Dynamic model

The calculations for the dynamic model are mostly built on “Modeling and Simulation of the LAUV Autonomous Underwater Vehicle” written by Jorge Estrela da Silva, Bruno Terra, Ricardo Martins and Joao Borges de Sousa.

The dynamic model consists of the required matrixes and functions to describe the AUV’s dynamic response when sailing a mission. The dynamic model is a solution to the following equation

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{v}} \\ \dot{\eta} \end{bmatrix} = A \cdot \begin{bmatrix} \boldsymbol{v} \\ \eta \end{bmatrix} + \ B \cdot \boldsymbol{\tau}$$

Where

$$A = \begin{bmatrix} M^{-1}[-C(v) - D(v) - L(v)] & -M^{-1}G(\eta) \\ J(\eta) & 0 \end{bmatrix}$$

and

$$B = \begin{bmatrix} M^{-1} \\ 0 \end{bmatrix}$$

The model responds to change in the velocity and position components, which are

$$\boldsymbol{v} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \begin{matrix} surge \\ sway \\ heave \\ roll \\ pitch \\ yaw \end{matrix} \text{ and } \eta = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix}$$

Where each of the parameters is as described in “Modeling and Simulation of the LAUV Autonomous Underwater Vehicle”, most of the values for the matrixes are taken from the same report but some are gotten from Simulink model made by Valur Einarsson in 2003 for the Gavia AUV. The M is the constant inertia and the added mass matrix of the AUV, $C(v)$ is the Coriolis centripetal matrix, $D(v)$ is the damping matrix, $L(v)$ is the lift matrix $G(\eta)$ is the matrix of resorting forces. The M^{-1} matrix is the numeric invers of M where

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 & mz_G & 0 \\ 0 & m - Y_{\dot{v}} & 0 & -mz_G & 0 & 0 \\ 0 & 0 & m - Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & -mz_G & 0 & I_x - K_{\dot{p}} & 0 & 0 \\ mz_G & 0 & 0 & 0 & I_{\dot{q}} & 0 \\ 0 & \ddot{o} & 0 & 0 & 0 & I_{\dot{z}} - N_{\dot{r}} \end{bmatrix}$$

The matrix of resorting forces $G(\eta)$ was simplified as the center of buoyancy and center of mass for the Gavia where estimated to be in the center of Gavia's coordinates, $z_B = x_G = x_B = y_G = y_B = 0$, except for a small offset of the center of gravity in z direction $z_G = 0.1$. The resulting resorting force matrix is

$$G(\eta) = \begin{bmatrix} 0 & 0 & 0 & 0 & W-B & 0 \\ 0 & 0 & 0 & -(W-B) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & z_G & 0 & 0 \\ 0 & 0 & 0 & 0 & z_G W & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

W is the weight and B is the buoyancy of the Gavia.

The force vector τ had to be calculated and modified for the Gavia, it consists of the product of the matrix K and an input vector from the Gavia rudders angle of attack (α) and the RPM (δ_n) of the propeller.

$$K = \begin{bmatrix} K_{prop} & -0_{drag} & -0_{drag} & -0_{drag} & -0_{drag} \\ 0 & F_{lift} & F_{lift} & 0 & 0 \\ 0 & 0 & 0 & F_{lift} & F_{lift} \\ 0 & 0_{roll} & 0_{roll} & 0_{roll} & 0_{roll} \\ 0 & 0 & 0 & -x_r F_{lift} & -x_r F_{lift} \\ 0 & -x_r F_{lift} & -x_r F_{lift} & 0 & 0 \end{bmatrix}$$

The input vector is

$$\boldsymbol{input} = \begin{bmatrix} 1 \\ \alpha_{r\_UR} \\ \alpha_{r\_LR} \\ \alpha_{r\_RP} \\ \alpha_{r\_LP} \end{bmatrix} \cdot {\delta_n}^2$$

The result is each of the force and moment components in x, y and z

$$\boldsymbol{\tau} = \begin{bmatrix} X_\tau \\ Y_\tau \\ Z_\tau \\ K_\tau \\ M_\tau \\ N_\tau \end{bmatrix} = K \cdot \boldsymbol{input}$$

The calculation for the force on Gavia's rudders is based on Helgi Þorgilsson master thesis "Control of a Small Unmanned Underwater Vehicle Using Zero Optimized PID Controllers" where the angle of attack is as can be seen in figure 1. The angle of attack produces a force $F_{lift} = \frac{1}{2}\rho A r_{blade}^2 2\pi$ and the propeller produces a force $K_{prop} = \frac{1}{2}\rho A 2\pi sin(\alpha) r_{blade}^2$. The airfoil area (A) and the propeller force (K_{prop}) are estimated as in Helgi's master thesis where $K_{prop} = 95 \cdot 10^{-6}$. Forces that produce roll and drag on the Gavia are neglected.

