Humanoid Robotics

Broad Overview & Sample Exam Questions

Maren Bennewitz
Oral Exam

- July 15-17
- If you are admitted you should have got a time slot
- If you don’t want to participate anymore or if you think you fail anyway, please send an email to pkulkarni@cs.uni-bonn.de (we will be less annoyed and you will anyway get a 5.0 for the first exam)
About These Slides

- Most important topics
- No guarantee for completeness regarding the exam
- Can be seen as basis for a good grade
Least Squares
Problem Definition

- Let
  - $\mathbf{x}$ be the state vector (unknown)
  - $\mathbf{z}_i$ be a measurement of the state $\mathbf{x}$
  - $\hat{\mathbf{z}}_i = f_i(\mathbf{x})$ be a function that maps $\mathbf{x}$ to a predicted measurement $\hat{\mathbf{z}}_i$
- Given $n$ noisy measurements $\mathbf{z}_{1:n}$ about the state $\mathbf{x}$
- **Goal:** Estimate the state $\mathbf{x}$ that best explains the measurements $\mathbf{z}_{1:n}$
Error Function

- Error $e_i$ is the **difference** between the **predicted** and the **actual** measurement

  $$e_i(x) = z_i - f_i(x)$$

- Assumption: The error has **zero mean** and is **normally distributed**
- Gaussian error with information matrix $\Omega_i$
- The **squared error** of a measurement depends on the state and is a scalar

  $$e_i(x) = e_i(x)^T \Omega_i e_i(x)$$
Goal: Find the Minimum

- Find the state $\mathbf{x}^*$ that minimizes the error over all measurements

\[
\begin{align*}
\mathbf{x}^* & \quad = \quad \arg\min_{\mathbf{x}} F(\mathbf{x}) & \text{global error (scalar)} \\
& \quad = \quad \arg\min_{\mathbf{x}} \sum_i e_i(\mathbf{x}) & \text{squared error terms (scalar)} \\
& \quad = \quad \arg\min_{\mathbf{x}} \sum_i e_i^T(\mathbf{x}) \Omega_i e_i(\mathbf{x}) & \text{error terms (vector)}
\end{align*}
\]
Solve via Iterative Local Linearizations (Gauss-Newton)

- Linearize the error terms around the current solution/initial guess
- Compute the first derivative of the approximated global error function
- Set it to zero and solve the linear system
- Obtain the new state (that is hopefully closer to the minimum)
- Iterate
Minimize the Global Error

- Approximate the error functions around an current guess $x$ via **Taylor expansion**
  \[
  e_i(x + \Delta x) \approx e_i(x) + J_i(x) \Delta x
  \]

- Write the global error as a quadratic form
- Compute the derivative, set it to zero, solve the linear system
Questions

- Describe the class of problems to which least squares can be applied
- How is the error defined?
- Sketch the error minimization process with Gauss Newton
- Write down the equations and explain how the best increment is obtained
Projective Geometry
Homogenous Coordinates
Homogeneous Coordinates

- H.C. can simplify mathematical expressions
- Allow for easy chaining and inversion of transformations
- Modeled through an extra dimension
Questions

- How can homogeneous coordinates be obtained from Euclidean coordinates, and vice versa?
- Why do we use homogeneous coordinates instead of Euclidean coordinates?
- How does a rigid body transformation look like?
Camera Calibration
Calibration

- Assumption: pinhole camera model
- A camera projects 3D world points onto the 2D image plane
- Calibration: Finding the parameters of the camera that affect this process
- **Extrinsics**: describe the pose of the camera in the world
- **Intrinsics**: describe the mapping of the scene in front of the camera to the pixels on the sensor
Questions

- Describe the pinhole camera model
- What are the assumptions in this model?
- Which parameters do we need to calibrate?
- Write down the calibration matrix for the intrinsic parameters
Question: Describe the mapping
Humanoid Calibration
Parameters to Estimate

- True position of a joint: \( q = \tilde{q} + q^{\text{off}} \)

- Estimate joint offsets:

\[
q^{\text{off}} = \begin{pmatrix}
q_{1}^{\text{off}} \\
q_{2}^{\text{off}} \\
\vdots \\
q_{n}^{\text{off}}
\end{pmatrix}
\]

- Camera extrinsics (relative to the reference frame neck joint): \( R, C \)

- Camera intrinsics: \( f_x, f_y, x_H, y_H, \kappa \)
Question: Which kind of parameters to estimate?

