BAYESIAN ROBOTICS ME 5524

SPRING 2018

REPORT 2 – CHALLENGE 1 OF MBZIRC 2019

TEAM 2

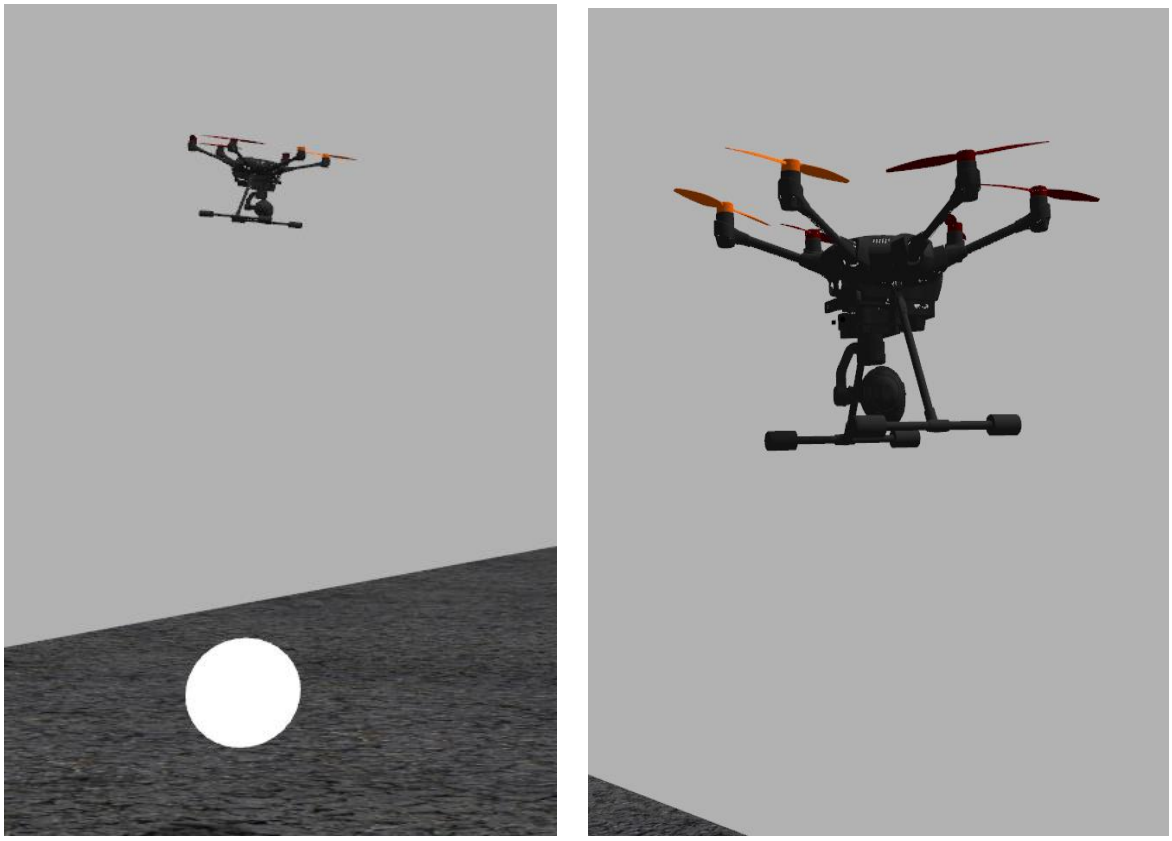
Ioannis Papakis, Sijing Guo, Rui Lin

All members participated equally.

Contents

I. UAV/UGV used and their motion/sensor models

Two UAVs were used in our simulation environment. One UAV will be carrying a ball underneath as the target. The other UAV will try to track and catch the target. The UAVs used were both Typhoon 4800, and the UAV model were provided to us in the simulation environment. Motion model and sensor model were developed based on the UAVs used.



(a) (b)

Fig. 1 (a) Target drone with a white ball, (b) Robot drone with two cameras

II. Approximate motion/sensor models for RBE

II.I Motion model

In this project, both the robot and the target are moving and thus we did a series RBE for both them. Motion models are built for both the robot and the target. For the robot, the control input of the motion model is set by us and the motion model is used for the prediction step. The correction step is then combined with the GPS and IMU. For the target, the control input of the motion model is set as a random vector obeying a Gaussian distribution and the motion model is used for the prediction step. The correction step is then combined with the two cameras.