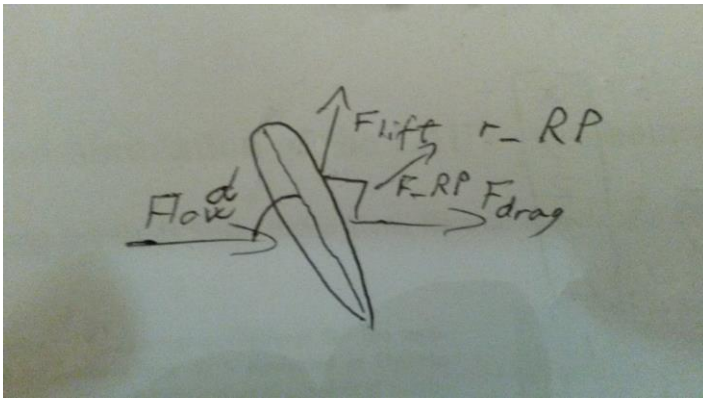

Figure 1: Angle of attack for the rudders.

The names of the corresponding rudders are as shown in figure 2

- Upper rudder => r_UR
- Lower rudder => r_LR
- Right rudder => r_RP
- Left rudder => r_LP

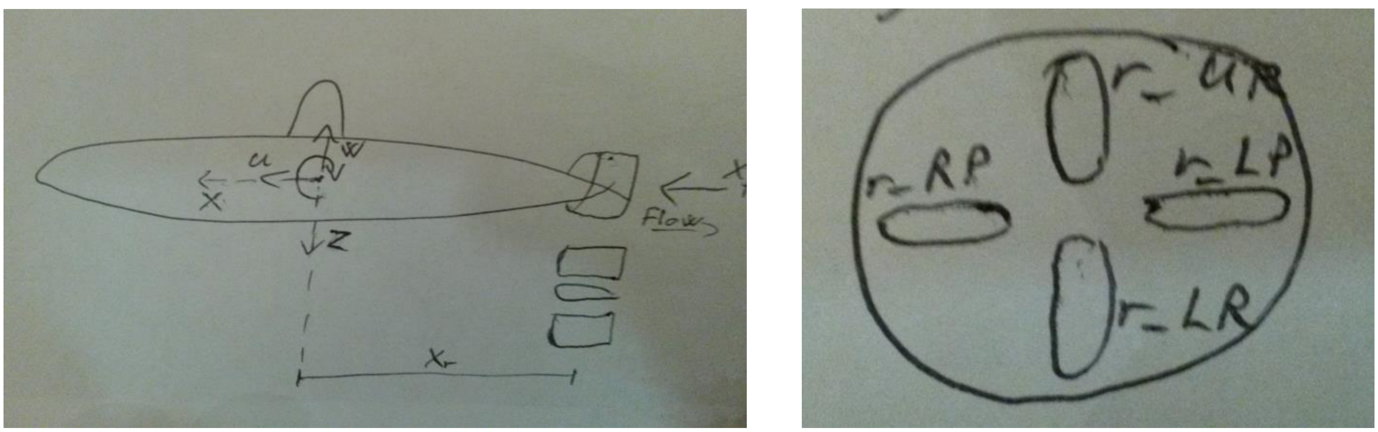

Figure 2: Position of rudders shown.

