Lab class: Autonomous robotics
Exercise sheets

Jean-Stephane Jokeit, Oliver Lomp,
Mathis Richter, Stephan Zibner, Prof. Dr. Gregor Schöner

September 4, 2015

Student information

Name

ID number (German: Matrikelnummer)

Grade overview

Problem	Tutor signature	Points
1		/ 10
2		/ 20
3		/ 20
4		/ 20
5		/ 20
6		/ 20
	Total points	/ 110
	Grade (%)	

1 Controlling the e-puck

Problem: Let the robot drive from the starting position to the target on a printout of the first environment (Figure 1). The trajectory of the robot does not have to be smooth. The robot has to remain within the dashed area and may not touch any obstacles.
(Hint: Set wheel speeds, pause, repeat.)

Educational objectives of the problem:

- Understanding basics of Matlab
- Getting to know basic robot control

Theoretical questions:

- Your robot finds its way to the target with the help of your program. Would you call the robot autonomous? If yes, explain what makes it autonomous. If no, explain why not and what is missing.
- What would you have to change in your program if we chose a different starting or end point or moved the obstacles?
- Which information does the robot need so that it could cope with such a variable environment? Where could the robot get this information? Could it acquire it on its own?

Educational objectives of the theoretical questions:

- Establish an understanding of what autonomy means
- Understanding how much knowledge about the world is inherent in the program
- Understanding how much the program generalizes
- Thinking about what kind of information an autonomous agent needs to navigate

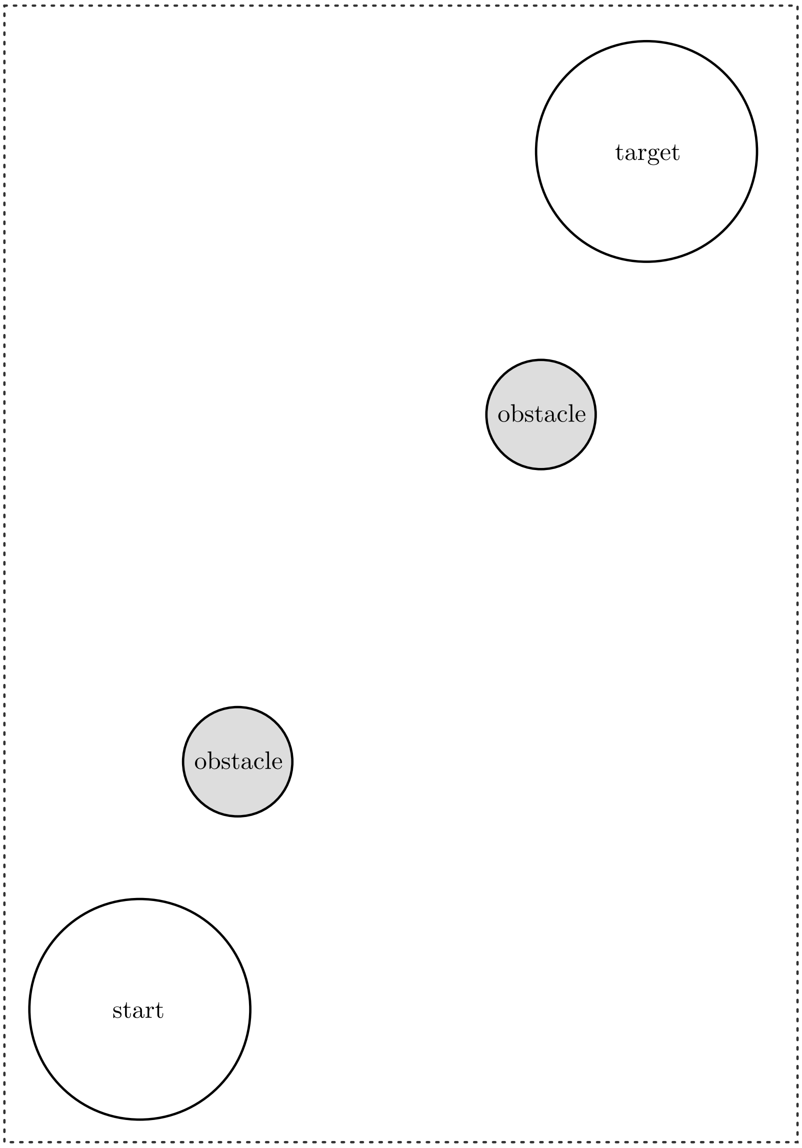

Figure 1: The environment for the first problem.