Two types of motion model are developed, the dynamic motion model and the random walk motion model. The dynamic motion model is relatively complicated and thus we did not apply it in the project at this stage. The random walk model is selected in the project, due to its simplicity and easy implementation.

A. Random walk model

A.1 Model of the target

The random walk motion model is given by

$$\dot{\mathbf{x}}_k^t = \mathbf{w}_k^t, \mathbf{w}_k^t \sim N(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{w}}^{\mathbf{t}})$$

$$\begin{cases} \dot{x}_k^t = \left(w_x^t\right)_k \\ \dot{y}_k^t = \left(w_y^t\right)_k \\ \dot{z}_k^t = \left(w_z^t\right)_k \end{cases}$$

where the vector $\boldsymbol{x}_k^t = [x_k^t, y_k^t, z_k^t]^T$ and it represents the spatial position of the Hexacopte at three directions. The $\boldsymbol{w}_k = \left[w_x^t, w_y^t, w_z^t\right]^T$ and it represents the velocity of the Hexacopte at three directions. Since the velocity input of the target is supposed to be unknown, they are treated as random values obeying the Gaussian distribution.

The corresponding discrete motion model will be used for RBE and it is represented by

$$\begin{aligned} \mathbf{x}_k^t &= \mathbf{x}_{k-1}^t + \Delta t \dot{\mathbf{x}}_{k-1}^t \\ &= \mathbf{x}_{k-1}^t + \Delta t \mathbf{w}_{k-1}^t \\ \mathbf{w}_{k-1}^t &\sim N(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{w}}^t) \end{aligned}$$

where the covariance matrix is roughly selected based on the prediction values and the measured values. The numbers will be introduced in the RBE part.

A.2 Model of the robot

For the robot, the velocity input is known, and its motion model is represented by

$$
\begin{aligned}
\mathbf{x}_k^r &= \mathbf{x}_{k-1}^r + \Delta t \dot{\mathbf{x}}_{k-1}^r \\
&= \mathbf{x}_{k-1}^r + \Delta t(\mathbf{v}_{k-1}^r + \mathbf{w}_{k-1}^r) \\
&= \mathbf{x}_{k-1}^r + \Delta t \mathbf{v}_{k-1}^r + \Delta t \mathbf{w}_{k-1}^r \\
\mathbf{w}_{k-1}^r &\sim N(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{w}}^{\mathbf{r}})
\end{aligned}
$$

where the velocity input $\mathbf{v}_k^r = \left[v_x^r, v_y^r, v_z^r\right]_k^T$, and uncertainly caused by the wind speed stuff is represented by $\mathbf{w}_k^r = \left[w_x^r, w_y^r, w_z^r\right]_k^T$. The covariance matrix is roughly selected based on the prediction values and the measured values. The numbers will be introduced in the RBE part.

B. Dynamics motion model

B.1 Dynamics model of the target

In this section, we built the motion model based on its dynamics first. Then, the terms related with the 6 motor rotational speed inputs are treated as random values and thus the motion model is used for prediction of the target.

There are two frames for the Hexacopter, the body frame and the earth frame, as shown in Fig. X1. In the earth frame, the Hexacopter position is described by the vector $\boldsymbol{\xi} = [x, y, z]^T$. In the body frame, the Hexacopter orientation, attitude and heading is described by $\boldsymbol{\eta} = [\phi, \theta, \varphi]^T$. In the body frame, the linear and rotational speeds of the Hexacopter are given by $\boldsymbol{V} = [\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}]^{\boldsymbol{T}}$ and $\boldsymbol{\omega} = [\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r}]^{\boldsymbol{T}}$.

