Autonomous control of field robot

Individually study activity in field robotics

Authors:
Christian Liin Hansen

Instructors:
Kjeld Jensen

University:
The Technical Faculty, University of Southern Denmark

Course:
Extension of X-ROBO-U1 by individually study activity. 5 ECTS point

Project period:
September 1^{st} - January 1^{st} 2014

Contents

1 Introduction

- General introduction for the course and the different tasks of the project

1.1 Project description

The University of Southern Denmark (SDU) was providing a summer course in Field Robotic in summer of 2014. This course was inspired from the annual Field Robotic Event (FRE) [1] and hence a field with growing maize was established outside the university. The goal was to design an autonomous field robot capable of making its way in the field of the university from a START position to a FINISH position. The path was similar to the field sketched in figure 1.1. The outcome of the summer course was a mechanical robot, which is shown in on the front page of this rapport. The field robot contained motors, H-bridges, FroboMind controller board and camera. The camera was mounted in the front, adjusted to make visual contact to the rows of maize in the field. An fully image processing algorithm with row extraction was implemented by the author during the summer course. The result of this is shown in figure 1.2 and figure 1.3. Doing test inside RoboLab the algorithm was performing better, since the environment was ideal compared to the field. The result in figure 1.2 was only produced by controlling the robot using a keyboard. The result in figure 1.3 was only produced by walking with a handheld webcam in the field outside.

