0.1 læsevejledning

- Preface skal der blot ses bort fra
- Jeg har lavet en indholdsfortegnelse over hvad jeg forstiller mig raporten skal indeholde
- Kapitel 1, introduktion: Jeg har ikke skrevet det i nu, men forestiller mig at skrive noget med robotter i produktionen generelt og slutte den af med et initierende problem
- Kapitel 2: problem analysis, Det hele er klar til gennemlæsning og vil gerne have feedback omkring indhold, den røde tråd osv.
- kapitel 3: Udkast til problemformulering, hvad synes I? derudover tænker jeg der skal laves en afgrænsning

AALBORG UNIVERSITY
STUDENT REPORT

Visual Control of SCARA Adept Robot

Master Thesis

Author:
Thomas Pank Roulund

Supervisor:
Ole Madsen
Rasmus Skovgaard Andersen

AALBORG UNIVERSITY
STUDENT REPORT

http://www.m-tech.aau.dk/
Electro-Mechanical System Design
Fibigerstræde 16
Telefon 99 40 71 17
Fax 99 40 71 10
http://info@m-tech.aau.dk

Title:

Vision Control of SCARA Adept robot and planning of pick and place operation

Theme:

Master Thesis

Project period:
P10, EMSD4

Project group:
4.119f

Authors:

Thomas Pank Roulund

Supervisors:

Ole Madsen

Rasmus Skovgaard Andersen

Printed copies: x

Pages: x

x annex and x appendix

Completed: 18.12.2015

Synopsis:

Preface

This 4th semester master thesis has been written a student attending the Electro-Mechanical System Design engineering line at Aalborg University. The project period spans from the 2nd of February to the 2nd of June.

The references are created based on the Harvard method, meaning that each reference is written with both author and year of publish. All figures are named X.Y, where X shows the coherent chapter and Y the figure number. Whenever a table or figure is used a describing text follows to explain the context. If the figure doesn't include a reference its made by the group itself. The chapters in the annex and appendix are written with letters to differentiate between main matter and attachments.
A CD will be attached to the printed project, including the annex, appendix and all SolidWorks parts with drawings. The drawings are also printed and assembled in a separate holder, to provide easy accessibility for the reader.
Software The following software has been used in the process of making this project.

Name	**Area of use**	**Name**	**Area of use**
LaTeX	Assembling the project	SolidWorks	3D drawing of the skateboard
Maple 16	Calculations	Ansys	FE analysis
Matlab	Calculations and scripts	Eagle	Circuit drawing of PCB

Thomas Pank Roulund

Contents

Chapter 1

Introduction

When setting up a robot today, the most important factor is having a structured environment. The meaning of this is, that the robot needs to be placed at a precise location and it takes time to configure it to do one type of operation. furthermore, if it is working with conveyor belt, for example, it is necessary to forward relevant information about speed and position to the robot control unit. Organisations like SPARC keeps researching into how to improve robots' capabilities in the industry, and they have released an article, with the purpose of describing in what direction research into robotics should go (SPARC, 2015, p. 190). The aim of this article, that are relevant for this report, is adaptability. This means that research needs to focus on making it easier to adept a robots for new work scenarios, environment and conditions.

Chapter 2

Problem Analysis

For this report, Aalborg University has made a robot with a conveyor belt available and have a demo assembly, a picture of the test setup is shown on figure 2.1.

Figure 2.1: The test setup made available by Aalborg University

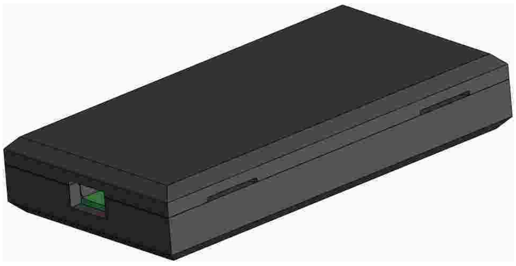
The University's desire is to design a visual system for the robot, which enables it to assemble the demo assembly without any information about the conveyor belt's velocity or position. Likewise the solution needs to be efficient, which means the robot needs to pick up the parts for the assembly while the conveyor belt is moving. The purpose of this chapter is to present the assembly, test setup and analysing which modifications are necessary before the robot is able to assemble the assembly. This leads to the solution strategy for the remainder of the report.

2.1 Test setup

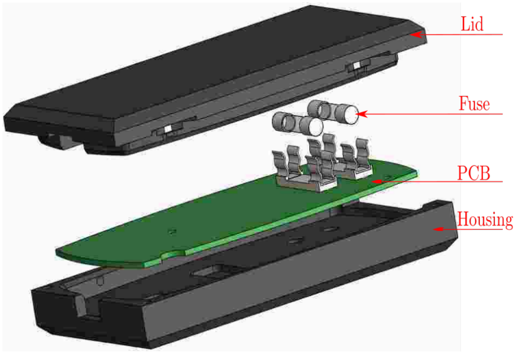
The purpose of this section is to present the demo assembly and its' individual parts. Furthermore this section introduces the test setup consisting of a Adept robot and a conveyor belt, and will analyse the physical work space and which modifications are necessary.

2.1.1 The Assembly

The demo assembly is made for production line tests and is shown on figure 2.2a and the assembly order is shown as a exploded view on figure 2.2b.