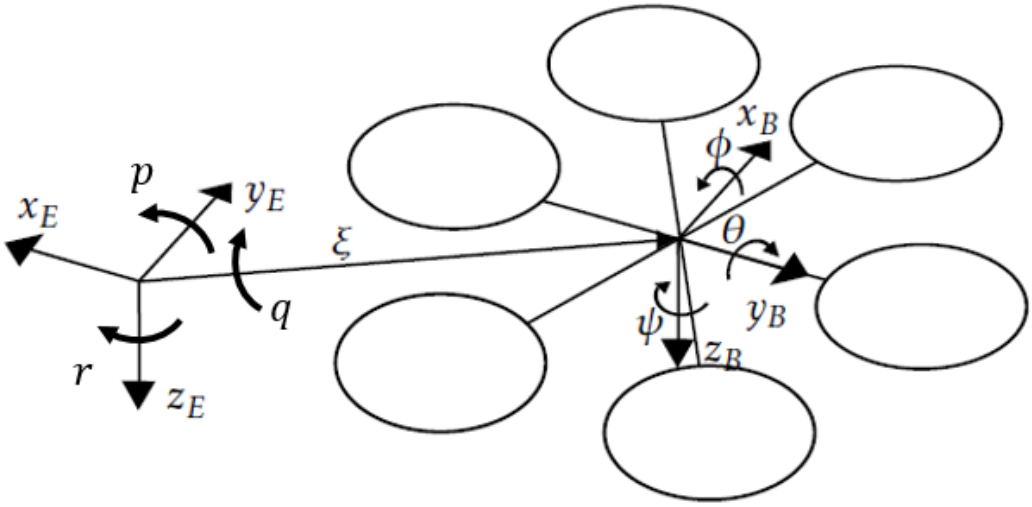


Fig. 2 Two frames of hexa-copter

The Newton second law can only be applied in the earth frame. Thus, we need transform the body frame to the earth frame. The transformations are given by

$$
\dot{\boldsymbol{\xi}} = \mathbf{R}_{\mathbf{L}}^{\mathbf{T}} \mathbf{V}, \ \dot{\boldsymbol{\eta}} = \mathbf{R}_{\mathbf{A}}^{-1} \boldsymbol{\omega} \tag{1}
$$

where

$$\mathbf{R_L} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\psi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{bmatrix}, \mathbf{R_A^{-1}} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{s\theta} & \frac{c\phi}{c\theta} \end{bmatrix}.$$

The *c*, *s*, *t* in Equation (1) are cos, sin, tan. $\mathbf{R_L^T} = \mathbf{R_L^{-1}}$.

By applying the Newton second law, the translational motion equation is given by

$$\mathbf{F}^E = m\ddot{\boldsymbol{\xi}} \tag{2}$$

where the upper right notation E is for the earth frame and B for the body frame.

Since it would be much more convenient to express the dynamics in the body fixed frame, with the rotational matrix from Equation (1), we can rewrite Equation (2) in the body frame as

$$\mathbf{R_L}\mathbf{F}^B = m\mathbf{R_L}\left(\frac{d\mathbf{V}}{dt}\right)^E \tag{3}$$

By using the transport theorem, the derivative of V in the inertial frame can be transformed into the body frame, as

$$\mathbf{F}^B = m\left(\left(\frac{d\mathbf{V}}{dt}\right)^B + \boldsymbol{\omega} \times \mathbf{V}\right) \tag{4}$$

Hence, the final equation of linear motion can be expressed as

$$\mathbf{F}^B = m\left(\frac{d\mathbf{V}}{dt}\right)^B + m\boldsymbol{\omega} \times \mathbf{V} \tag{5}$$

Similarly, we can get the equation of rotational motion, as

$$\mathbf{M}^B = J\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times J\boldsymbol{\omega} \tag{6}$$

The translation forces include the gravity, the air friction, and the thrust force, which is produced by the aerodynamic forces. The rotational torques include the aerodynamic torque, gyroscopic torque, and the torque induced by the air friction.

The total aerodynamic forces in the three linear directions are given by

$$\mathbf{F_A} = \begin{bmatrix} 0 \\ 0 \\ -k_T(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2 + \Omega_5^2 + \Omega_6^2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ F_w \end{bmatrix} \tag{7}$$

where k_T is the propeller specific constant, and Ω_i is the rotational speed of propeller i.

The aerodynamic torques around x_b, y_b, z_b axis are also expressed as

$$\mathbf{M_A} = \begin{bmatrix} -\frac{l}{2}k_T\Omega_1^2 - l\Omega_2^2 - \frac{l}{2}k_T\Omega_3^2 + \frac{l}{2}k_T\Omega_4^2 + l\Omega_5^2 + \frac{l}{2}k_T\Omega_6^2 \\ \frac{\sqrt{3}l}{2}k_T\Omega_1^2 - \frac{\sqrt{3}l}{2}k_T\Omega_3^2 - \frac{\sqrt{3}l}{2}k_T\Omega_4^2 + \frac{\sqrt{3}l}{2}k_T\Omega_6^2 \\ -k_Q\Omega_1^2 + k_Q\Omega_2^2 - k_Q\Omega_3^2 + k_Q\Omega_4^2 - k_Q\Omega_5^2 + k_Q\Omega_6^2 \end{bmatrix} = \begin{bmatrix} M_p \\ M_q \\ M_r \end{bmatrix} \tag{8}$$

where k_Q is the propeller specific torque constant.