- Joint offsets
- Extrinsic camera parameters
- Intrinsic camera parameters
- End-effector marker poses
Question: Define the error function

\[ e_i(\theta, z_i, \hat{q}_i) = z_i - \text{predictmarker}_{M_{EEF}}(\theta, \hat{q}_i) \]

- **Measurement:** observed marker in the image
- **Current values of calibration parameters:**
- **Joint readings:**
- **Predicted marker location:** given the kinematic structure
Question: How is the predicted marker location computed?

\[
predictmarker_{M_{EEF}}(\theta, \hat{q}_i) = K_\theta \left[ R_\theta \right] - R_\theta t_\theta \mathcal{F}^N_{EEF}(\theta, \hat{q}_i) \hat{m}_{EEF}
\]

- **Calibration matrix** with current estimates of intrinsic camera parameters
- **Current estimates** of extrinsic camera parameters
- **Forward kinematics**: transformation from the EEF frame into the neck frame from joint encoder readings and current estimates of the offsets
Question: Explain forward kinematics

The complete transform between $E$ and $B$ corresponding to forward kinematics is given by

$$\mathcal{F}^B_E(\theta, \hat{q}) = T^B_0 \left( \prod A^{-1}_i(q_i) \right) T^n_E$$

- transformation depending on joint encoder readings and current estimates of the offsets
- transformation matrix for the transform between the joints $i$ and $i - 1$ given $q_i$
- position of joint $i$ according to $\hat{q}$ and $\theta$

[see exercise for the computation of $A^{-1}_i(q_i)$]
3D World Representations
Questions

- Which 3D world representations exist? Briefly explain them.
- What are their advantages / drawbacks?
- Explain the ICP algorithm and how the error function is defined
- How can the SVD be used to find the best rotation and translation?
Monte Carlo Localization
Question: Derive the Recursive Bayes Filter

\[ \text{bel}(x_t) = p(x_t \mid z_{1:t}, u_{1:t}) \]

\[ = \eta p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t}) \]

\[ = \eta p(z_t \mid x_t) p(x_t \mid z_{1:t-1}, u_{1:t}) \]

\[ = \eta p(z_t \mid x_t) \int p(x_{t-1}, z_{1:t-1}, u_{1:t}) \]

\[ p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \, dx_{t-1} \]

\[ = \eta p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_t) p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \, dx_{t-1} \]

\[ = \eta p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_t) p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1}) \, dx_{t-1} \]

\[ = \eta p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_t) \text{bel}(x_{t-1}) \, dx_{t-1} \]

observation model \quad motion model
Monte Carlo Localization

- Each particle is a pose hypothesis
- **Prediction**: For each particle, sample a new pose from the motion model
  \[ x_t[j] \sim p(x_t \mid x_{t-1}^j, u_t) \]
- **Correction**: Weight samples according to the observation model
  \[ w_t[j] \propto p(z_t \mid x_t^j) \]
- **Resampling**: Draw sample \( i \) with probability \( w_t^{[i]} \) and repeat \( J \) times \( (J = \#\text{particles}) \)
Localization for Humanoids

3D environments require a 6D pose estimate

\[ x = (x, y, z, \varphi, \theta, \psi) \]

2D position, height, yaw, pitch, roll

estimate the 6D torso pose
Kinematic Walking Odometry

Question: How is the odometry estimate $u_t$ from two consecutive torso poses computed?

Figure 3.2: Odometry estimate $u_t$ from two consecutive torso poses during the walking cycle.