(a) The assembly

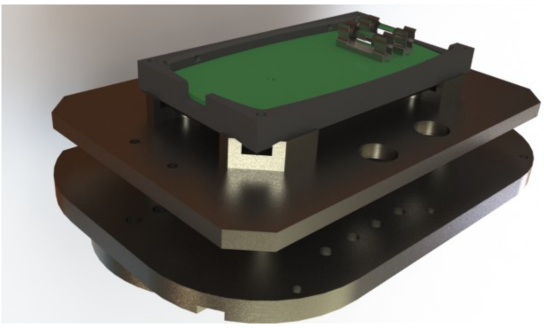


(b) Exploded view showing the assembly order of the assembly

Figure 2.2

The assembly consist of a housing, the PCB, two fuses which are inserted into a spring socket and a lid, the lid and housing comes in both black and blue. Today Aalborg University uses the demo assembly for their existing smart lab, which is a production line built with interchangeable modules. A picture of this production line is shown on figure 2.3a.

(a) The smart production lab at Aalborg University



(b) The assembly on the pallet for the smart production

Figure 2.3

The demo assembly is made for this smart production line and it is placed on a pallet, shown on figure 2.3b, which is then moved around the different modules via a conveyor belt. The smart production line is able to handle mass customisation because each pallet is equipped with a RFID chip. This chip is programmed with all information about what color the housing should be and what manufacturing operations need to be made to the assembly. As it is now, the pallet is moved to a operation station, where the conveyor belt stops until the operation is complete. If the operation is to drill a hole, it is necessary to stop while the hole is being driller. If it instead is to place the lid on the housing with a robot, it is more efficient for the robot to place the lid while the pallet with the housing is passing by. Aalborg University is therefore interested in developing a vision system which makes a robot place a lid with the correct color on the housing. As mentioned, this should be done without the vision system communicating with the smart production line. This means the vision system needs to detect position, orientation and velocity of the housing, such that the robot can match the speed and orient the lid such that it will fit on the housing. The reason for this is, such a solution is fast to implement and flexible since it does not set any requirements for positioning except that the robot should be able to reach the conveyor and there should not be anything the robot can hit within its' work space.

2.1.2 The Adept Robot

The robot used for this test setup is a Adept Cobra s600 robot, shown on figure 2.4 and consist of three revolute joints and one prismatic. These joints are indicated on the figure with red lines.

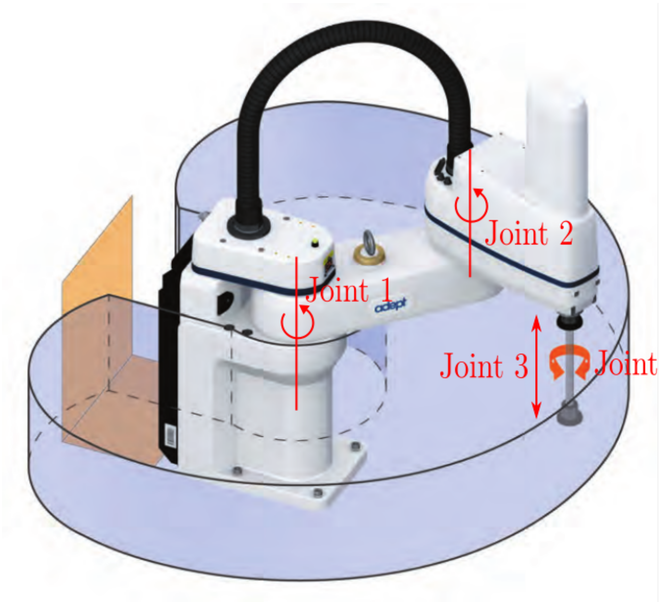


Figure 2.4: The available work space for the Cobra s600 (Adept, 2017, modified)

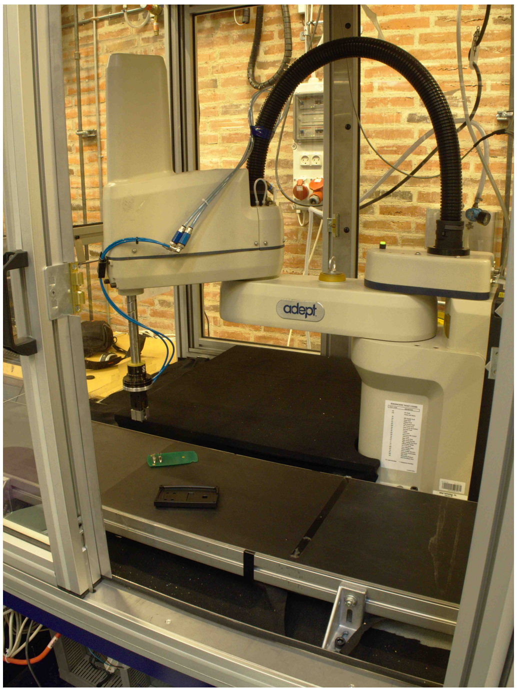
Joint 1	$\pm 105^o$
Joint 2	$\pm 150^o$
Joint 3	210 [mm]
Joint 4	$\pm 360^o$
Inner limit	163 [mm]
Outer limit	600 [mm]

Table 2.1: Joint ranges for the Cobra s600 (Adept, 2017)