The gravity force in the body frame is

$$\mathbf{F}_G^B = \mathbf{R_L}\begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = mg\begin{bmatrix} s\theta \\ s\phi c\theta \\ c\phi c\theta \end{bmatrix} \tag{9}$$

The total gyroscopic torque is

$$\mathbf{M_G} = \begin{bmatrix} qJ_{p,zz}(-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 - \Omega_5 + \Omega_6) \\ -pJ_{p,zz}(-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 - \Omega_5 + \Omega_6) \\ 0 \end{bmatrix} = \begin{bmatrix} qJ_{p,zz}W_G \\ -pJ_{p,zz}W_G \\ 0 \end{bmatrix} \tag{10}$$

where $J_{p,zz}$ is the rotational inertia around z axis.

The force and torque from the air friction are expressed as

$$\begin{aligned} \mathbf{F_R} &= -\mathbf{A_T}\mathbf{V} \\ \mathbf{M_R} &= -\mathbf{A_R}\boldsymbol{\omega} \end{aligned} \tag{11}$$

where $\mathbf{A_T}$ *and* $\mathbf{A_R}$ are diagonal matrices with diagonal elements a_T *and* a_R respectively.

The rotational speed of the propeller motor can be easily controlled by the input voltage, shown in the following equation.

$$\Omega = \frac{V_{in} - IR}{k_v} \tag{12}$$

where I is the current, V_{in} is the voltage input, and k_v is the speed constant.

Substitution of Equation (7) ~ (11) to Equation (5)(6) yields the final system model, as

$$
\begin{aligned}
\dot{x} &= c\theta c\psi u + (s\phi s\theta c\psi - c\phi s\psi)v + (c\phi s\theta c\psi + s\phi s\psi)w \\
\dot{y} &= c\theta s\psi u + (s\phi s\theta s\psi + c\theta s\psi)v + (c\phi s\theta s\psi - s\phi c\psi)w \\
\dot{z} &= -s\theta u + s\phi c\theta r + c\phi c\theta w \\
\dot{u} &= rv - qw + s\theta g - \frac{a_T}{m}u \\
\dot{v} &= pw - ru + s\phi c\theta g - \frac{a_T}{m}v \\
\dot{w} &= qu - pv + c\phi c\theta g - \frac{1}{m}F_w - \frac{a_T}{m}w \\
\dot{\phi} &= p + s\phi t\theta q + c\phi t\theta r \\
\dot{\theta} &= c\phi q - s\phi r \\
\dot{\psi} &= \frac{s\phi}{c\theta}q + \frac{c\phi}{c\theta}r \\
\dot{p} &= \frac{J_{yy} - J_{zz}}{J_{xx}}qr + \frac{1}{J_{xx}}M_p + \frac{J_{p,zz}}{J_{xx}}qW_G - \frac{a_R}{J_{xx}}p \\
\dot{q} &= \frac{J_{zz} - J_{xx}}{J_{xx}}rp + \frac{1}{J_{yy}}M_q - \frac{J_{p,zz}}{J_{yy}}pW_G - \frac{a_R}{J_{yy}}p \\
\dot{r} &= \frac{J_{xx} - J_{yy}}{J_{zz}}pq + \frac{1}{J_{zz}}M_r - \frac{a_R}{J_{zz}}r
\end{aligned}
\tag{13}
$$

where M_p, M_q, M_r are in the nonlinear relation with Ω_i^2.