The dynamic model is written in C++ and depends on Boost 1.56 C++ libraries. Example of usage with the initial state $\boldsymbol{x} = [u, v, w, p, q, r, x, y, z, \phi, \theta, \psi] = [surge, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ can be seen in figure 3.

```cpp
#include "DynamicModel.h"


int main(int argc, char **argv)
{
    double t = 0.0;
    const double dt = 0.1;
    double surge = 0.5;
    double initState[12*1] = {surge, .0, .0, .0, .0, .0, .0, .0, .0, .0, .0, .0};
    AUVModel* model = new AUVModel();
    AUVModel* integrator = model->create(model, initState, t, dt);

    for(size_t i=0 ; i<50000 ; ++i,t+=dt )
    {
        integrator->dostep(model);
        model->input(surge/0.0025, 0, 0, 0, 0);
    }

    delete integrator;
    return 0;
}
```

Figure 3: Example usage of the dynamic model

Bottom and Navigation sonar model

The navigation sonar model represents the sonar beam with lines starting for the noes of the Gavia and extends for a set max sonar range looking for an intersection with the bottom. If an intersection is detected then the nearest intersection with the bottom is reported as the range to bottom.

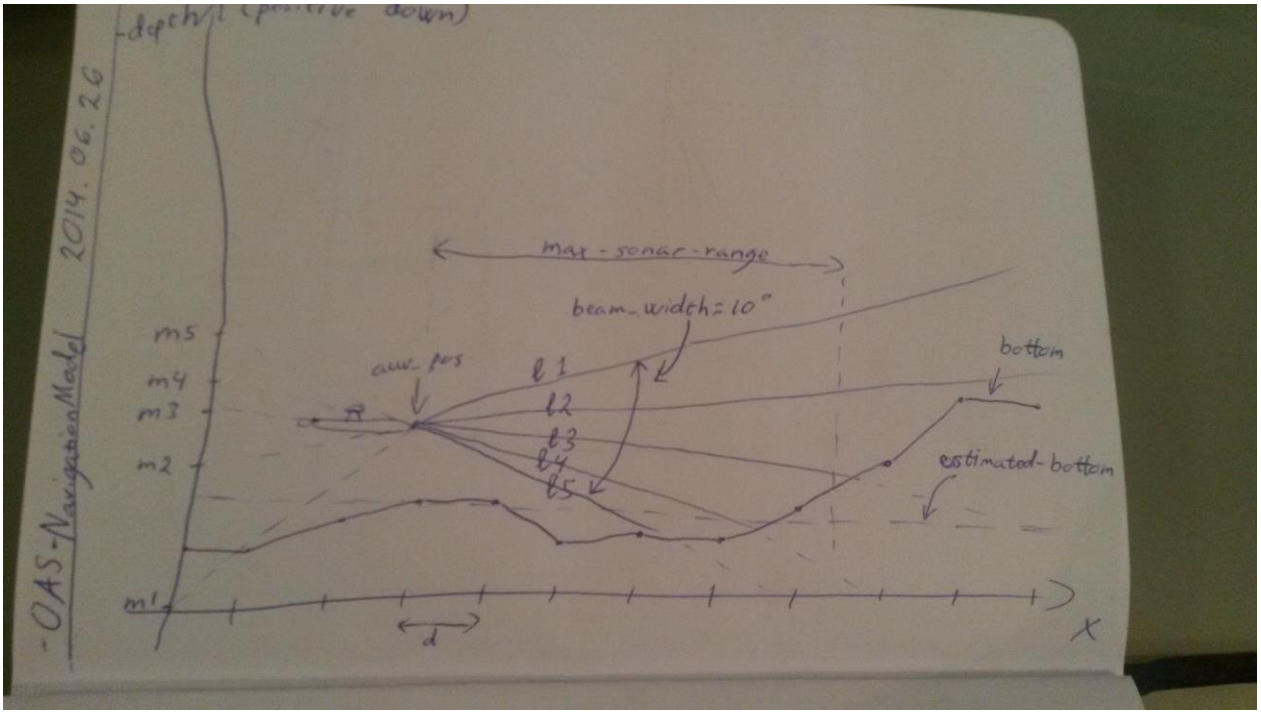

Figure 4: Demonstration of the navigation sonar model

Example of use for the navigation sonar and the bottom model can be seen in figure 6.

The bottom can be generated with generate_bottom.py which takes a JPG image as an input parameter. The JPG image should be a white image with one pixel thick black line representing the bottom, drawn from end to end as can be seen in figure 5. The result will be an array representation of the drawn bottom in a bottom.txt file which the bottom model can interpret as a bottom.