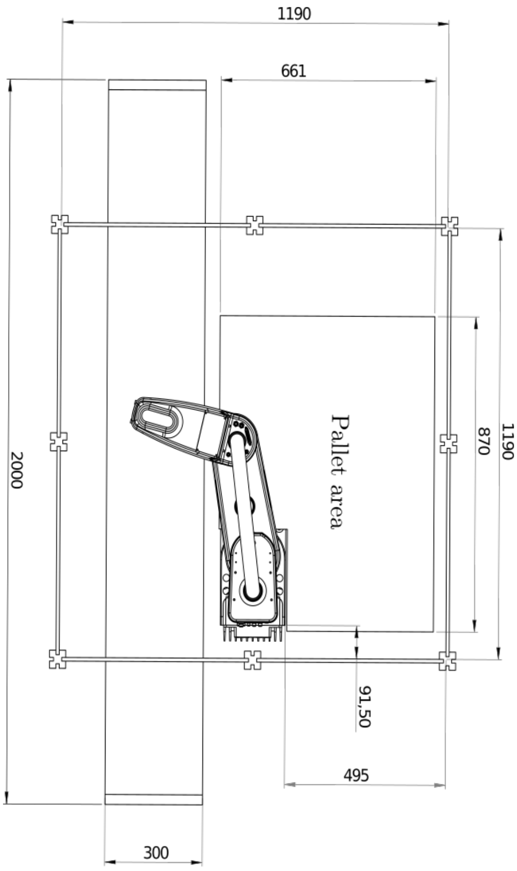
The work space the robot's end effector is able to reach is determined from the joints, and this available work space is indicated on figure 2.4 by the blue envelope. The joints' ranges are specified in the datasheet for the Cobra s600, and are summarised in table 2.1 along with the inner and outer radius of the work space.

2.1.3 Work space

With the work space of the Cobra defined, it is now necessary to define the available work space in the test setup. The Cobra is placed inside a cage next to a conveyor belt, shown on figure 2.5a. By making a CAD model of the work space, it is possible to better understand possibilities and limitations of it. A plan view of this CAD model is shown on figure 2.5b, where only important features are included.



(a) Picture of the actual test setup



(b) Plan view of the work space created from the CAD model, all measurements are in millimetre. The CAD model of the Cobra s600 is available online (Adept, 2017)

Figure 2.5

Inside the cage it is necessary to place two lid buffers, one in with blue and one with black lids, inside the pallet area of the plan view. The desired pick and place operation is then to pick a lid in the correct color and place it on the housing. The housing is place in an arbitrary position and orientation on the conveyor and moving with unknown speed that might change. The control of the conveyor belt is done by a frequency transformer, and is not connected to robot control unit. It is therefore necessary for the robot to match the speed of the conveyor belt, when performing the picking operation. For this test setup to be used for vision servoing of the Cobra, it is necessary to modify it with a camera for visual feedback and setup the communication between the robot control unit and the vision system. Furthermore, it is assumed it is necessary to install lighting to ensure proper visual feedback.

2.2 Solution strategy

When solving a complex problem as this problem of developing a vision system for a robot, it is possible to dedicate time and resources to a solution approach that turns out not to achieve the desired result. It is therefore important to come up with a solution strategy which ensures the available time and resources are not wasted. The purpose of this section is to present the solution strategy for this report.

2.2.1 Lean startup method

Solving the task of locating the housing on the conveyor belt and placing a lid on it, is complicated because there are several sub-problems which also needs to be solved to achieve the final goal. To ensure all these sub-problems are solved in an efficient way, one approach is to use the methodology called the lean startup method, and the idea is to follow the loop shown on figure 2.6

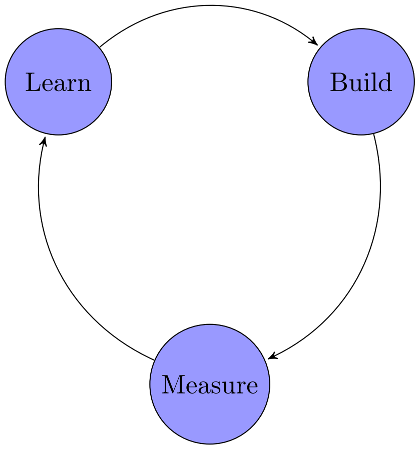


Figure 2.6: Lean startup ideology

With the context of this report, the approach is to spilt the problem into modules and then define a simple task for the modules, solve it and learn from the procedure. At each iteration the complexity of the modules' tasks are increase until the original goal is solved. The modules for this report are:

1. The modifications for the test setup; type of and number of cameras and lighting inside the cage
2. Assembling of the assembly; number of parts used and color of the parts
3. Conveyor belt; the maximum velocity of the conveyor belt and what kind of changes can happen to the velocity
4. Visual system; detecting the parts on the conveyor belt and deriving position, orientation and velocity
5. Control strategy and trajectory tracking for the robot; with the information from the visual system, it is necessary to plan a trajectory for the robot and control strategy to ensure the robot is tracking the trajectory
6. Communication between visual system and the robot control unit; how the control strategy output is translated into signals for the robot, while still ensuring the system bandwidth is sufficient

Module 1 to 3 are physical modules, where module 3 is defined to challenge the solution for each iteration. The modules 4 to 6 are software modules and should only depend on some specific inputs and give specific outputs such that it is possible to change them if the need arises.

Chapter 3

Problem Statement

With all the necessary steps to modify the test setup and necessary challenges to solve, it is possible to specify a problem statement which the remainder of this report works towards solving:

Is it possible with the modules defined in section 2.2.1 to use the Lean startup method to solve the goal of using visual control of the Cobra s600 to place a lid in the correct color on the housing part, without knowledge about the conveyor belt's velocity?

3.1 Project limitations

This section outlines some of the limitations of this report.

- Dette skal formuleres lidt bedre senere i projektperioden

  Change of conveyor belt velocity at the same time of the place operation. If the velocity of the conveyor belt changes just as the robot is about to place the lid on the housing, the vision system might not be able to detect it due to the bandwidth. In this case the lid might miss the housing.

Chapter 4

Vision System

The purpose of this chapter is to design a vision system which is able to identify

4.1 Camera

4.2 Lighting

4.3 Image processing

The purpose of this section is to identify the location housing part in a image. This is done in two parts, firstly by detecting the blob in the image and secondly calculating the centroid of the blob.