2 Kinematics

Problem: We now have an environment without obstacles, where the starting and end position can be varied (Figure 2). Write a program that brings the robot from an arbitrary starting position to an arbitrary end position. The program should get the coordinates (e.g., in millimeters) of the starting position and end position as well as the initial orientation (e.g., in degrees) of the robot as parameters. The coordinates should be expressed relative to the global (allocentric) coordinate frame as defined in the printout. The final orientation of the robot does not matter. We have marked some exemplary positions $P1, \ldots, P4$ in the environment that you may use, but other coordinates also have to work as starting and end positions. Try all combinations of starting positions and end positions with a number of initial orientations to make sure there are no errors in your code.[1]
(Hint: Turn first, drive later.)

Educational objectives of the problem:

- Understanding the trigonometry for determining the direction of the target
- Understanding how to rotate the robot by a given angle
- Understanding how to make the robot drive a given distance

Theoretical questions:

- Explain how you can give the robot coordinates in a metric unit (e.g., millimeters)—how does the robot know where and how far to drive?
- Explain the technical term 'kinematics' and its connection to the problem. Did you have to program a 'forward kinematics' or 'backward kinematics'? Explain.
- Could the program replan if we moved the target along the way? If yes, explain how this works. If no, explain why this does not work.
- Let us assume that we now introduced fixed obstacles into the environment.

[1] You will continue to use this code on subsequent problems so make sure that it is free of errors.

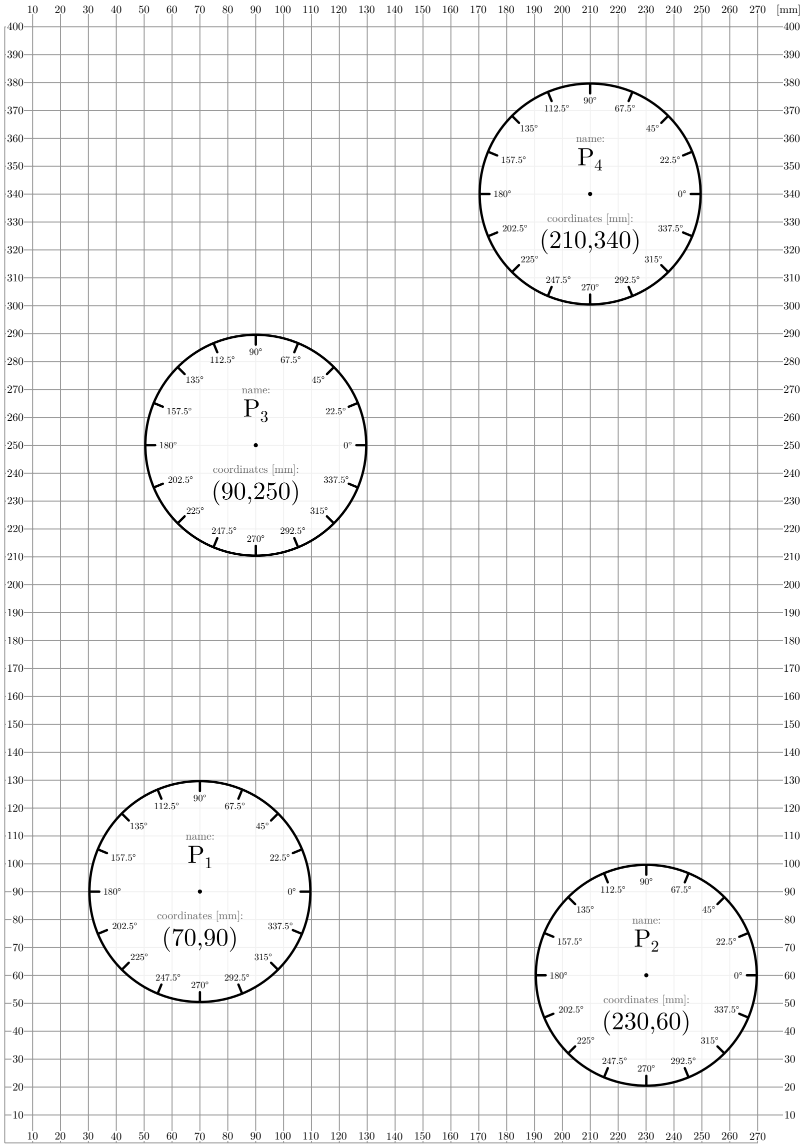

Figure 2: The environment for the second problem.

- Could this problem be solved without using any of the robot's sensors? What kind of information would the robot need for this?
- What would a solution look like if you could use its sensors? What kind of sensors would you use?

Educational objectives of the theoretical questions:

- Understanding how much knowledge the robot has about its surroundings and about itself.
- Understanding the term 'kinematics' and its relevance to this problem.
- Understanding what information the robot needs in order to interact with the world.
- Thinking about the next problem.