Selecting the state vector of the target as $\mathbf{x_h^t} = [x, y, z, u, v, w, \phi, \theta, \psi, p, q, r]^T$, the motion model of the target can be expressed as

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\\ \dot{u}\\ \dot{v}\\ \dot{w}\\ \dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\\ \dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\\ \dot{x}_6\\ \dot{x}_7\\ \dot{x}_8\\ \dot{x}_9\\ \dot{x}_{10}\\ \dot{x}_{11}\\ \dot{x}_{12}\end{bmatrix} = \begin{bmatrix} c\theta c\psi\, x_4 + (s\phi s\theta c\psi - c\phi s\psi)x_5 + (c\phi s\theta c\psi + s\phi s\psi)x_6 \\ c\theta s\psi\, x_4 + (s\phi s\theta s\psi + c\theta s\psi)x_5 + (c\phi s\theta s\psi - s\phi c\psi)x_6 \\ -s\theta x_4 + s\phi c\theta x_5 + c\phi c\theta x_6 \\ x_{12}x_5 - x_{11}x_6 - \frac{a_T}{m}x_4 \\ x_{10}x_6 - x_{12}x_4 - \frac{a_T}{m}x_5 \\ x_{11}x_4 - x_{10}x_5 - \frac{a_T}{m}x_6 \\ x_{10} + s\phi t\theta x_{11} + c\phi t\theta x_{12} \\ c\phi x_{11} - s\phi x_{12} \\ \frac{s\phi}{c\theta}x_{11} + \frac{c\phi}{c\theta}x_{12} \\ \frac{J_{yy} - J_{zz}}{J_{xx}}x_{11}x_{12} - \frac{a_R}{J_{xx}}x_{10} \\ \frac{J_{zz} - J_{xx}}{J_{xx}}x_{10}x_{12} - \frac{a_R}{J_{yy}}x_{10} \\ \frac{J_{xx} - J_{yy}}{J_{zz}}x_{10}x_{11} - \frac{a_R}{J_{zz}}x_{12} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ s\theta g \\ s\phi c\theta g \\ c\phi c\theta g - \frac{1}{m}F_w \\ 0 \\ 0 \\ 0 \\ \frac{1}{J_{xx}}M_p + \frac{J_{p,zz}}{J_{xx}}x_{11}W_G \\ \frac{1}{J_{yy}}M_q - \frac{J_{p,zz}}{J_{yy}}x_{10}W_G \\ \frac{1}{J_{zz}}M_r \end{bmatrix} \tag{14}$$

Since we are not aware of the voltage input of the target, namely, we are not aware of the rotational speed of the propellers, the terms related with the rotational speed are treated as noise. Namely, F_w, W_G, M_p, M_q, M_r are noises. Assuming the inputs are Gaussian distributed, they are denoted as $\mathbf{W^t} \sim N(\mathbf{0_{5\times 1}}, \mathbf{\Sigma^t_{5\times 5}})$. From Equation (14), it can be seen that the noise and states are coupled together, and also the states are in a nonlinear relation. Thus, the motion model is expressed in the following way

$$\dot{\mathbf{x}}_\mathbf{h} = \mathbf{f}(\mathbf{x}_\mathbf{h}, \mathbf{W^t}) \tag{15}$$

The discrete motion model used for RBE is expressed as

$$\mathbf{x^t_{h,k}} = \Delta t \times \mathbf{f}(\mathbf{x^t_{h,k-1}}, \mathbf{W^t_{k-1}}) + \mathbf{x^t_{k-1}} \tag{16}$$

with a covariance as

$$\mathbf{\Sigma^t_{x_{h,k}}} = \mathbf{A_{k-1}}\mathbf{\Sigma^t_{x_{h,k-1}}}\mathbf{A^T_{k-1}} + \mathbf{\Sigma^t_{W_{k-1}}} \tag{17}$$

where

$$\mathbf{A}_{\mathbf{k\text{-}1}} = \frac{\partial f^{t}(\mathbf{x}^{\mathbf{t}}_{\mathbf{h,k\text{-}1}}, \mathbf{W}^{\mathbf{t}}_{\mathbf{k\text{-}1}})}{\partial \mathbf{x}^{\mathbf{t}}_{\mathbf{h,k\text{-}1}}}\Big|_{\mathbf{x}^{\mathbf{t}}_{\mathbf{h,k\text{-}1|}}} + \frac{\partial \mathbf{x}^{\mathbf{t}}_{\mathbf{h,k\text{-}1}}}{\partial \mathbf{x}^{\mathbf{t}}_{\mathbf{h,k\text{-}1|}}}\Big|_{\mathbf{x}^{\mathbf{t}}_{\mathbf{h,k\text{-}1}}} =$$