4.3.1 Blob detection

A single frame from the camera in the test setup is shown on figure 4.1.

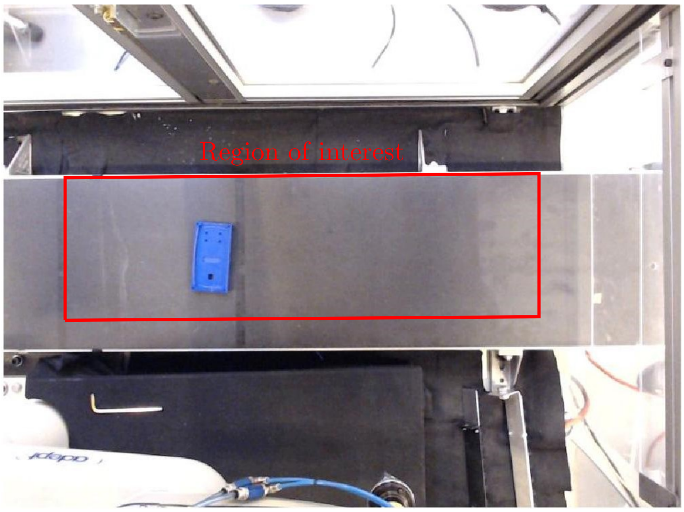


Figure 4.1: A single image from the camera in the test setup

First step is to define the region of interest, which is shown on the figure. This is done by defining the determining the pixels spanning this region, and creating a new image containing all the pixels within the region of interest. The resulting image is shown on figure 4.2.

Figure 4.2: The region of interest, which is the conveyor belt

With the new image only showing the housing part and conveyor belt, the next step is the algorithm needs to determine the difference between these two. The first attempt is background subtraction, the idea is, that if the background is static, then by subtracting it from the image, only the housing will remain. A base picture of the background is shown on figure 4.3.

Figure 4.3: The base image, which only shows the conveyor belt

Comparing this base image with figure 4.2, a conflict with the assumption about static background arises. The conveyor belt's appearance is not uniform and there are tape lines from previously experiments, these might cause noise later in the vision algorithm. Before the subtraction it is decided to work with greyscale images, so where the frame and base currently are in the RGB space, this is done with a weighted summation

$$f(u,v) = \alpha\, R + \beta\, G + \gamma\, B \tag{4.3.1}$$

$$\alpha + \beta + \gamma = 1 \tag{4.3.2}$$

In this first attemp $\alpha = 1$ and $\beta = \gamma = 0$. The background subtraction is done with the following equation:

$$g(u,v) = |f(u,v) - b(u,v)| \tag{4.3.3}$$

Where u is the pixel number in the vertical direction, v the pixel number in the horizontal direction, f is the actual frame and b is the base image, the absolute value is used, since it is likely some pixels' value might become negative and negative pixel values does not correspond to a nuance of grey. Figure 4.4 shows the result of subtracting the base image from figure 4.2.

Figure 4.4: The resulting image after the base is subtracted from the frame

The background subtraction did not manage to remove the entire background, which results in some additional grey areas in the frame besides where the housing is located. Next step is to try and remove these additional grey areas which is acting a noise in the frame. This is done by performing thresholding on the frame, which transform it from greyscale form into binary form. To determine which thresholding value to use, it is necessary to make a histogram for figure 4.4. The appropriate thresholding value is a compromise between false positive and false negatives, meaning backgrounds pixels which are set to 1 and pixels belonging to the housing set to 0. The histogram is shown on figure 4.5.

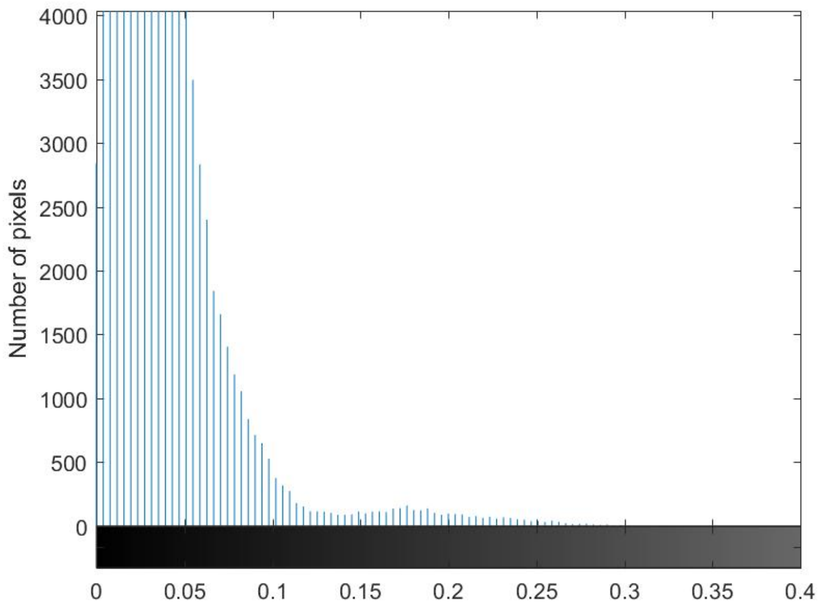


Figure 4.5: Histogram for the frame in figure 4.4

The leftmost part of the histogram is the background, so a candidate thresholding value is $t = 0.15$. The logic used for creating the resulting binary frame is:

$$T = \begin{cases} g(u,v) \geq t \rightarrow T(u,v) = 1 \\ g(u,v) < t \rightarrow T(u,v) = 0 \end{cases} \tag{4.3.4}$$

Doing this for the entire frame yields the binary frame shown on figure 4.6 A blob is defined as