Figure 5: The bottom drawn on 100x300 pixels image

```cpp
#include "navigationSonar_model.h"

using namespace TestOAS;

int main()
{
    double range;
    const char* filename = "bottom.txt"
    coordinate auvPos(.0, .0);
    int numbBeams = 4;
    int maxRange = 50;
    NavigationModel sonarBeam(numbBeams, maxRange);
    BottomModel bottom;

    bottom.readFile(filename);
    double step = 0.5;
    for (auvPos.x = 0; !bottom.done(auvPos.x); auvPos.x += step)
    {
        range = sonarBeam.getRange(auvPos, bottom);
        /*
         * React to changes in range
         */
    }
    return 0;
}
```

Figure 6: Example usage for the navigation sonar

Plotter

The plotter.py file is able to plot the desired data if the data is stored in the correct format. The format for the data should be like here below and one of the titles must be x_pos for the plotter to know which row to plot against.

title1, title2, title3, title4,.....,

data1, data2, data3, data4,.....,

data1,data2,...

.

.

The plotter takes the data file as an input parameter along with a number one to three which tells the plotter in what manner the data should be displayed. Example on how to invoke the plotter with

terminall
$ python plotter.py data.txt 1

If case one is chosen then all the data is sub plotted on the same graph, as can be seen in figure 7. If case

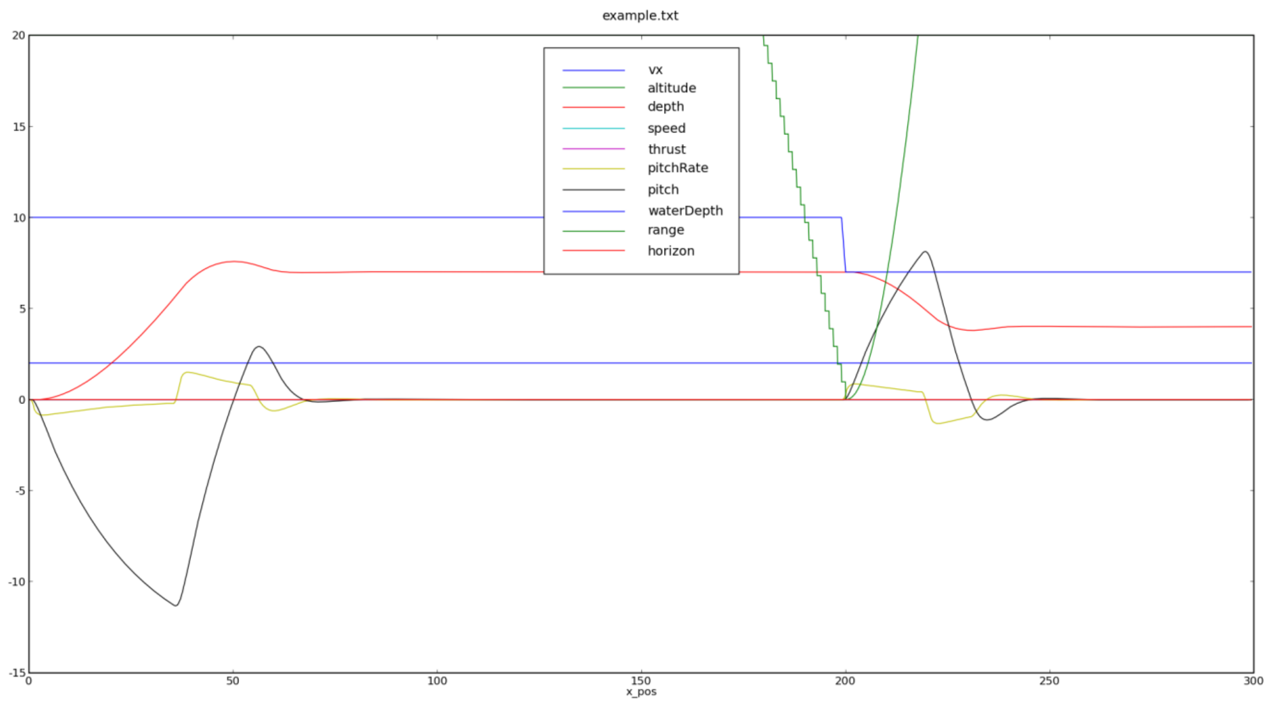

Figure 7: All the data sub plotted

two is chosen then each data in the data file is plotted against the x_pos as can be seen in figures 8 to 11. If case three is chosen then the movement of the Gavia over the bottom is shown along with the range of the navigation sonar. Case three is shown in figures 12 to 17.