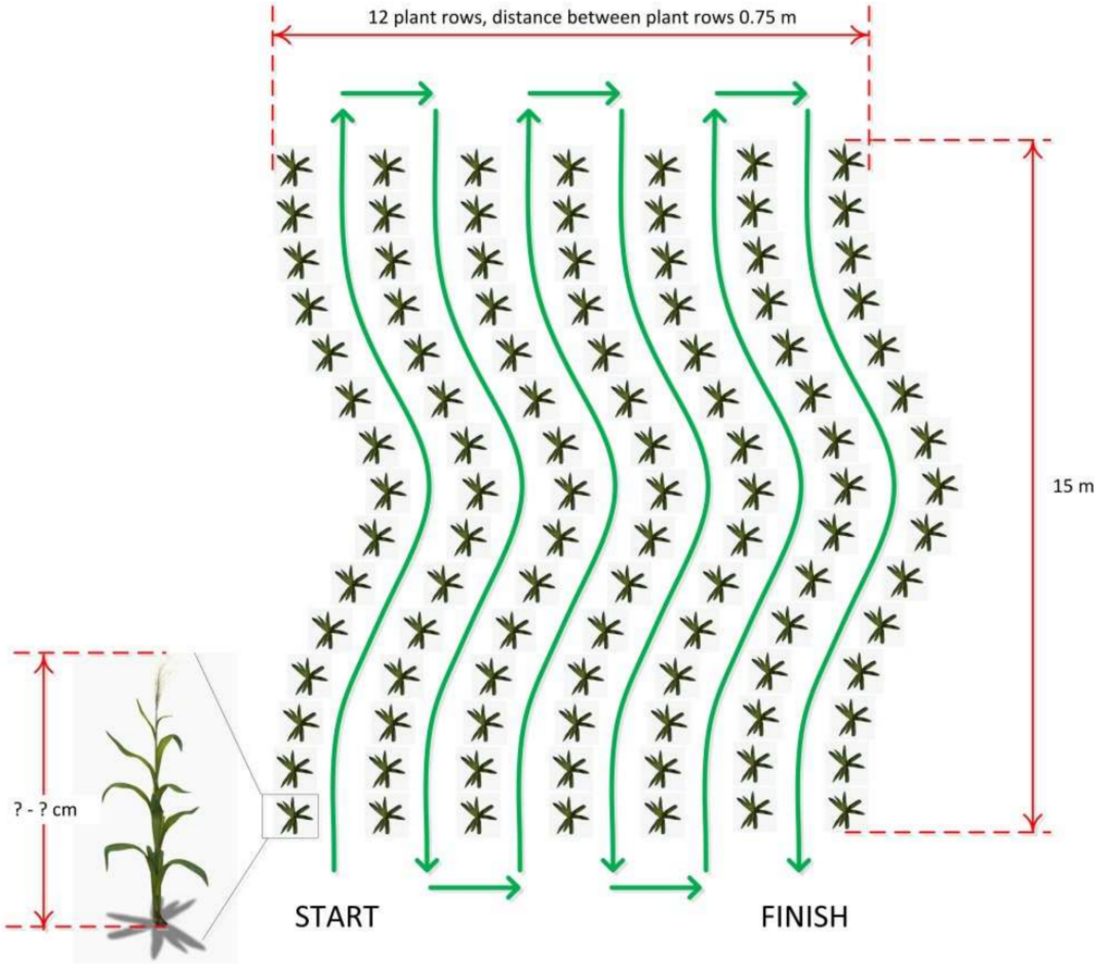

Figure 1.1: A sketch of the test field at FRE 2014

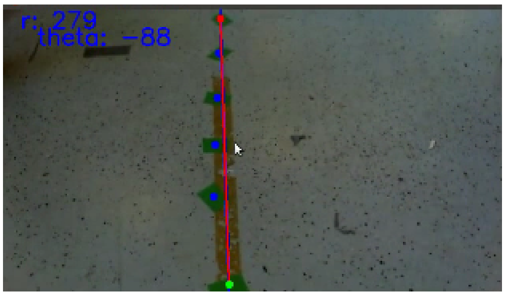

Figure 1.2: Testing algorithm inside RoboLab

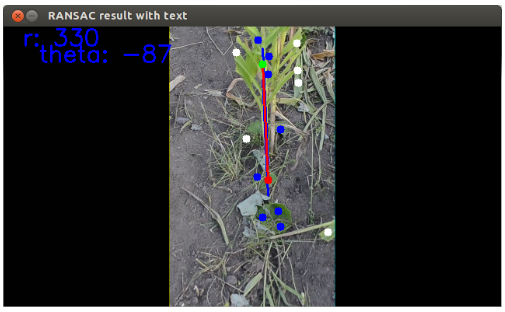

Figure 1.3: Testing algorithm outside in field

1.2 Problem statement

At the end of the summer course, the mechanically robot was constructed and software for the camera was implemented to do row extraction, but nothing more was completed regarding implementation of the field robot in ROS, doing the summer course. That means that the field robot never came to the point, where it could drive autonomously through the field or even inside in RoboLab. This is the fundamental problem statement, which justify the reason for the authors individual study activity. The goal of this project is to continue the work after the summer course and be able to implement an system in order to experience a fully autonomously field robot.
The problem in this project is to build up a FroboMind architecture, bases on ROS, that make the field robot able to drive autonomously by using at least one webcam for navigation and IMU for turning. To start with the indoor environment, the robot must be able to follow the line of green tape on the floor, by having the two driving wheels on each side of the line. At the end of a line, the robot must be able to turn and follow a line back. Further extension is to have the field robot be able to follow the line of maize out in the field. This would include more optimizing work on the image processing node, due to the fact that the environment is much more complex.

1.3 Related work - row extraction

From previous work [5] in the summer course 2014, ideas for implement the mission control was discussed. The image processing node output the distance r and the angle θ. Additionally an *end of line* flag parameter is added doing this project. The image processing node has the following interface:

- Input: 640x480 RGB image from webcam at 30 fps
- Output: The distance r, which is the perpendicular distance from the line extracted to the upper left corner of the images. r is between 0 and 639 pixels
- Output: The angle θ, which is the angle between the horizontal x-axis and the line extraction. θ is between 0 and 180 degrees.
- Output: The boolean flag *end of line* is true when the robot has approached an end of a line of maize. Otherwise false. angle between the horizontal x-axis and the line extraction.