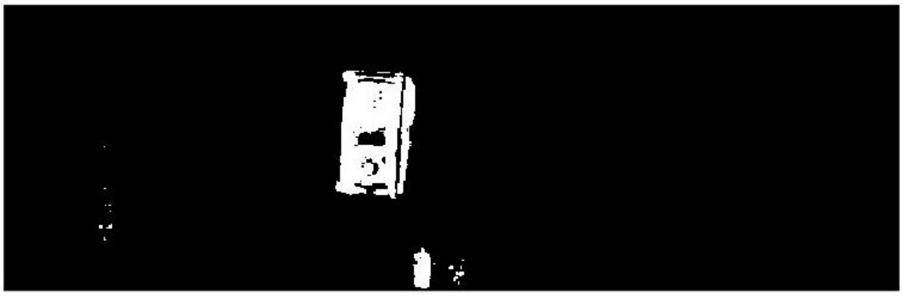


Figure 4.6: The resulting binary frame

a connected white area, but the problem here is, the blob generated by the housing is deformed and broken. Following the same approach for different frames, shows that the blob is always deformed and broken, so it is necessary to use a different method for blob detection.

HSI colorspace

In the book (Moeslund, 2012) it is proposed to use the HSI color space for video, since it is feasible to assume that the housing color is unique throughout the video sequence. The acronym HSI stands for; Hou, Saturation and intensity. Hou is the pure color, Saturation is the amount of white light mixed with the Hou color. Lastly the Intensity is calculated as. (Moeslund, 2012)

$$I = \frac{R+G+B}{3}, \quad I \in [0, 255] \tag{4.3.5}$$

The two remaining are calculated from the RGB color space as such: (Moeslund, 2012)

$$H = \begin{cases} cos^{-1}\left(\frac{1}{2}\frac{(R-G)+(R-B)}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}}\right), & G \geq B \\ 360^o - cos^{-1}\left(\frac{1}{2}\frac{(R-G)+(R-B)}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}}\right), & G < B \end{cases}, \quad H \in [0, 360[\tag{4.3.6}$$

$$S = 1 - 3\frac{\min\{R, G, B\}}{R+G+B}, \quad S \in [0, 1] \tag{4.3.7}$$

Doing this for the frame in figure 4.2 yields the new frame which is shown on figure 4.7.

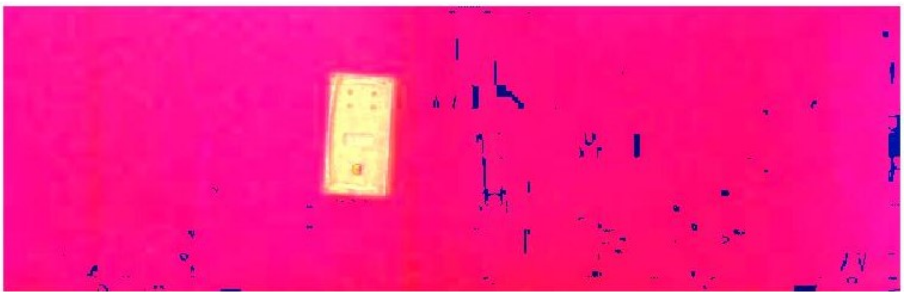
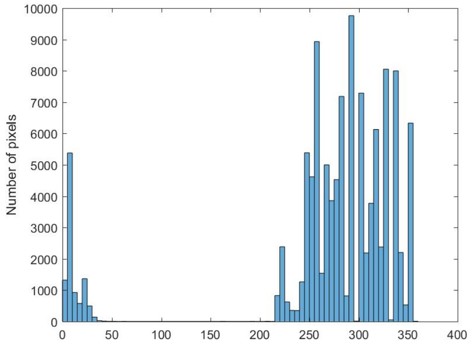
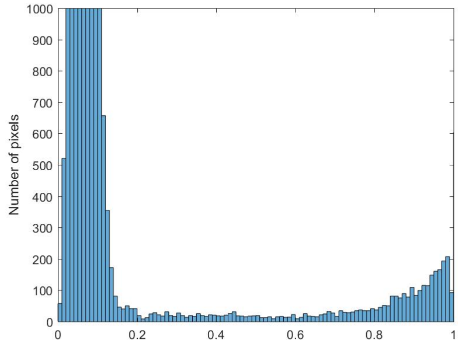


Figure 4.7: The frame converted into the HSI color space

There is a clear distinction between the housing and the conveyor belt, the most notable difference is the vertical lines on the conveyor, which caused noise before, are not viseable in the HSI frame. It is also possible to do background subtraction on the HSI frame, but to begin with, the thresholding is done on the Hou and Saturation values. The histogram for these two are shown on figure 4.8.



(a) Histogram for the Hou values in the HSI frame



(b) Histogram for the Saturation values in the HSI frame

Figure 4.8

By probing the pixel values on figure 4.7 it is possible to determine the range for the Hou and Saturation values, which are as follows:

$$H \in [180, 250] \tag{4.3.8}$$
$$S \in [0.4, 1] \tag{4.3.9}$$

The programming used for thresholding is:

```
for i = 1:ilg
        for j = 1:ihg
                if (HSI(i,j,1) >= 180 && HSI(i,j,1) <= 230
                    && HSI(i,j,2) >= 0.4 && HSI(i,j,2) <= 1)
                        g(i,j) = 1;
                else
                        g(i,j) = 0;
                end
        end
end
```

Where *ilg* and *ihg* is respectively the height and width of the frame in pixels. As before this thresholding yields a binary image, which is shown on figure 4.9.