3 Sensors

Problem: We will now make the environment more difficult by placing an obstacle between the starting position and the target (see Figure 3). Write a program that makes the robot drive from a (variable) starting position (e.g., one of $A_1, \ldots, A_3$) and a (variable) initial orientation toward a (variable) end position (e.g., one of $B_1, \ldots, B_3$). The robot may not touch the obstacle. Do not hard-code the obstacle's position into your program. Instead, use the infrared sensors to detect when the robot is close to an obstacle and then avoid it by changing course. It is fine if the robot drives off the paper. *After avoiding the obstacle, the robot does not have to resume its approach of the target and it thus does not have to reach the target, but it should stop driving at some point.* Make the obstacle avoidance dependent on the direction of the obstacle, that is, if the obstacle is right in the middle of the robot's path, avoid it more strongly than if it is off to the side of the path.
Once the obstacle avoidance works, try it with obstacles made from different materials.

Educational objectives of the problem:

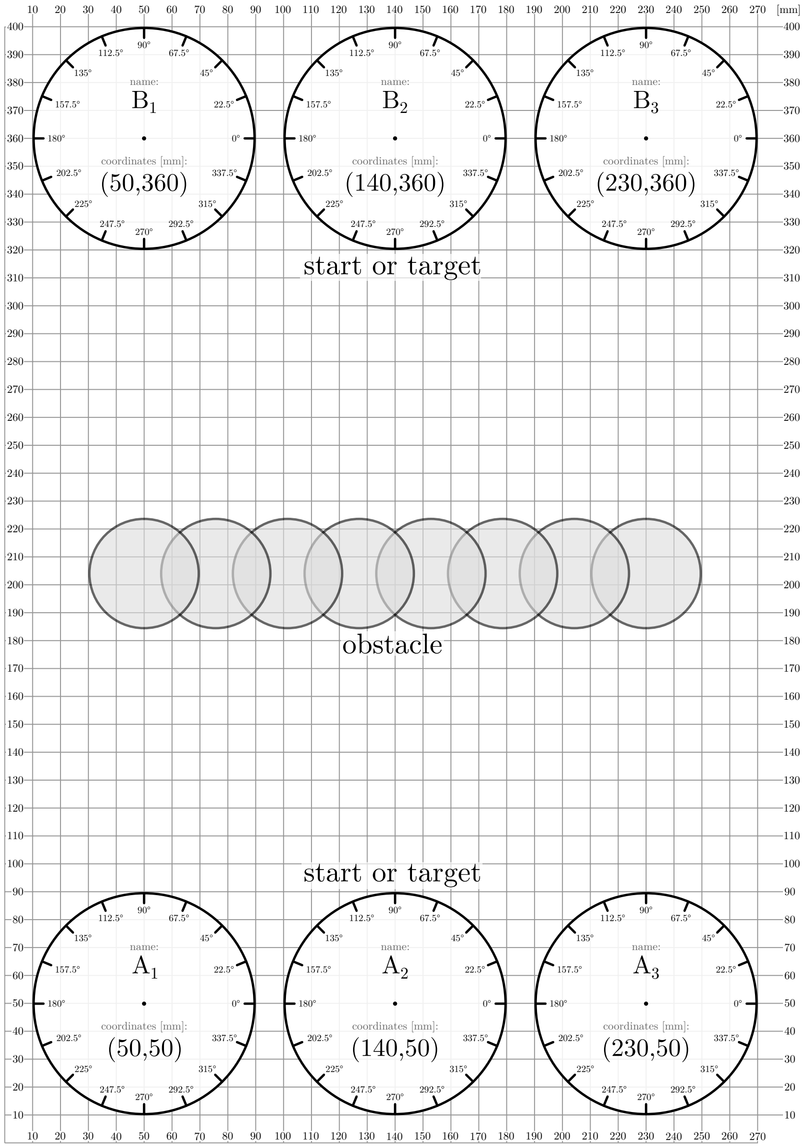

Figure 3: The environment for the rest of the lab class.

- Understanding how to read out and integrate the infrared sensors of the e-puck robot
- Calibrating the infrared sensors
- Seeing a simple obstacle avoidance system in action
- Understanding how the material of the obstacle influences the infrared sensors

Theoretical questions:

- Explain how you use the infrared sensors to detect obstacles.
- Explain in detail how you have programmed the obstacle avoidance. How and when does the robot avoid an obstance? In which direction does it drive to avoid it? How long does it keep avoiding the obstacle?
- How does your obstacle avoidance depend on the position of the obstacle in the path of the robot?
- How does your program deal with the change in course due to the obstacle avoidance? If the robot were to resume the target approach, where would it drive?
- What information is needed to bring the robot to the target after it has avoided an obstacle?
- What kind of influence does the material of the obstacle have on the robot's behavior? Describe and explain the effect you observed.

Educational objectives of the theoretical questions:

- Understanding the sensors and their integration
- Understanding the approach for obstacle avoidance
- Understanding how much knowledge the robot has about its surroundings and about itself

4 Odometry

Problem: Extend your program from the last problem so that the robot reaches the target after avoiding the obstacle. Once this works, extend the program further so that the robot drives back and forth indefinitely between the starting position and the ending position.
(Hint: The current position and orientation of the robot can be determined by integrating sensor readings, i.e., encoder values, over time.)