The image processing algorithm contains three components. The first component is extracting the green nuance for each image and together with a threshold, contours finding and raw moments calculation, the central mass coordinates for each green segment in each images is found. The second component is applying the RANSAC [4] algorithm on the dataset that contains the (x,y) central mass coordinates. The RANSAC stands for Random Sample Consensus, which means a line is spanned by two random points and the number of inliers is counted for that specific line. An inlier is defined as a point, where the perpendicular distance from the point to the line spanned by the two random is below a given distance threshold. If a given points exceed that given threshold it is an outlier. The RANSAC algorithm repeats its procedure 1000 times for each image and the line that contains the most inliers is defined to be the best fitted line in the current image. The third and last component in the image processing algorithm is to calculate the perpendicular distance r from the best fitted line and to the origo, which is defined to be the upper left corner of the image. Additional the angle of the best fitted line in respect to the horizontal x-axis in the images is calculated. The image processing node with the RANSAC algorithm was tested both inside RoboLab and outside in the field. The performance is illustrated in figure 1.2 and figure 1.3, where the distance r and angle theta is displayed.

1.4 Aim of project

The aim of the project aim is to implement a FroboMind [2] architecture, which is shown in figure 1.4. This figure is described below.

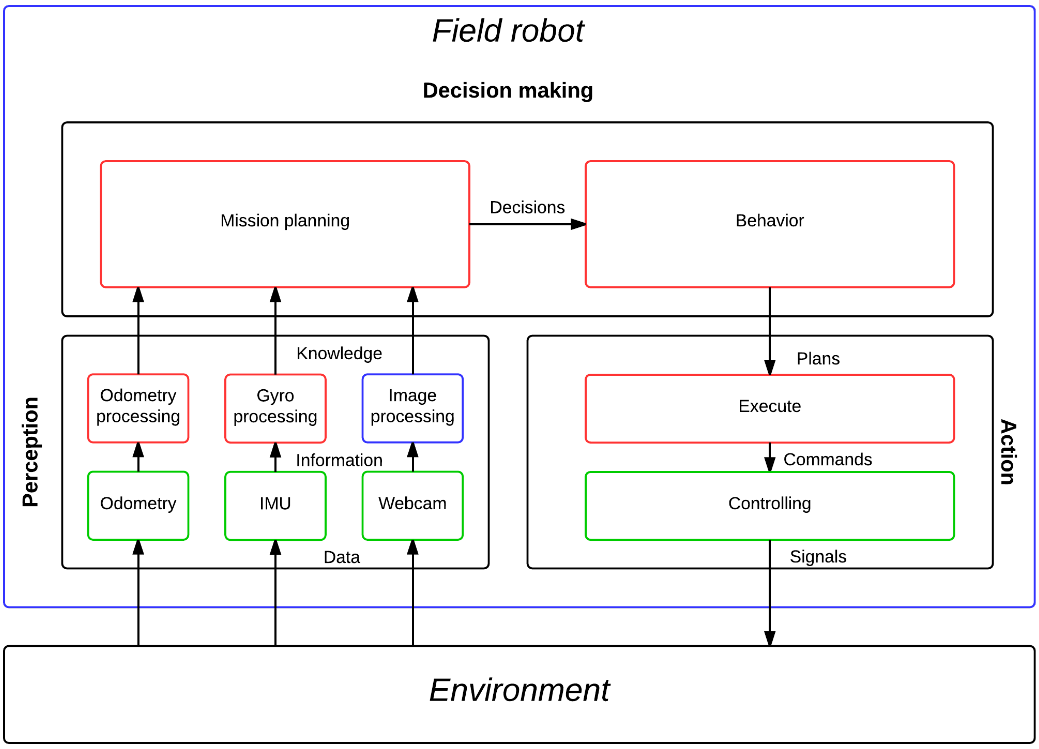

Figure 1.4: The FroboMind architecture of the field robot. Each component has been color coded. Green color indicates already implemented components, blue indicates components that needs optimization and red indicates components that is not implemented.