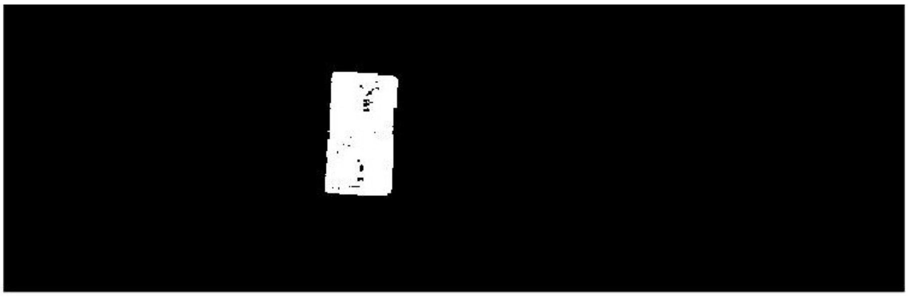


Figure 4.9: The resulting binary frame after thresholding the HSI frame

There is some small holes in the blob but all the background is removed and the blob is deemed a adequate representation of the housing.

4.3.2 Determination of centre and orientation

The purpose is to determine position and orientation of the housing such that the robot can place a lid correctly on it. The way this is done to begin with by the vision system, is to determine the centroid of the blob found in the previous section and afterwards it is possible to find the orientation. The first step is to determine the size of the blob and number of blobs in the frame, this is done by comparing a pixel to its' neighbours. If the neighbours are white pixels they are connected and if they are black, they are not. Thereby it is possible to set up a book keeping matrix of the same size as the frame which keeps track of which pixels are background and which pixels belongs to a given blob. This is all handled by the MATLAB command: (Corke, 2013)

```
blob = iblobs(g,'area',[min max])
```

This command takes the binary frame and only returns the blobs which areas are within the limits, the area is defined as the number of pixels within the blob. Once the blob is known it is necessary to determine the bounding box, this is the smallest possible box which still fully contain the blob. Firstly it is necessary to determine the maximum and minimum non-zero pixel in both the u- and v-direction within the blob, since these will form the corners of the box. The result of this is shown on figure 4.10, where the box derived from the binary frame is plotted on top of the frame from the camera. (Corke, 2013)

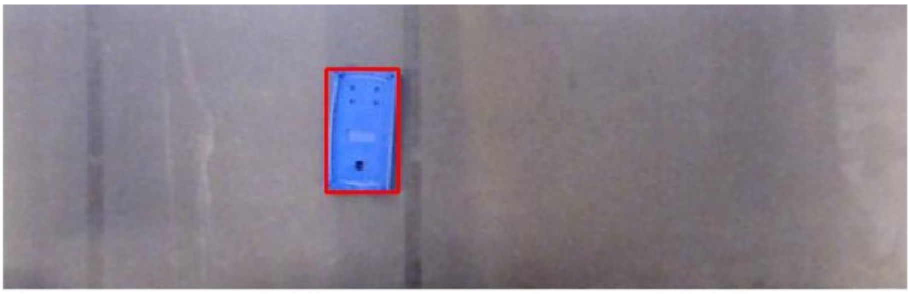


Figure 4.10: The frame from the camera where the bounding box is laid on top

As expected the box fully contains the housing, which verifies the quality of the binary frame. The next step is to calculate the center of the blob, this is done using the image moment. If the all the pixels within the housing blob and only these is said to be contained in the new frame $H(u, v)$, which is the same size as the binary frame, then the moment for H is defined as: (Corke, 2013)

$$m_{qp} = \sum_{(u,v)\in H} u^p \, v^q \, H(u,v) \tag{4.3.10}$$

The sum $(p+q)$ is the order of the moment, where the zero order moment is defined as

$$m_{00} = \sum_{(u,v)\in H} H(u,i) \tag{4.3.11}$$

A physical interpretation of the moments, is mass distribution where the pixels represent units of area and mass. With this in mind it is possible to calculate the centre of mass or more commonly within vision, the centroid of the region as:

$$u_c = \frac{m_{10}}{m_0} \tag{4.3.12}$$

$$v_c = \frac{m_{01}}{m_{00}} \tag{4.3.13}$$

Adding this centroid to the previous plot yields the result shown on figure 4.11 Lastly to calculate

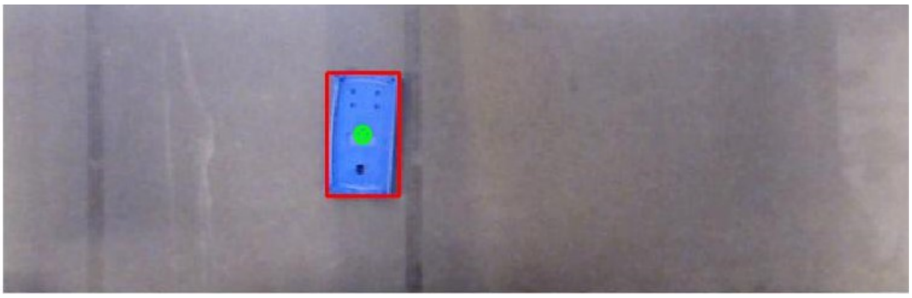


Figure 4.11: The frame from the camera with the bounding box and the centroid plotted

the orientation of blob, this is done by introducing the central moments μ_{pq} defined as:

$$\mu_{pq} = \sum_{(u,v)\in H} (u-u_c)^p (v-v_c)^q H(u,v) \tag{4.3.14}$$

The central moments are invariant with respect to position and relate to the moments in the following way:

$$\mu_{10} = 0 \tag{4.3.15}$$

$$\mu_{01} = 0 \tag{4.3.16}$$

$$\mu_{20} = m_{20} - \frac{m_{10}^2}{m_{00}} \tag{4.3.17}$$

$$\mu_{02} = m_{02} - \frac{m_{01}^2}{m_{00}} \tag{4.3.18}$$

$$\mu_{11} = m_{11} - \frac{m_{01}\, m_{01}}{m_{00}} \tag{4.3.19}$$

Giving these central moments a physical interpretation, as before, they describe the inertia of the blob and can be inserted in a inertia matrix.