$$\begin{bmatrix}
1 & 0 & 0 & c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & c\theta s\psi & s\phi s\theta s\psi + c\theta s\psi & c\phi s\theta s\psi - s\phi c\psi & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -s\theta & s\phi c\theta & c\phi c\theta & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{a_T}{m}+1 & x_{12,k-1|k-1} & x_{11,k-1|k-1} & 0 & 0 & 0 & 0 & x_{6,k-1|k-1} & x_{5,k-1|k-1} \\
0 & 0 & 0 & -x_{12,k-1|k-1} & -\frac{a_T}{m}+1 & x_{10,k-1|k-1} & 0 & 0 & 0 & x_{6,k-1|k-1} & 0 & -x_{4,k-1|k-1} \\
0 & 0 & 0 & x_{11,k-1|k-1} & x_{10,k-1|k-1} & -\frac{a_T}{m}+1 & 0 & 0 & 0 & x_{5,k-1|k-1} & x_{4,k-1|k-1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & s\phi t\theta & c\phi t\theta \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & c\phi & -s\phi \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{a_R}{J_{xx}}+1 & \frac{J_{yy}-J_{zz}}{J_{xx}}x_{12,k-1|k-1} & \frac{J_{yy}-J_{zz}}{J_{xx}}x_{11,k-1|k-1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{J_{zz}-J_{xx}}{J_{xx}}x_{12,k-1|k-1}-\frac{a_R}{J_{yy}} & 1 & \frac{J_{zz}-J_{xx}}{J_{xx}}x_{10,k-1|k-1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{J_{xx}-J_{yy}}{J_{zz}}x_{11,k-1|k-1} & \frac{J_{xx}-J_{yy}}{J_{zz}}x_{10,k-1|k-1} & -\frac{a_R}{J_{zz}}+1
\end{bmatrix}$$

B.2 Dynamics model of the robot

The only difference between the dynamics model of the robot and the target is that the motor speeds of the 6 motors are known, and thus F_w, W_G, M_p, M_q, M_r are known. Let the control inputs as $\mathbf{U^r} = [F_w, W_G, M_p, M_q, M_r]$ and the wind-induced noised as $\mathbf{W^r} \sim N(\mathbf{0_{5\times 1}}, \mathbf{\Sigma^r_{5\times 5}})$. The motion model of the robot then becomes

$$\dot{\mathbf{x}}^{\mathbf{r}}_{\mathbf{h}} = \mathbf{f}(\mathbf{x}^{\mathbf{r}}_{\mathbf{h}}, \mathbf{U}^{\mathbf{r}}, \mathbf{W}^{\mathbf{r}}) \tag{18}$$

The discrete motion model is thus

$$\mathbf{x}^{\mathbf{r}}_{\mathbf{h,k}} = \Delta t \times \mathbf{f}(\mathbf{x}^{\mathbf{r}}_{\mathbf{h,k\text{-}1}}, \mathbf{U}^{\mathbf{r}}_{\mathbf{k\text{-}1}}, \mathbf{W}^{\mathbf{r}}_{\mathbf{k\text{-}1}}) + \mathbf{x}^{r}_{\mathbf{k\text{-}1}} \tag{19}$$

This can be used for the prediction of the robot.

II. II Sensor model

A. Exact sensor model

In our set up, two cameras were used to form a stereo vision system to localize the target. GPS was used to determine the location of the tracking UAV. The following equations describe the sensor model used in our simulation environment

$$^{G}Z_k^t = {}^{G}h^t(x_k^t,\ x_k^{si}) + {}^{G}v_k^t$$

$$^{G}Z_k^r = {}^{G}h^r(x_k^r,\ x_k^{si}) + {}^{G}v_k^r$$

where h is the global state determined from the prediction step. That means we denote h as $^{G}x_k^t$ for the target and as $^{G}x_k^r$ for the tracking UAV, maintaining the reference in the global frame. Variables v represent noise in the system.

B. Approximate motion model

Since the sensor model is relatively simple and does not result in high computational cost, we used the exact model instead of an approximate one for RBE.

III. RBE technique and detection techniques
III.I RBE technique

The recursive Bayesian technique that was used is the Extended Kalman filter. The selected technique is chosen in our challenge for the following reasons. It provides a precise and fast technique for nonlinear phenomena that follow the Gaussian distribution. That said we describe why both nonlinear and Gaussian best describe this challenge.