The outer blue *Field robot* frame, in figure 1.4 indicate that the physical robot might need optimization of the camera mounting location. The *Image processing* node needs to handle *End of line* detection and interface the parameters r and θ to the rest of the system through ROS. Optimization in general if the row detection should work outside in the field. An IMU will be connected to the system, and hence a *Gyro processing* node is needed to track the orientation. A *Decision making* node is needed to handle the inputs from the vision system and data from the IMU. This is referred to the *Mission Planning* component in figure 1.4. Inside the decision node, a state machine is implemented in order to let the robot have different *behaviours*. Regarding which behaviour the robot is in, different *executing* plans will be transferred to a FroboMind layer, which handles motor control. The *Odometry processing* component has not been implemented in this project.

1.4.1 Milestones in project

The project is decomposed into different milestones (MS). The aim is to complete all the MS. To follow a structured approach towards the aim, the MS has been prioritized. In this project the focus is to complete the chain in the project, i.e. have a field robot running autonomously inside RoboLab, where all components is working in the ideal environment. Taking the project further to handle outdoor environment is an optimization process, where each component can be optimized.
The milestones is defined as followed:

- MS 1: The robot is able to sense the green tape inside RoboLab by only using the webcam. The FroboMind architecture in figure 1.4 is implemented and hence the robot can follow a line of dotted green tape autonomously.
- MS 2: The robot is able to detect *End of line* zones inside RoboLab in the vision node, which can set the robot in different behaviours, i.g turning or line-following.
- MS 3: The gyro data from the IMU can be used to perform an approximately 180 degree turning around the center of mass of the robot inside RoboLab and following the same line again.
- MS 4: With a setup of two parallel line of green tape inside Robolab, the robot is able to turn around one of the driving wheel pivot point, when entering the end of line and hence follow the new line.
- MS 5: A implemented Kalman filter will provide the sensor fusion between data from vision node and gyro node and hence further stabilizing the line-following.
- MS 6: The vision algorithm is optimized to handle the outdoor complex environment regarding to *End of line* detection. Therefore the robot is able to sense and follow a line of maize outside in the field by using the webcam, gyro and sensor fusion with a Kalman filter.
- MS 7: The robot can successfully follow a line of maize in the field, turn at the end and follow the new line of maize.
- MS 8: The robot can successfully follow a line of maize in the field, perform an approximately 180 degree turning around the center of mass of the robot and following the same line of maize.
- MS 9: The robot can successfully follow a line of maize in the field,turn around one of the driving wheel pivot point, when entering the end of line and hence follow the new line of maize.
- MS 10: The robot can successfully complete multiple rows with successful line following and turning at end outside in the field.

2 Materials And Methods

In this section the different component in the project will be presented. For better overview and more efficient process, the project has been divided into several components in a *Work-Breakdown-Structure* diagram, shown in figure 2.1.