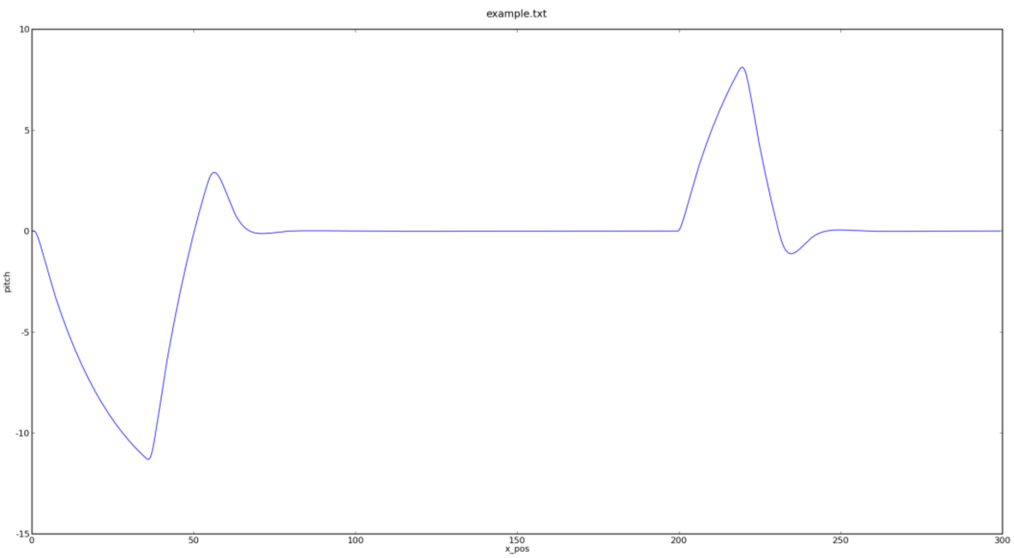

Figure 8: Pitch plotted against x_pos

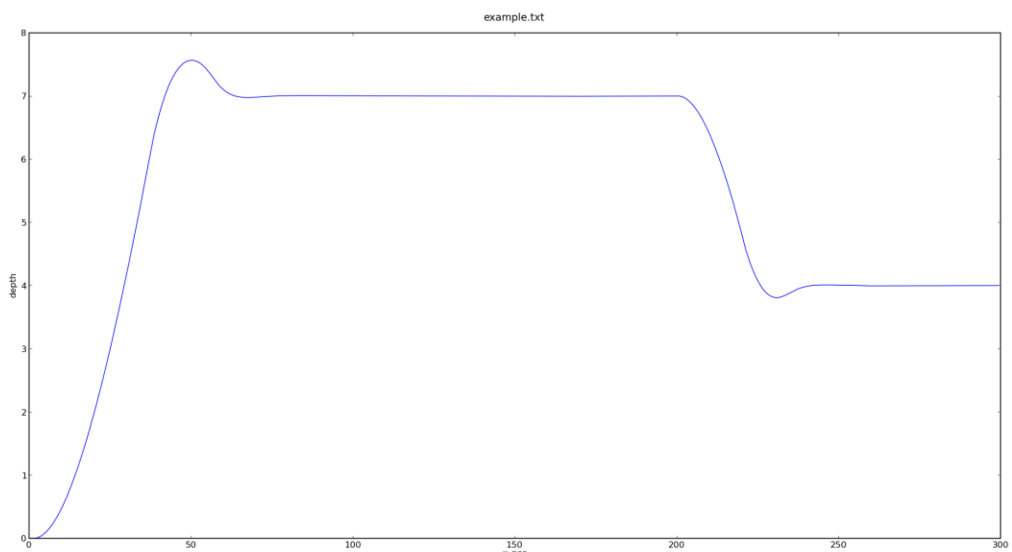

Figure 9: Depth plotted against x_pos

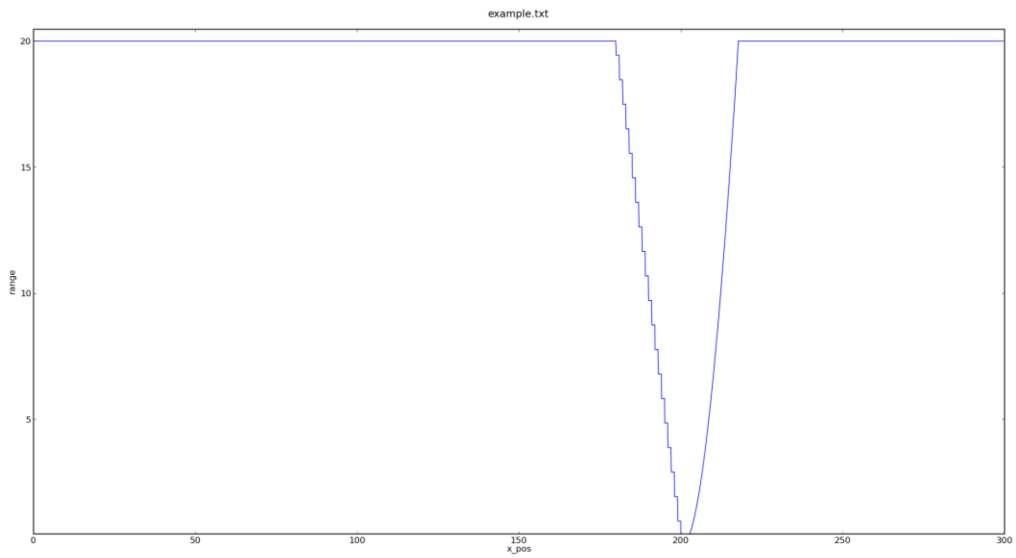

Figure 10: Navigation sonar range plotted against x_pos

