Home assignment #1

Quick start with V-REP and exercise:

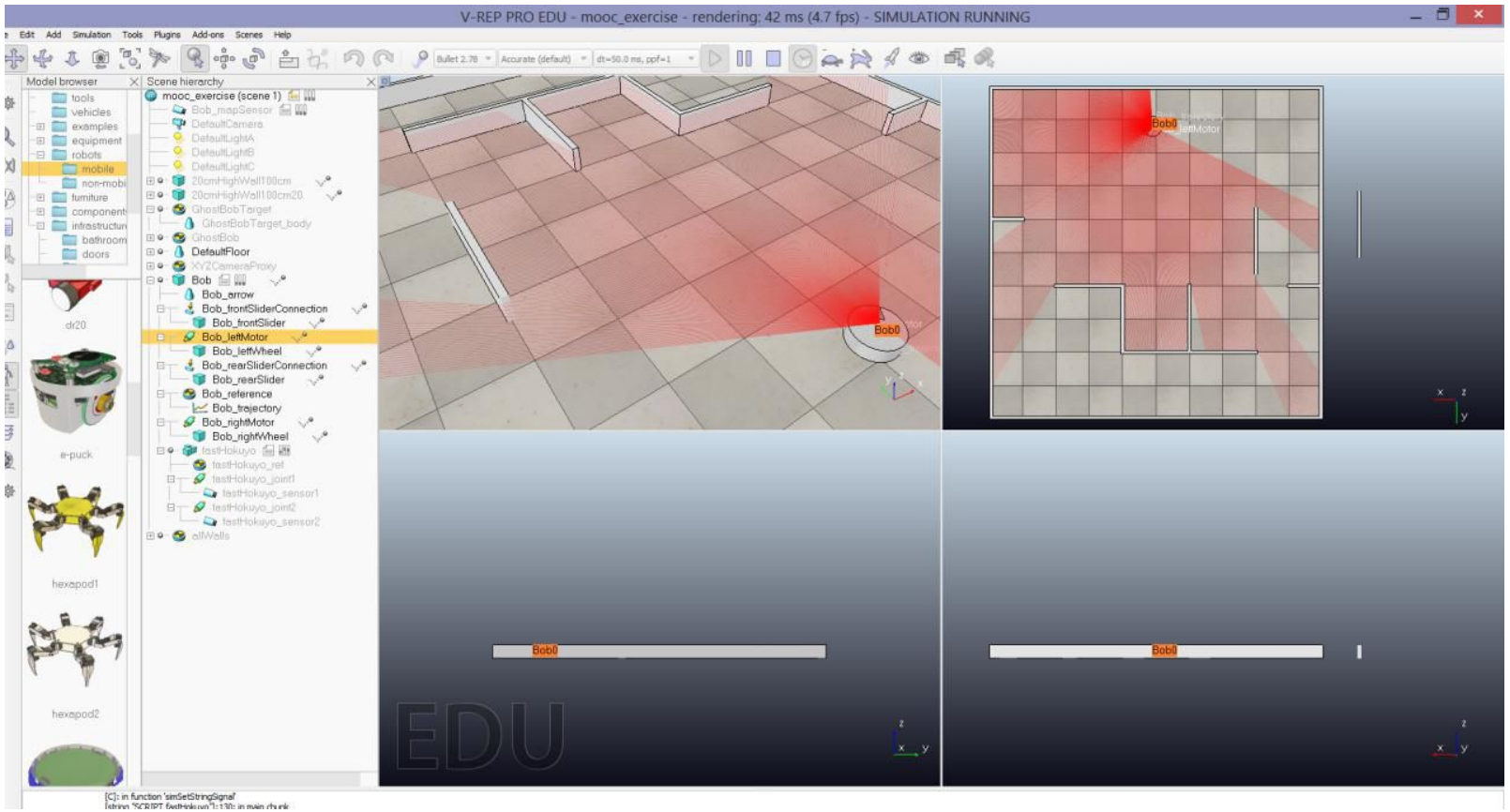
- Download the exercise task from the web-page of Autonomous Systems Lab (Institute of Robotics and Intelligent Systems in ETH Zurich): https://www.ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/asl-dam/documents/lectures/autonomous_mobile_robots/spring-2016/ethzasl_amr_exercise2.zip
- Open “exercise 2-assignment.pdf” and use this tutorial step-by-step
- To execute hw1 use V-REP PRO EDU and MATLAB/Octave