$$J = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} \tag{4.3.20}$$

The orientation is calculated by finding a equivalent ellipse which has the same inertia matrix as H, this is done by using the eigenvectors of the inertia matrix. The eigenvectors gives the ellipses' principal axes. The major axis, v_y, correspond to the eigenvector associated to the largest eigenvalue of the inertia matrix and likewise the minor axis, v_x, is given by the eigenvector associated with the smallest eigenvalue. Thereby the orientation with respect to the horizontal axis is given by:

$$\theta = \tan^{-1}\left(\frac{v_y}{v_x}\right) \tag{4.3.21}$$

The equivalent ellipse is added to the frame from the camera on figure 4.12.

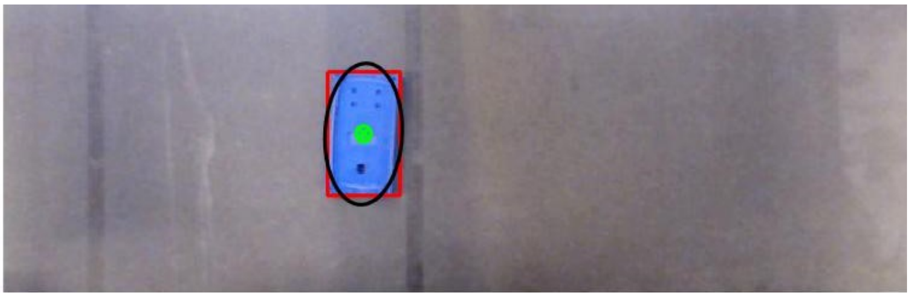
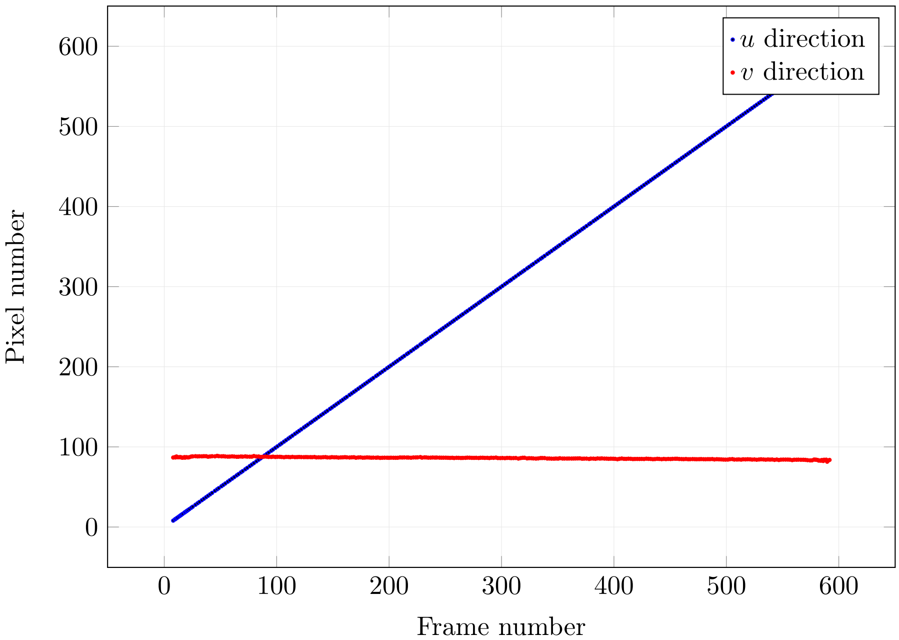


Figure 4.12: The frame from the camera with the bounding box, centroid and equivalent ellipse plotted

There is two considerations with the approach used for determination of position and orientation: firstly, for the equivalent ellipse to exist, the object needs to be rectangular, which is always the case for the housing.Secondly, the orientation is not to a specific point on the housing, since it is simplified to a blob, this means that say the housing is rotated 180^o on figure 4.12, the orientation will remain the same .

Er dette rigtigt?



u direction
v direction
600
500
400
300
200
100
0
Pixel number
0
100
200
300
400
500
600
Frame number

Chapter 5

Modeling

5.1 Kinematics

5.2 Dynamic model of tracked object

5.3 Kinematics

Following the standard David-Hartenberg convention for assignment of local coordinate systems, it is possible to derive the David-Hartenberg parameters. The local coordinate systems are shown on figure 5.1. From the figure it is possible to setup the DH-table, which is shown in table 5.1