First, the nonlinear characterization we give, is for the trajectory of the target drone which is not a straight line moving at a constant velocity. More specifically, the target is performing an eight shape path, moving at a maximum of 10 m/s. This means that its trajectory cannot be linearly formulated in the global coordinate frame, which is why by using the EKF we linearize the model, while maintaining a good estimation since the path is also simple. Moreover, it needs to be said that as mentioned earlier in our models description, the motion model we use, is linear and thus the simple Kalman filter could have been used for it. We decided however, to create the platform in our code for future implementation of more complex trajectories if the testing moves to the actual competition and thus we use the EKF.

Secondly, the Gaussian distribution is commonly used to model measurement and motion model noise. We will use it to model the measurements and adjust the covariances and means accordingly in our testing process.

Please not that each matrix is a 3×3 matrix and especially but not only, the x and P are diagonal referring to the 3 dimensions.

A. EKF general formulation

Prediction step:

$$\hat{x}(k+1|k) = f(\hat{x}(k|k), u(k), k)$$

$$P(k+1|k) = F(k)P(k|k)F(k)^T + V(k)$$

$$F(k) = \frac{\partial f}{\partial x}\Big|_{x=\hat{x}(k|k)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial x_n} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Correction step:

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + Rv$$

$$P(k+1|k+1) = P(k+1|k) - RH(k+1)P(k+1|k)$$

$$v = y(k+1) - h(x(k+1|k), k+1)$$

$$R = P(k+1|k)H(k+1)^T S^{-1}$$

$$S = H(k+1)P(k+1|k)H(k+1)^T + \mathrm{W(k+1)}$$

$$H(k) = \frac{\partial h}{\partial x}\Big|_{x=\hat{x}(k+1|k)} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_n}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_1}{\partial x_n} & \cdots & \frac{\partial h_n}{\partial x_n} \end{bmatrix}_{x=\hat{x}(k+1|k)}$$

B. EKF problem specific formulation

Prediction step:

$$\hat{x}(k+1|k) = \hat{x}(k|k) + uDt$$

$$P(k+1|k) = I\,P(k|k)\,I^T + V(k)$$

$F(k) = I$, where I is the identity matrix

$$V = \begin{bmatrix} v1 & 0 & 0 \\ 0 & v2 & 0 \\ 0 & 0 & v3 \end{bmatrix}$$

Correction step:

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + Rv$$

$$P(k+1|k+1) = P(k+1|k) - RH(k+1)P(k+1|k)$$

$$v = y(k+1) - h(x(k+1|k), k+1)$$

$$R = P(k+1|k)\,I^T\,S^{-1}$$

$$S = I\,P(k+1|k)\,I^T + \mathrm{W(k+1)}$$

Since H(k) = I, the noise matrix is

$$\mathrm{W} = \begin{bmatrix} w1 & 0 & 0 \\ 0 & w2 & 0 \\ 0 & 0 & w3 \end{bmatrix}$$

The W and V matrices are defined for the robot and target separately and after testing they are found to have a good estimate of:

$$V_{robot}=\begin{bmatrix}0.2 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 2\end{bmatrix}$$

$$W_{robot}=\begin{bmatrix}0.1 & 0 & 0\\ 0 & 0.1 & 0\\ 0 & 0 & 0.1\end{bmatrix}$$

$$V_{target}=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$W_{target}=\begin{bmatrix}3 & 0 & 0\\ 0 & 3 & 0\\ 0 & 0 & 1.5\end{bmatrix}$$

Also, the $v = y(k+1) - h(x(k+1|k), k+1)$ term needs further description and **for the robot** is the diagonal matrix of the x,y,z position as measured from the GPS minus the corresponding matrix of the state prediction step. **For the target** the v term is defined as the 3D position as measured from the camera added with the GPS from the robot minus the prediction step again.

Overall v is in the form of $v=\begin{bmatrix}x & 0 & 0\\ 0 & y & 0\\ 0 & 0 & z\end{bmatrix}$.

III.II Detection technique

At this point, we should start discussing about the detection technique that was used in the process. Our code is using an opencv package to detect pixels of specific color in the image frame. The image frame is similar for right and left camera but these two camera system are able to provide us with a 3D position derivation module. The image color we choose to detect is that of the hanging ball from the target drone.