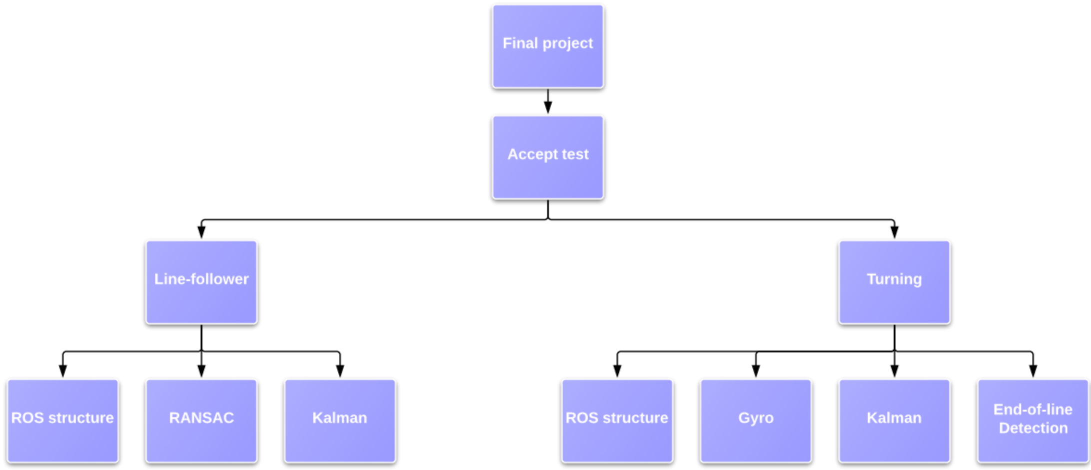

Figure 2.1: *Work-Breakdown-Structure* illustration shows how the project has been devided into several components

2.0.2 Test

The standard measurements between the rows in the field is 750 mm [3].The dimension of the field robot is 830 mm x 815 mm. The inner distance between the wheels is 600 mm. The robots translation and orientation offset is illustrated in figure 2.3 and figure 2.4 respectively. A combination between these scenarios will occur doing testing. As a starting point, the hypothesis will relate to test inside RoboLab. The green line indicate green tape on the floor. This scenario will be extended later in the project to the outside test field, where the green lines indicate rows of maize.
An overview of the different components, where all the phases has been completed is shown in figure 2.2. The PC with ROS contains the FroboMind architecture, which is illustrated in figure 1.4.
An solution for the problem is to implement the FroboMind [2] architecture, which is shown in figure 1.4. The Field robot frame, in figure 1.4 has been color coded blue, to indicate that the physical robot might need optimization of the camera mounting location. The image processing needs to handle “End-of-Line” detection and optimization in general if the row detection should work outside in the field.
The hypothesis is stated as followed: The robot will navigate through the line by having the driving wheels between the rows. The robot is able to drive sufficiently smooth by having a maximum offset in rotation and translation. These parameters is defined as followed:

- The field robot is able to follow the line with a maximum offset in translation of 100 mm. I.e. the distance from the inner line toward the driving wheels will be in range 200-400 mm.
- The field robot is able to follow the line with a maximum offset in orientation of 25 degrees. I.e. the angle between the direction vector and the horizontal x-axis is in range from -90 to -65 and from 90 to 65 degrees.