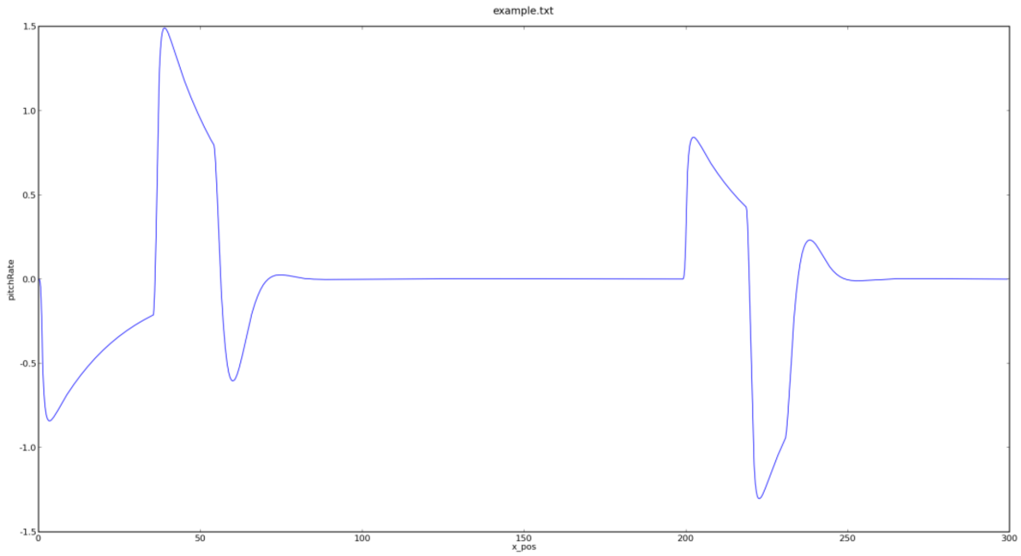

Figure 11: Pitch rate plotted against x_pos

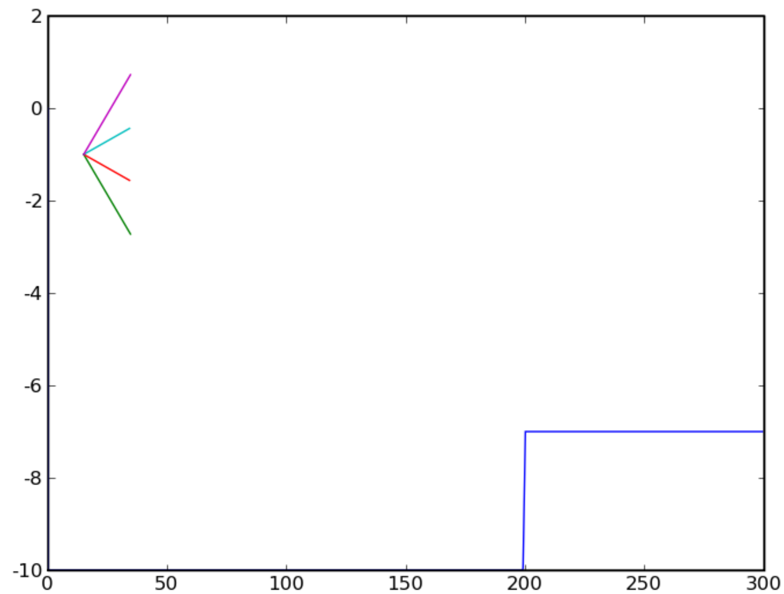

Figure 12: The movement of the sonar beam over the bottom

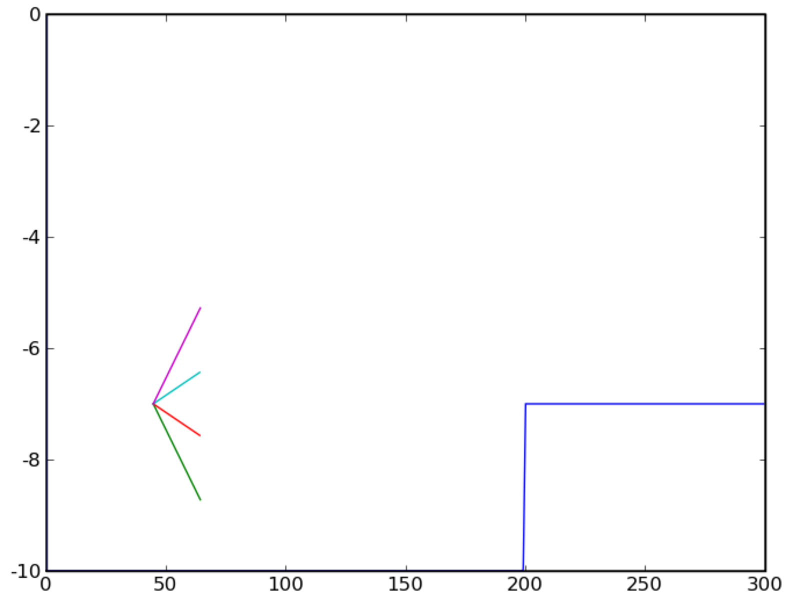