Only noisy estimates of the odometry are available while walking and a substantial amount of drift accumulates over time. Consequently, a particle filter has to account for that noise with a higher variance, requiring a higher number of particles and thus more computational power for successful pose estimation. By learning the motion model parameters instead, the localization performance can be increased, both in terms of computational load as well as accuracy.

Here, we consider the most general case of any kind of 3D positional and rotational displacement, for instance originating from an omnidirectional walking engine. We furthermore assume that systematic drift affects the motion reported by odometry in the 2D plane, i.e., only $(e_x, e_y)$ are affected from $e_x = (e_x, e_y, e_z, e_j, e_q, e_y)$. This is not a strong restriction as long as the humanoid walks on a solid surface, since its motion is constrained by this surface and it cannot leave the ground. Even when climbing challenging terrain such as stairs, the drift of the motion occurs in the 2D plane of the stance leg as long as the robot does not fall or slide down a slope. General noise in the kinematic estimate of the humanoid's height above the ground does not lead to a systematic drift.

3.2.1. Motion Model Calibration

For odometry calibration, we will refer to the reduced state vectors containing 2D position and orientation as $x_0 = (x, y, y)$. Corresponding to Eq. (3.8), $u_0 t = (u_x t, u_y t, u_y t)$ estimates the displacement between two poses reported by odometry $e_x 0 t = e_x 0 t + u_0 t$.

To calibrate the drift of $u_0 t$, we assume that a ground truth pose $x_0$ is available in a prior learning phase, e.g., from an external motion capture system, scan matching, or visual odometry. Based on the deviations from the ground truth, values of a calibration matrix $M_{2 \times 3}$ can be determined to correct the 2D drift of odometry, such that $x_0 t = x_0 t + Mu_0 t$.
Observation Model $p(o_t \mid x_t)$

$p(o_t \mid x_t) = p(r_t, \tilde{z}_t, \tilde{\varphi}_t, \tilde{\psi}_t \mid x_t) =$

$p(r_t \mid x_t) \cdot p(\tilde{z}_t \mid x_t) \cdot p(\tilde{\varphi}_t \mid x_t) \cdot p(\tilde{\psi}_t \mid x_t)$

- **Range data**: Ray-casting or endpoint model in 3D map
- **Torso height**: Compare measured value from kinematics to predicted height (accord. to motion model)
- **IMU data**: Compare measured roll and pitch to the predicted angles
- **Use individual Gaussians to evaluate the difference**
Questions

- What are the three steps of the particle filter?
- What are the components of the state space of our humanoid?
- What are the components of the observation model?
- How are the individual terms computed?
Path Planning and Walking
Footstep Planning with A*

- Planning of footstep locations by building up a search tree of potential successor states
- Fixed set of possible relative footstep changes
- Foot placements are checked for collisions during tree expansion

source: Kuffner et al.
Heuristics for Footstep Planning

- Critical for A* performance
- Estimate the costs to the goal taking into account the largest possible forward step
- Usual heuristic functions are based on:
  - Euclidean distance (straight line)
  - Dijkstra path cost (shortest 2D path) with safety margin around obstacles

Dijkstra heuristic is inadmissible for humanoids!
Footstep Planning with A*

Small set → fast planning
limited search space

Large set → large coverage
long planning time
Adaptive Node Expansion

- Add only a small set of nodes at each expansion step
- Systematically search for valid successors

- Leads to a high success rate, short paths, and fast planning times
Questions

- How does A* for footstep planning work?
- What are possible heuristics, what are their advantages/drawbacks?
- What is the idea of ARA*?
- What is the idea and the advantage of adaptive node expansion
Inverse Kinematics and Whole-Body Motion Planning
Basic Jacobian IK Technique

while (e is too far from g) {
    Compute $J(e, q)$ for the current config. $q$
    Compute $J^{-1}$
    $\Delta e = \alpha (g - e)$  // choose a step to take
    $\Delta q = J^{-1} \Delta e$  // compute change in joints
    $q = q + \Delta q$  // apply change to joints
    Compute resulting $e$
    // apply FK compute new pose of end-effector
}
Questions