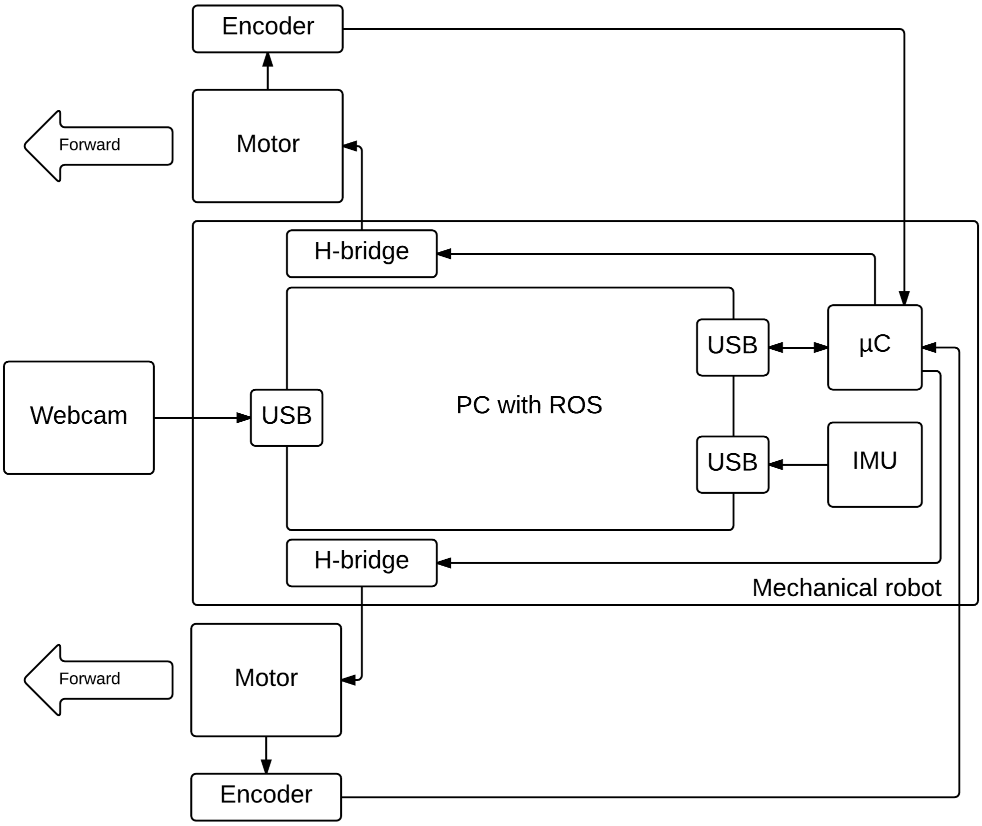

Figure 2.2: Implementation of all the physical components on the field robot.

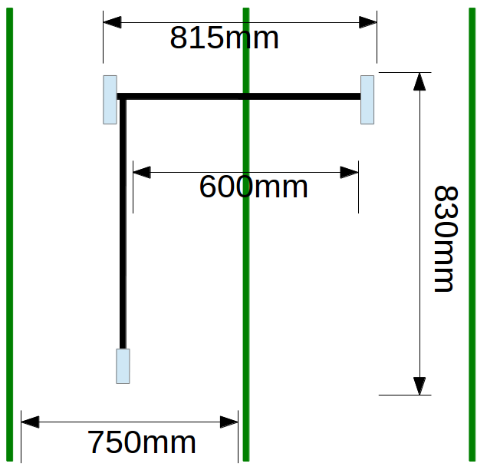

Figure 2.3: Translation offset equal to 0

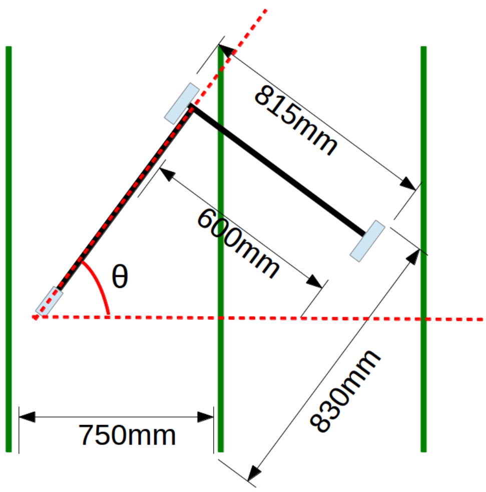

Figure 2.4: Orientation offset equal to θ

- The robot is able to turn 180 degrees with an maximum offset of 20 degrees. Each turn is within 1,5 meters, measured from the end of line to the center point of the robot. At the end of a turn the robot is able to follow the next row of maize.

2.1 Simulation line-follower

A line-follower has been implemented by extending previous work [5], which was the RANSAC algorithm, with a ROS structure. The Kalman filter has not yet been implemented in this component, either has the PID regulator for velocity commands. The result is a simulation of the following nodes:

- A vision node that publish the distance r and angle theta. The parameters is described is section **??**.

- A decision node that subscribe the parameters with a callback function and publish the decided angular and linear velocity commands.

Using *rqt_graph* from the rqt package, the nodes and topic is illustrated in figure 2.5

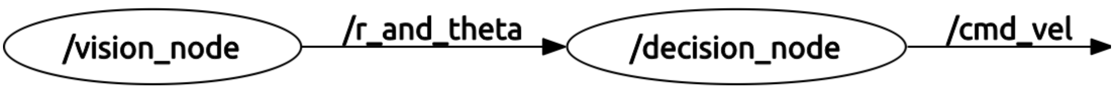

Figure 2.5: ROS structure with vision node that publish parameters on the *r_and_theta* topic. The decision node subscribes on the same topic and publish a linear and angular velocity on the *cmd_vel* topic. A dummy node listener was created to illustrate the last topic, which is not illustrated

To simulate the line-follower, the video input stream is based on a still image, figure 2.6, which can be rotated as illustrated in figure 2.7.