Figure 13: The movement of the sonar beam over the bottom

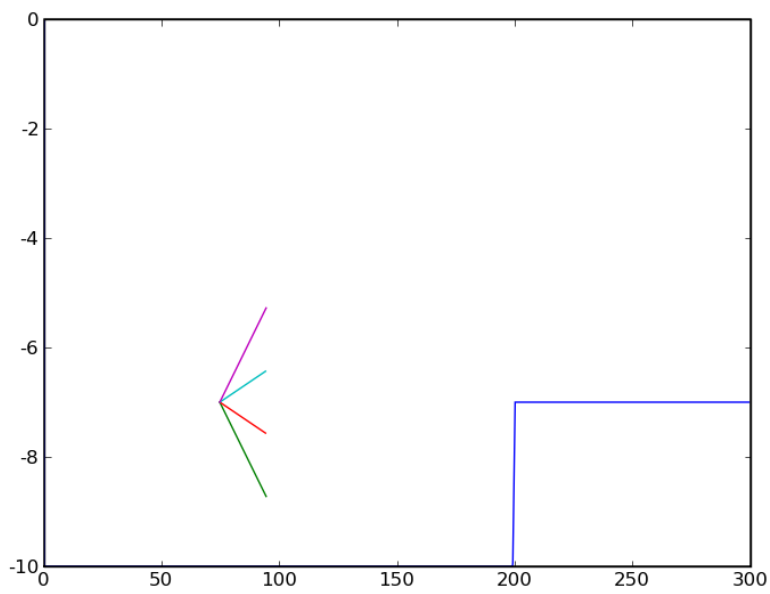

Figure 14: The movement of the sonar beam over the bottom

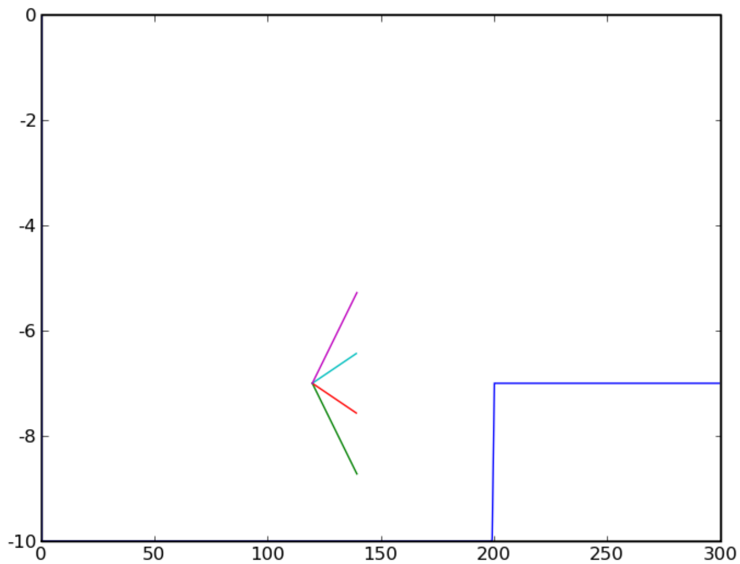