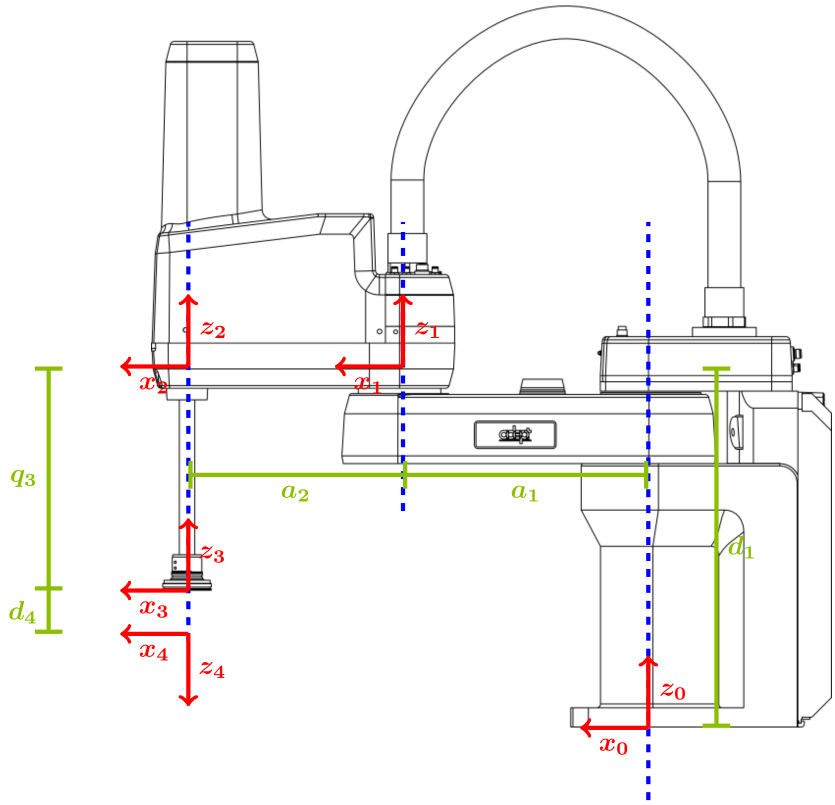


Figure 5.1: The Cobra s600 with local coordinate systems.

i	α_i	a_i	d_i	θ_i
1	0	a_1	d_1	q_1
2	0	a_2	0	q_2
3	0	0	q_3	0
4	π	0	d_4	q_4

Table 5.1: The David-Hartenberg parameters for the Cobra s600

The general DH-matrix is as follows when using compact notation:

$$ {}^{i-1}A_i(q_i) = \begin{bmatrix} c\theta_i & -c\alpha_i\, s\theta_i & s\alpha_i\, s\theta_i & a_i\, c\theta_i \\ s\theta_i & c\alpha_i\, c\theta_i & -s\alpha_i\, c\theta_i & a_i\, s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.3.1} $$

To derive the transformation matrix from the tool tip to the Cartesian space, it is necessary to multiply the four transformation matrices. This is done in appendix A. The final transformation matrix is:

$$
{}^{0}A_4 = \begin{bmatrix} c_{124} & s_{124} & 0 & a_1\, c_1 + a_2\, c_{12} \\ s_{124} & -c_{124} & 0 & a_1\, s_1 + a_2\, s_{12} \\ 0 & 0 & -1 & d_1 + q_3 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.3.2}
$$

Chapter 6

Solution Iterations

6.1 Iteration 1

6.2 Iteration 2

Todo list

Bibliography

Adept (2017). Adept cobra scara robots. http://www.adept.com/products/robots/scara/cobra-s600/downloads. Mar 2017. 5, 6

Corke, P. (2013). *Robotics, Vision and control*. Springer, 2 edition. 16, 17

Moeslund, T. B. (2012). *Introduction to Video and Image Processing*. Springer, 1 edition. 14

SPARC (2015). Multi-annual roadmap. *Strategic Research Agenda*. 1

Annex A

Diviation of transformation matrix

From section 5.3 the DH table is defined as And the general DH matrix is:

i	α_i	a_i	d_i	θ_i
1	0	a_1	d_1	q_1
2	0	a_2	0	q_2
3	0	0	q_3	0
4	π	0	d_4	q_4

Table A.1: The David-Hartenberg parameters for the Cobra s600

$$^{i-1}A_i(q_i) = \begin{bmatrix} c\theta_i & -c\alpha_i\, s\theta_i & s\alpha_i\, s\theta_i & a_i\, c\theta_i \\ s\theta_i & c\alpha_i\, c\theta_i & -s\alpha_i\, c\theta_i & a_i\, s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.0.1}$$

To derive the transformation matrix from the base of the robot to tool tip, it is necessary to substitute into the general DH matrix which yields four transformation matrices. The matrices are written in super compact form, which means $c_1 = \cos(q_1)$ and $s_{12} = \sin(q_1 + q_2)$:

$$^{0}A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1\, c_1 \\ s_1 & c_1 & 0 & a_1\, s_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.0.2}$$

$$^{1}A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2\, c_2 \\ s_2 & c_2 & 0 & a_2\, s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.0.3}$$

$$^{2}A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.0.4}$$

$$^{3}A_4 = \begin{bmatrix} c_4 & s_4 & 0 & 0 \\ s_4 & -c_2 & 0 & 0 \\ 0 & 0 & -1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.0.5}$$

These transformation matrices is now multiplied to derive the desired matrix.

$$ {}^{0}A_2 = {}^{0}A_1\,{}^{1}A_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1\,c_1 + a_2\,c_{12} \\ s_{12} & c_{12} & 0 & a_1\,s_1 + a_2\,s_{12} \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.0.6} $$

$$ {}^{0}A_3 = {}^{0}A_2\,{}^{2}A_3 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1\,c_1 + a_2\,c_{12} \\ s_{12} & c_{12} & 0 & a_1\,s_1 + a_2\,s_{12} \\ 0 & 0 & 1 & d_1 + q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.0.7} $$

$$ {}^{0}A_4 = {}^{0}A_3\,{}^{3}A_4 = \begin{bmatrix} c_{124} & s_{124} & 0 & a_1\,c_1 + a_2\,c_{12} \\ s_{124} & -c_{124} & 0 & a_1\,s_1 + a_2\,s_{12} \\ 0 & 0 & -1 & d_1 + q_3 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.0.8} $$

Equation (A.0.8) is the transformation matrix from the base to the tool tip.