Educational objectives of the problem:

- Understanding the equations and trigonometry involved in odometry

Theoretical questions:

- Explain the technical term 'odometry' and its connection to the problem.
- Could you calculate the current position from the generated motor commands instead of the encoder values? Does it have advantages over calcuting the position from the encoders? If so, name them. Does it have disadvantages? If so, what are they?
- Are there other ways of estimating the current position? Name at least one and sketch how it would work. You may assume that you could equip the robot and the environment with sensors.
- Write about what you notice when the robot drives back and forth between the two positions. For how long does it work? If there are any problems, explain how they might come about. Can you think of a way of resolving these issues?

Educational objectives of the theoretical questions:

- Understanding odometry and its limitations
- Thinking about other methods of estimating the current position

5 Attractor dynamics: target approach

We will now solve the last problem again, but this time, using an attractor dynamics approach. In this problem, you will develop a dynamical system that is influenced by two components, one that orients the robot to the target, and one that lets it avoid obstacles along the way. We split this problem into two parts: first, we implement the target approach, then we add the obstacle avoidance to this approach.
Problem: Have the robot drive from the starting position to the target using attractor dynamics. Use the previous environment (Figure 3), without obstacles for now. Assume that the coordinates of the robot's starting position, its initial orientation and the coordinates of the target are given as parameters to your program.
First, make the robot turn toward the target on the spot, that is, without moving forwards. Use a linear dynamical system (see background material) defined over the orientation of the robot. *Hint: This involves numerically solving dynamics. The structure of the program for that will be different from what you have programmed so far.*
Once turning on the spot works, add a constant forward speed to drive the robot to the target while turning. Use the odometry functions from the previous experiment and stop the robot at an appropriate point.
Finally, replace the linear dynamical system with a sine-based one (see background material).

Educational objectives of the problem:

- Understanding how attractor dynamics can orient the robot toward the target
- Understanding a numerical method for solving dynamics
- Investigating the properties of dynamics as a mechanism for controlling a robot

Theoretical questions:

- Explain the dynamical system and why it makes the robot turn toward the target. Over which variable is it defined?
- Create at least two different figures (i.e., a phase plot and a plot that shows how the system develops over time) and refer to your figures

while explaining. What does each figure mean with respect to the robot?

- Explain attractors and repellors using plots.
- Create another phase plot of a dynamical system that shows both an attractor and a repellor and mark them accordingly.
- In which cases does the robot fail to reach its target? Explain how this depends on different dynamical systems and the chosen parameters.
- What does numerically solving the dynamics mean programmatically? Did you make any assumptions, simplifications? How did you treat time?
- Could you have easily modified your program from the previous problem (problem 4) to drive forward while turning? If so, how? If not, why? Are there differences between the solution that uses dynamics and previous solutions? If so, what are they?

Educational objectives of the theoretical questions:

- Understanding the difference between the time course of a dynamics and the dynamics itself, and how these map to the robot's behavior
- Understanding the importance of parameterization
- Developing an intuition for the details of numerical methods for solving dynamics
- Thinking through what makes attractors attractive

6 Attractor dynamics: obstacle avoidance

Problem: Extend your program so that the robot can avoid obstacles while it is driving toward the target. The robot should still move forward and turn at the same time. Additionally, it should be repelled from obstacles and avoid them in smooth trajectories. Solve the obstacle avoidance by modifying the dynamical system you have implemented in problem 5.
Hint: Use the force-lets described in the background material for obstacle avoidance. You will need to choose values for various parameters; make sure you understand the equations involved here first.

Educational objectives of the problem:

- Understanding the equation for obstacle force-lets and its parameters.
- Understanding the properties of a combination of different influences on the heading direction.

Theoretical questions:

- In the environment the robot is navigating, which elements represent attractors and which represent repellors? Why? Explain.
- Explain the equations you use to generate the influence of the obstacles. Explain how each parameter of the equation influences the shape of the function and how this impacts the robot's behavior. Make plots where appropriate.
- Discuss the robot's perception of obstacles: Does it perceive them as a discrete set of obstacles? (How) does this correspond with the explanation of the attractor dynamics in the background material?
- Explain the bifurcation that the dynamics undergoes between Figures 12 and 13 in the background material. Why is there a repellor for each obstacle in Figure 12, while there is only a single repellor in Figure 13? What does this mean for the robot's behavior? Make drawings and explain.

Educational objectives of the theoretical questions:

- Understanding the different components of the obstacle part of the dynamics.
- Observing emergent behavior that was not programmed in explicitly.
- Understanding the relation between bifurcations and behavior (decisions), and how these emerge from the chosen dynamical system