Figure 15: The movement of the sonar beam over the bottom

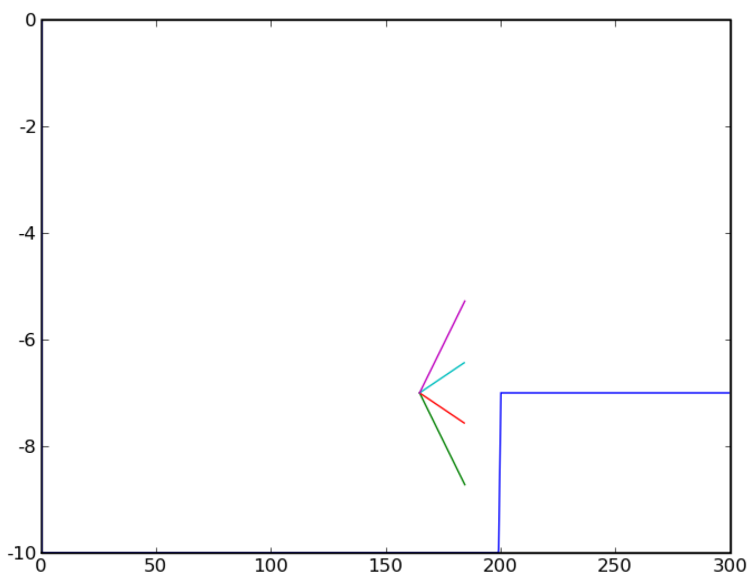

Figure 16: The movement of the sonar beam over the bottom

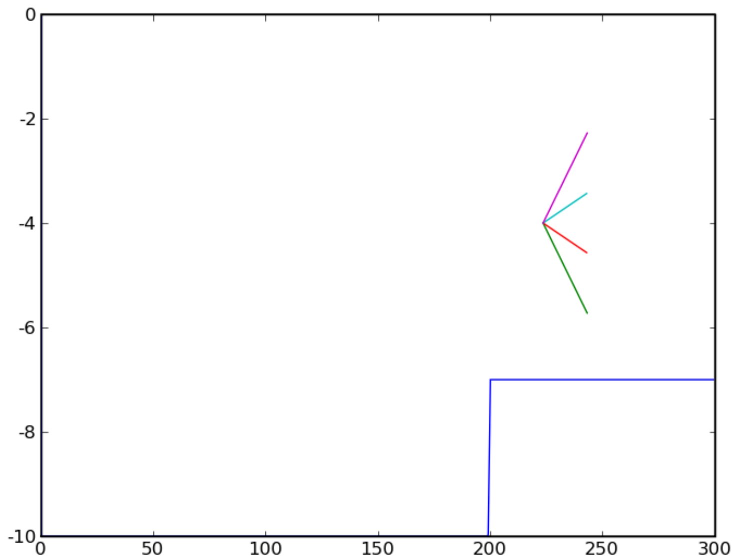

Figure 17: The movement of the sonar beam over the bottom