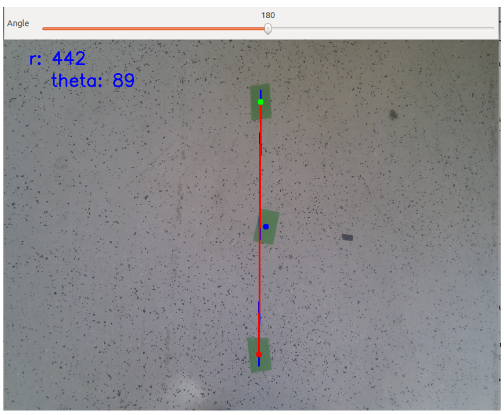

Figure 2.6: 89 degrees

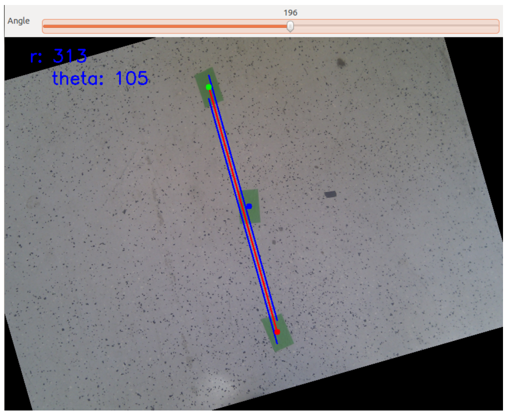

Figure 2.7: 105 degrees

The result of the simulation shows that is is possible to steer the robot. The simulation result is shown in figure 2.8