Reference:

- Margarita Chli, Paul Furgale, Marco Hutter, Martin Rufli, Davide Scaramuzza, Roland Siegwart, Autonomous Mobile Robots, ETH, 2016, http://www.asl.ethz.ch/education/lectures/autonomous_mobile_robots.html

Running a Scene with robot Bob

- Open V-REP scene: scene/mooc_exercises.ttt
- Set additional walls: Models/infrastructure/walls
- Run a mobile robot Bob (motion + lidar scanning)

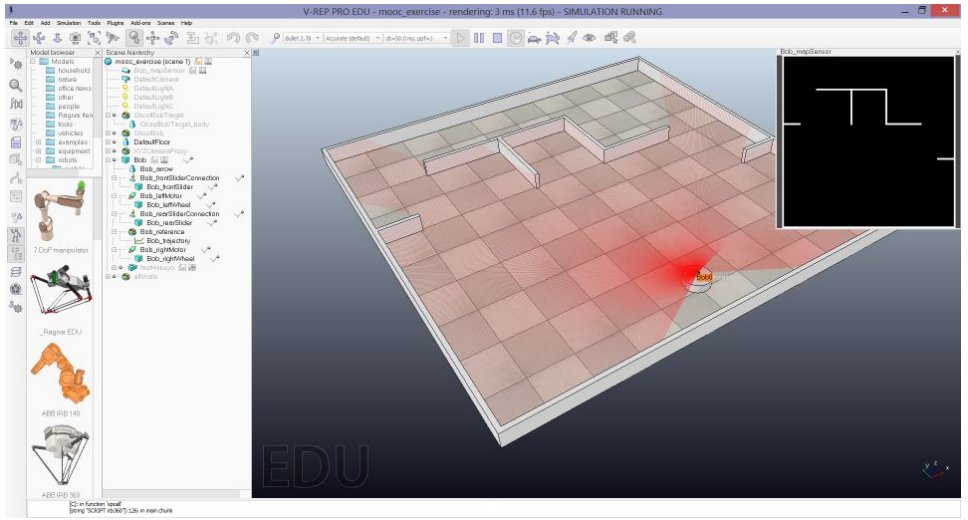


Start a remote API client from MATLAB

- Open MATLAB, copy relevant *.dll and files.
- Set up MATLAB - V-REP communication by the MATLAB commands

```
>> simulation_setup
Running Matlab win64
make sure you use the corresponding remoteApi library
(i.e. 32bit Matlab will not work with 64bit remoteApi, and vice-versa)

ans =

    vrep: [1x1 remApi]

>> con = simulation_setup
Running Matlab win64
make sure you use the corresponding remoteApi library
(i.e. 32bit Matlab will not work with 64bit remoteApi, and vice-versa)

con =

    vrep: [1x1 remApi]

>> simulation_openConnection(con,0)

>> simulation_start(ans)
```

Result



Feed Forward Control

- For a differential-drive robot, the kinematic model is described by the equations:

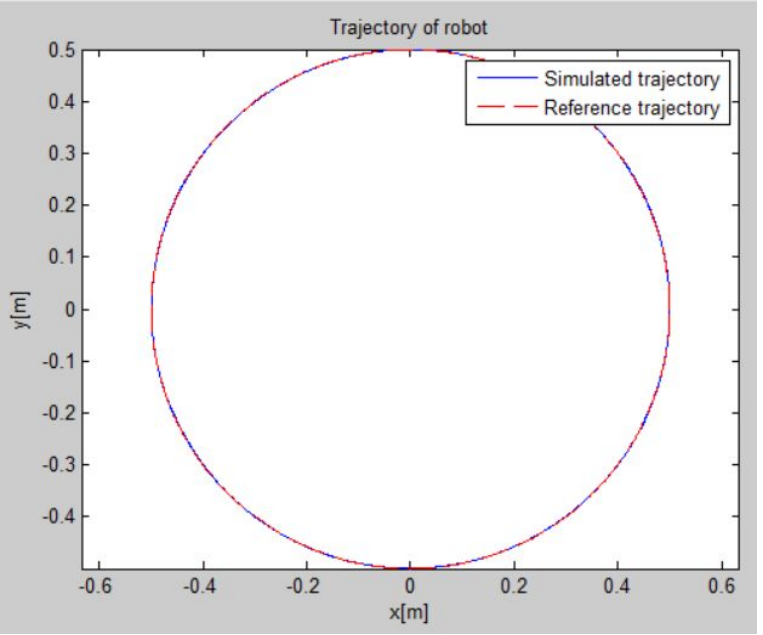
$$v = \frac{r\dot{\phi}_r}{2} + \frac{r\dot{\phi}_l}{2}$$
$$\omega = \frac{r\dot{\phi}_r}{2l} - \frac{r\dot{\phi}_l}{2l}$$

where (υ, ω) represent forward and rotational velocity of the robot platform, respectively, and $(\dot{\phi}_r, \dot{\phi}_l)$ the spinning speed of the right and left wheels. The wheel radius is given by *r* and *l* denotes half of the inter-wheel distance.

Task 1:

- Edit **calculateWheelSpeeds.m** in such a manner that it computes the spinning speeds $(\dot{\phi}_r, \dot{\phi}_l)$ based on the given velocities (υ, ω). Based on this feed forward controller, the robot will attempt to drive on a 0.5m radius circular trajectory.
- Validation: The feed forward controller can be evaluated by running the **test/testCircleDrive.m** script.

Result



Closed-loop Control

Task 2:

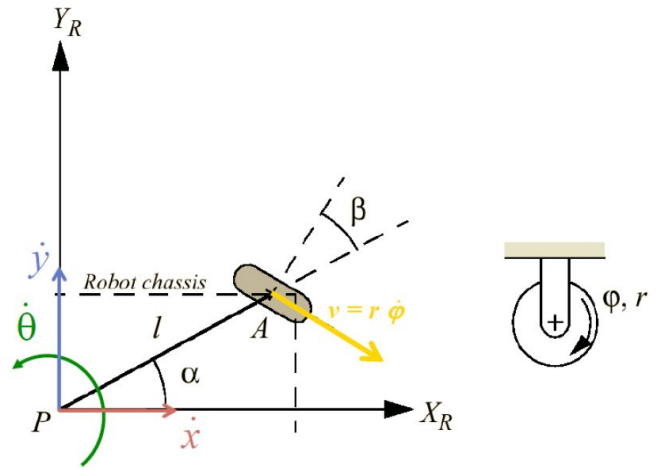
- Use control law

$$v = k_\rho \rho \qquad k_\rho > 0,$$

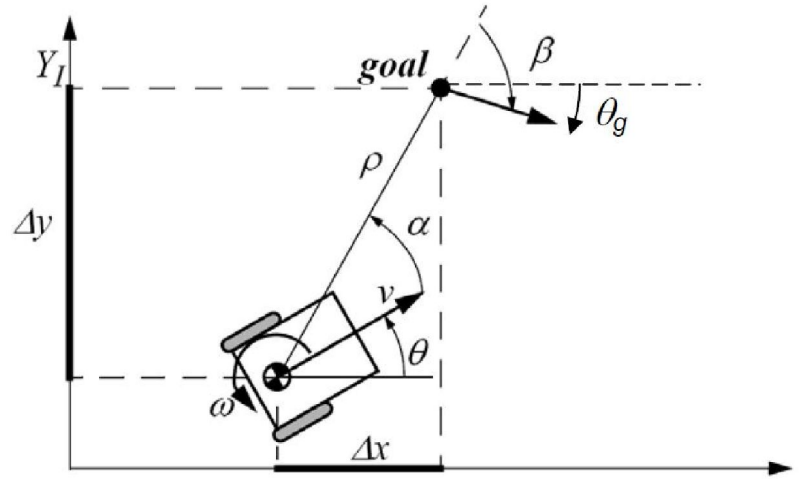
$$\omega = k_\alpha \alpha + k_\beta \beta \qquad k_\beta < 0,$$

