which results in a shape (e.g. box)
   - Grasps for the approximated shape are generated (e.g. based on a heuristic)
   - The resulting grasp candidates simulated and ranked (e.g. according to a stability metric)
   - The resulting grasp hypotheses are filtered w.r.t. the robot's reachability

3. (a) Two ways:
   - Learning by demonstration (human teacher)
   - Training data collection on the target system (the real robot)
   - Training data generation in simulation
   - Hand-labled data

   (b) Block diagram and description:
   Three examples:
   - Regression:
     - No pre/post processing required but possible
     - Directly feed camera image into a convolutional neural network (CNN) and predict the best grasp
     - Example: Schmidt et al.

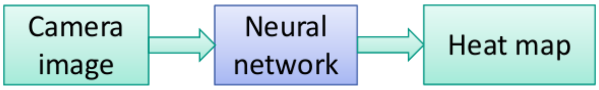


   - Discriminative:
     - Rate a grasp hypothesis according to sensor data
     - Convolutional neural network (CNN) estimates the quality of a grasp based on incomplete information
     - Assumption: Network will learn to internally complete the missing information
     - Example: Metrics, Mahler et al., 2015



   - Heatmap:

- Maps images to images
- For each pixel in the input image rate the grasp quality
- Use image to image techniques from computer vision
- Select pixel with highest predicted grasp score for execution
- Example: Nguyen et al., 2016



Exercise 4 *Active Perception*

1. Discuss the difference:
   Interactive Perception includes image processing, viewpoint selection, multi-modal sensory input, changing the agent's state and interacting with the environment to create novel sensory signals whereas in Active Vision the environment is not changed and only vision is used as sensory input, i.e. there is no multi-modal sensory input. Classical Computer Vision only makes use of image processing.

2. Heuristics for object discovery
   - textured objects: Planes, cylinders and spheres amongst SIFT features (RANSAC)
   - single colored objects: Unicolored regions of promising size (color MSERs (Maximally stable extremal regions)
   - neither textured nor unicolored: Visually salient regions (Difference of Gaussians filter)

3. Exploration and reconstruction of unknown objects
   - psi_1: Maximize Δ information: Reduce uncertainty
   - ψ_2: Stay local: Prefer targets, that are close
   - $\psi_{3,pos}$and $\psi_{3,rot}$: Minimize path cost: Minimize movement and rotation of the hand
   - With ψ_1 only: Standard approach. The robot would select targets that maximize the information gain. In general, these targets are far away from the current target.

Exercise 5 *Imitation Learning*

1. Mirror neurons:
   Mirror neurons are nerve cells that have been identified in the brain of humans (and other primates). They are equally active both during observation and during the execution of a particular activity. Mirror neurons thus connect the perception of an action with its execution and are active during the entire observation/execution.

2. Challenges:

   - Challenges of the observation step include:
     - How to estimate the poses of the demonstrator and the objects
     - How to identify the demonstrator (e.g. when multiple persons are recognized)
     - How to identify the objects involved to the demonstration
     - When does the demonstration starts, when does it end and is the observation appropriate in the current context
     - What aspects of the demonstration are essential for the task
     - How to identify already stored demonstrations
   - Challenges of the generalization step include:
     - How to segment / identify meaningful pieces from the observed demonstration
     - How to replicate the observed demonstration on own embodiment with e.g. different kinematic chains or missing limbs
     - How to extract constraints from the observed demonstration
   - Challenges of the reproduction step include:
     - How to detect if an executed action succeeded or not
     - How to reason about the correct execution, if an action fails and how to adapt the motion primitive

3. Extracted information:
   Object poses, object shapes (may be known in advance) and the demonstrator poses must be extracted in order to identify e.g. spatial / contact relations in the scene.

4. Task constraints:

   - Preconditions of actions / Order of actions / Temporal constraints:
     Simple symbolic representations, where actions have (temporal) preconditions and effects. Action A and B depend on each other if B is only applicable if action A has been started/is running/ ended
   - Spatial constraints:
     Can be represented as a constraint of the resulting motion primitive. Consider e.g. the case where a robot should carry a cup and the cup must be carried upright which means that the pose of the cup is constrained.

5. Hierarchical segmentation:

- First Level: Motion Segmentation: Segmented into most distinctive parts based on the motion characteristic. It uses a heuristic based on the acceleration profile, an iterative search for the best keyframe using a sliding window and an recursive segmentation until segment size or quality too small.
- Second Level: Semantic Segmentation: Extraction of hand-objects and objectobject contact relation changes between using 3D mesh models and collision detection algorithms.

6. - Canonical system:

$$\tau \dot{u} = -\alpha u$$

    - Describes the state u of the DMP in time
    - Changing τ => change the motion speed
    - Drives the perturbation to control the transformation system

   - Transformation system:

$$\tau \dot{v} = K(g - x) - Dv + (g - x_0)f(u)$$

$$\tau \dot{x} = v$$

    - (g : goal
      x : current position
      v: current velocity
      D : damping constant
      K: spring constant
      τ: temporal factor)
    - Consists of Damped Spring-Mass System (Spring term: $K(g - x)$, Damping term: $D\dot{x}$. Their difference is transformed to first-order system) and perturbation force term
    - The perturbation function f determines the shape of the trajectory
    - The perturbation force term $(g - x_o)f(u)$ is learned from demonstration

7. Can it be learned by multiple demonstrations?
   No, classical DMPs can only learn from a single demonstration because the shape characteristic is learned with respect to the demonstrated trajectory (position and velocity for each time step). But other approaches such as *Via-points Motion Primitives* can learn from multiple demonstrations.