- How does the basic Jacobian IK technique work? Write down the algorithm.
- How is the Jacobian defined?
- What are the limitations of Jacobian control techniques?
The algorithm terminates by checking if \( q_{\text{new}} \) is near the goal.
RRTs – Properties (1)

- Easy to implement
- Fast
- Produce non-optimal paths: solutions are typically jagged and may be overly long
- Post-processing such as smoothing is necessary
- Generated paths are not repeatable and unpredictable
- Rely on a distance metric (e.g., Euclidean)
RRTs – Properties (2)

- The probability of finding a solution if one exists approaches 1 (probabilistic completeness)
- Unknown rate of convergence
- When there is no solution (path is blocked due to obstacles or other constraints), the planner may run forever
- To avoid endless runtime, the search is stopped after a certain number of iterations
- All in all: good balance between greedy search and exploration
Questions

- How does the tree extension step in RRTs work?
- What is the idea of RRT-connect and how does the algorithm work?
- What are the properties of RRTs?
Inverse Reachability Map (IRM)

- The IRM represents the set of **potential stance poses** relative to the end-effector pose
- Allows for selecting an optimal stance pose for a given grasping target
- Computed once offline
- Queried online

Cross section through the IRM showing potential feet locations

- red=low
- green=high
Determining the Optimal Stance Pose Given a Grasp Pose

- Given a desired 6D end-effector pose with the transform $F_{\text{grasp}}$
- How to determine the optimal stance pose?
Determining the Optimal Stance Pose Given a Grasp Pose

- Transform the IRM and determine valid configurations of the feet on the ground.
- Transform the IRM voxel centroids according to $F_{grasp}$ to get $tIRM$.
- Intersect $tIRM$ with the floor plane $F$:
  \[ IRM_{floor} = tIRM \cap F \]
- Remove unfeasible configurations from $IRM_{floor}$ to get $IRM_{stance}$.
Questions

- How is the reachability map constructed?
- How can an IRM be obtained?
- How can we select an optimal stance pose from the IRM given an end-effector pose?
Bipedal Walking
Bipedal Walking

- Abstract kinematics
- Motion pattern generation
- Inverted pendulum equation of motion and linearization
- ZMP and CoM concepts
- ZMP preview control (basic concept)
- Capture step control (basic concept)
- Cart-pole model equation of motion
Questions

- Why is abstract kinematics simpler as IK?
- How does a motion pattern generator use abstract kinematics?
- Explain the inverse pendulum model and its linearization
- What is the relation between the IPM and the ZMP?
- How does ZMP preview control roughly work?
- What is the main idea of the capture step framework?
Path Planning with Obstacle Class Information
Key Idea to Speed-Up Planning

- Simplify planning by splitting the whole plan into several parts
- Exploit knowledge about different obstacle classes to choose appropriate robot actions
Approach

- Combine 2D path planning, 3D foot step planning, and object manipulation
- Segment objects using a CNN
- For each object class, estimate costs of pre-defined actions to overcome the obstacle
- Encode the action costs of detected objects in a 2D grid
- Use A* on this map for fast path planning
- Path implicitly contains the necessary actions to handle obstacles
Questions

- What are the main ideas of using obstacle class information?
- Why is this an efficient approach for planning?
Bag-of-Words Model & Appearance-Based Mapping
Bag of Visual Words

- Visual words = independent features
- Construct a dictionary of representative words (codewords)
- Represent the images using **histograms of word occurrences** (bag)

Each detected feature is assigned to the closest entry in the codebook

source: L. Fei-Fei
Learning the Dictionary

cluster center = code words

clustering (e.g., k-means)

source: L. Fei-Fei
Appearance-Based Navigation

current observation:

map:
FABMAP: Graphical Model

appearance model

detector model

Chow Liu tree

word existence

word observation
Questions

- What is the general idea of the bag-of-words approach?
- How is the dictionary be learned?
- What are the strengths of this method?
- How can it be used for appearance-based mapping?
Summary

- I hope you enjoyed the lecture
- Many thanks for the positive evaluation and helpful comments!
- Good luck with the exam and happy holidays!