$$0 < k_\alpha + \frac{5}{3} k_\beta - \frac{2}{\pi} k_c \rho$$

- Task: Please implement this close-loop position controller within **calculateControlOutput.m**. Helpful MATLAB commands/files are: atan2, and **normalizeAngle.m**. All angles are given in the positive right-hand coordinate frame (counter clockwise).
- Validation: Start V-REP, load scene **scene/mooc_exercises.ttt** and start the simulation. You should see a circular robot and a set of walls. Now run the script **vrep/vrepSimulation.m** within MATLAB.



Variables for kinematic model of the differential-drive robot

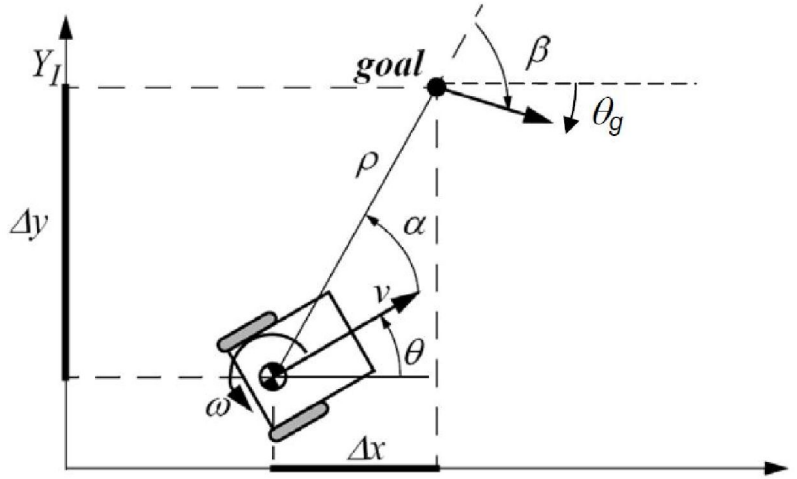


Variables of State feedback control for mobile robot Bob

Closed-loop Control Enhanced

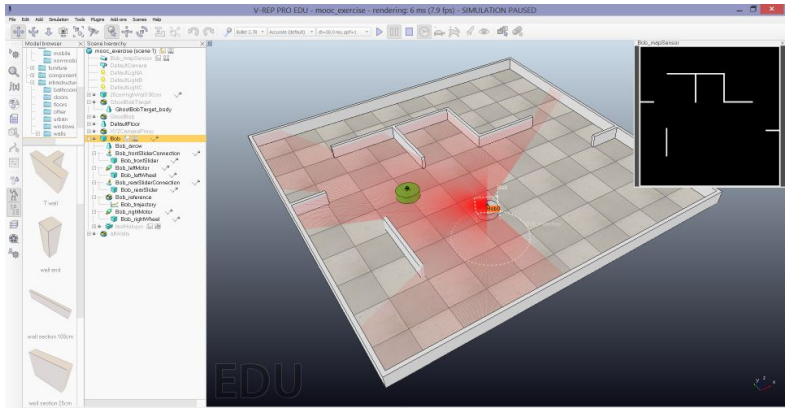
Task 3:

- Improve your code within **calculateControlOutput.m** such way that the robot is able to drive in both directions depending on where it is located with respect to the target pose, driving towards with a constant speed.
- Validation: Start V-REP, load scene **scene/mooc_exercises.ttt** and start the simulation.
- You should see a circular robot and a set of walls. Now run the script **vrep/vrepSimulation.m** within MATLAB. The robot should starts driving towards the green robot ghost and should come to a stop in the same spot and the same orientation as the target.



Variables of State feedback control for mobile robot Bob

Result



Home assignment #1

Task 4.
Complete either of the following two options:

- Make Bob to move to the target avoiding walls.

You may use the generated room map.

- Build a room map by using the lidar sensor (fastHokuyo) of Bob.

Bob will need to drive through the room in order to do this.