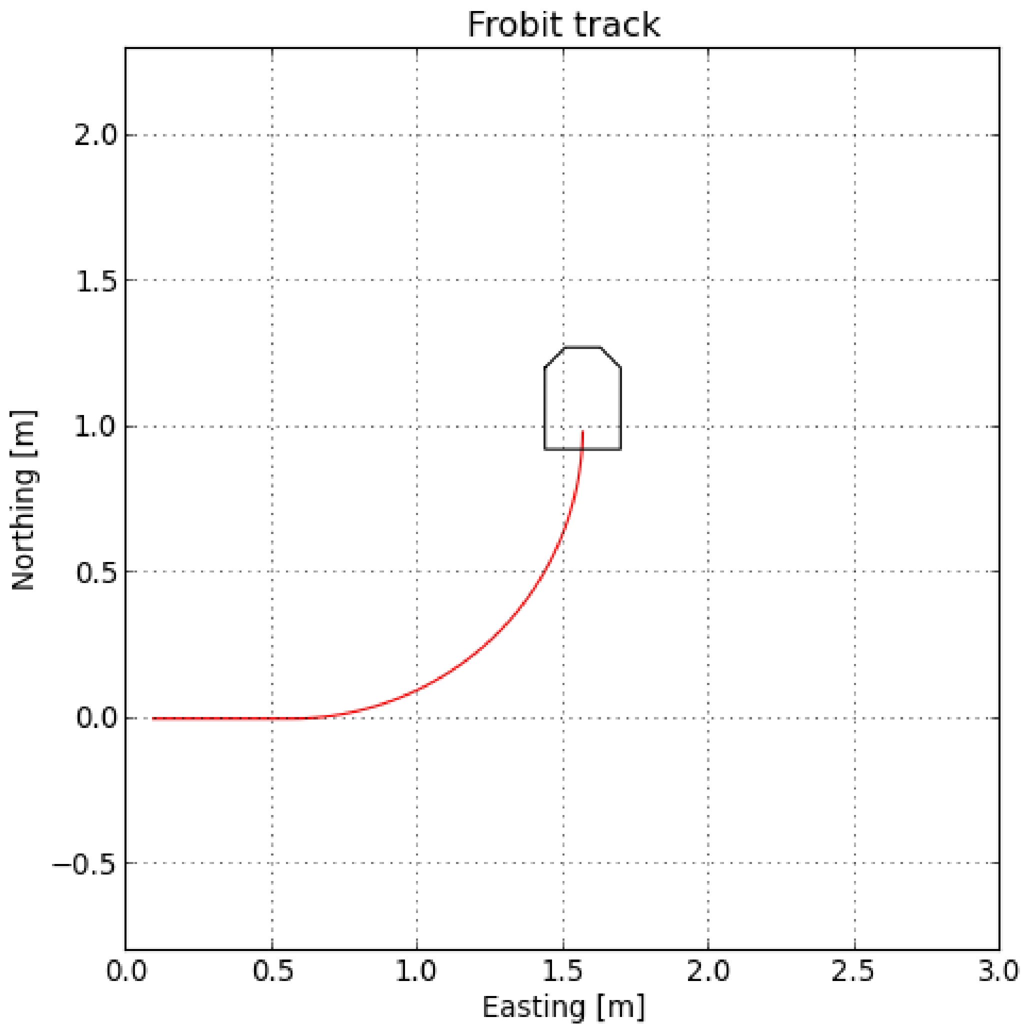

Figure 2.8: Simulating video stream with rotating still images result in change of angular velocity of robot in simulation. The linear velocity is constant.

2.2 Testing line-follower inside RoboLab

2.3 STuff that I did not reached before hand-in

Implementing: A PID regulator will be implemented in software to regulate the linear and angular velocity commands.

3 Deliverables

The project will be reviewed by the supervisor about halfway the trough the project period which is November 1st, 2014. By this date the following will be delivered:

- The sections “Introduction” and “Materials and methods” of the project report.

The project deadline is January 1st, 2015. By this date the following will be delivered:

- A project report documenting the fulfilment of:
  - Project aims
  - Deliverables
  - Learning goals
  - Accept test results
- FroboMind software modules and documentation
  - Mission control node implementation
  - Image processing node optimization

4 LearningGoals

At the end of this project, the student will be able to:

- Identify a specific problem domain through literature study.
- Apply engineering skills in the evaluation of existing commercial and research solutions related to the identified problem.
- Propose an optimal solution based on domain knowledge and literature review.
- Apply engineering approach to construct a prototype demonstrator based on the proposed solutions.

Furthermore the student has through the prototype demonstrated engineering skills within:

- Optimizing row detection algorithms for a web camera.
- Using FroboMind architecture to interface different sensors.
- Using a Kalman filter for sensor fusion between IMU, encoders and webcam.
- Using a PID regulator for regulate velocity commands.
- Project management.

5 Bibliography

[1] UNIVERSITÄT HOHENHEIM.
Field robot event 2014 - task description.
https://fre2014.uni-hohenheim.de/uploads/media/Booklet_content-v3-complete_01.pdf, 2014.
Available: 7-10-2014.

[2] Kjeld Jensen, Morten Larsen, Søren H. Nielsen, Leon B. Larsen, Kent S. Olsen, and Rasmus N. Jørgensen.
Towards an open software platform for field robots in precision agriculture.
Robotics, 3:207–234, 2014.
doi:10.3390/robotics3020207.

[3] Germany University of Hohenheim.
Field robot event 2014.
https://fre2014.uni-hohenheim.de/uploads/media/fre-2014-tasks-v06.pdf, 2014.
Available: 7-10-2014.

[4] Martin A. Fischler and Robert C. Bolles.
Random sample consensus - a paradigm for model fitting with apphcatlons to image analysis and automated cartography.
SRI International, 6:381–395, 1981.
doi:10.1145/358669.358692.

[5] Christian Liin Hansen.
Field robot summer course 2014.
https://www.dropbox.com/s/ew82lvrkpvb3pjh/FRSC14.pdf?dl=0, 2014.
Available: 7-10-2014.