Below is the detection process in steps including also the position derivation formulations:

- Step 1: The image is converted to black and white. White stands for white color, black stands for anything else.
- Step 2: The horizontal pixel number of all white pixels is summed up and also the vertical pixel number. The average is taken for both of them and the position of this centroid is derived, as shown in Fig. 3

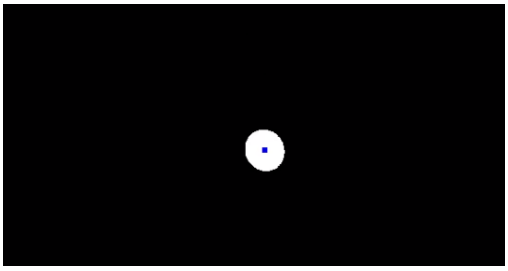


Fig. 3 Centroid in the grayscale

- Step 3: The position of the centroids of both cameras are combined to derive the measured 3D position of the target using the following derivations.

In Step 3, we used two cameras to detect and localize the target. For a single camera, it is not easy to interpret the Z value (distance between the sensor and the target), thus two normal cameras are employed. The camera parameters are shown in Table. 1.

Table. 1 Camera prarameters

Scaling factor	$\boldsymbol{\lambda}$	$\boldsymbol{Z^S}$
Focal length	f	1
Skew of pixels	τ	0
Aspect ratio	η	1920:1080 (4:3)
Principal point	$[O_x, O_y]$	[0.5, 0.5]

It is difficult to find ${}^{\{S\}}z$ with only one camera. For this reason, another camera is utilized to create a stereo vision. These two cameras are placed in parallel in fixed position and both are facing forward. Figure 4 shows the camera setup.

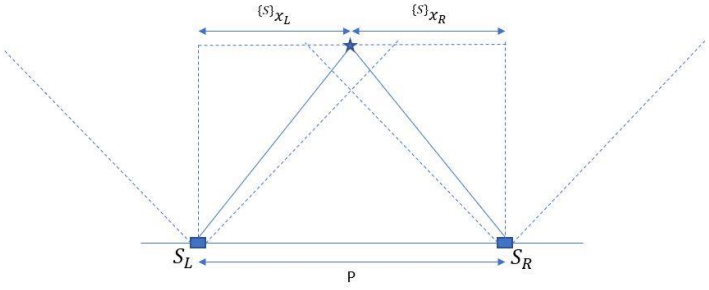


Fig. 4 Stereo Vison Setup

S_L represents the left camera; S_R represents the right camera; the target is represented by the star symbol; ${}^{\{S\}}x_L$ and ${}^{\{S\}}x_R$ are the X coordinates of the target relative to the two cameras respectively, so the summation of ${}^{\{S\}}x_L$ and ${}^{\{S\}}x_R$ will be equal to the distance between the cameras, P, which is predefined by us.

$$ {}^{\{S\}}x_L - {}^{\{S\}}x_R = b $$

The distances ${}^{\{S\}}z$ are assumed to be the same for both cameras.

Since ${}^{\{S\}}x_L - {}^{\{S\}}x_R = P$, we can determine the ${}^{\{S\}}z$ coordinate relative to the sensor platform,

$$ {}^{\{S\}}z = \frac{f * b}{{}^{\{S\}}x_L - {}^{\{S\}}x_R} $$

$$ {}^{\{S\}}y = \frac{{}^{\{S\}}x_L * z}{f} $$

$$^{\{S\}}x = \frac{^{\{S\}}y_L * z}{f}$$

- Step 4: In the calculation of the height of the object in the real world, we are also taking into account the rotation of the robot drone when its moving. As it can be seen below, while moving, the drone has an amount of inclination, as shown in Fig. 5. This inclination can be derived by the IMU in real time and the formulations used to derive the actual height in the global frame are also presented below.

Fig.5 Drone inclination while moving

In Fig. 6, the image frame is depicted in two different orientations. There are also two points on each image vertical plane, which are the bisection of the line connecting the object and centroid with each vertical plane. By defining as Vr the vertical distance of the point to the inclined center and as Dist the corresponding distance in the other frame, we can derive

$$\theta_2 = \tan^{-1}(\frac{Vr}{focal})$$

where focal is the focal length.

$$\theta_3 = \theta_1 - \theta_2$$

where θ_1 is the pitch angle

$$dist = focal * tan(\theta_3)$$

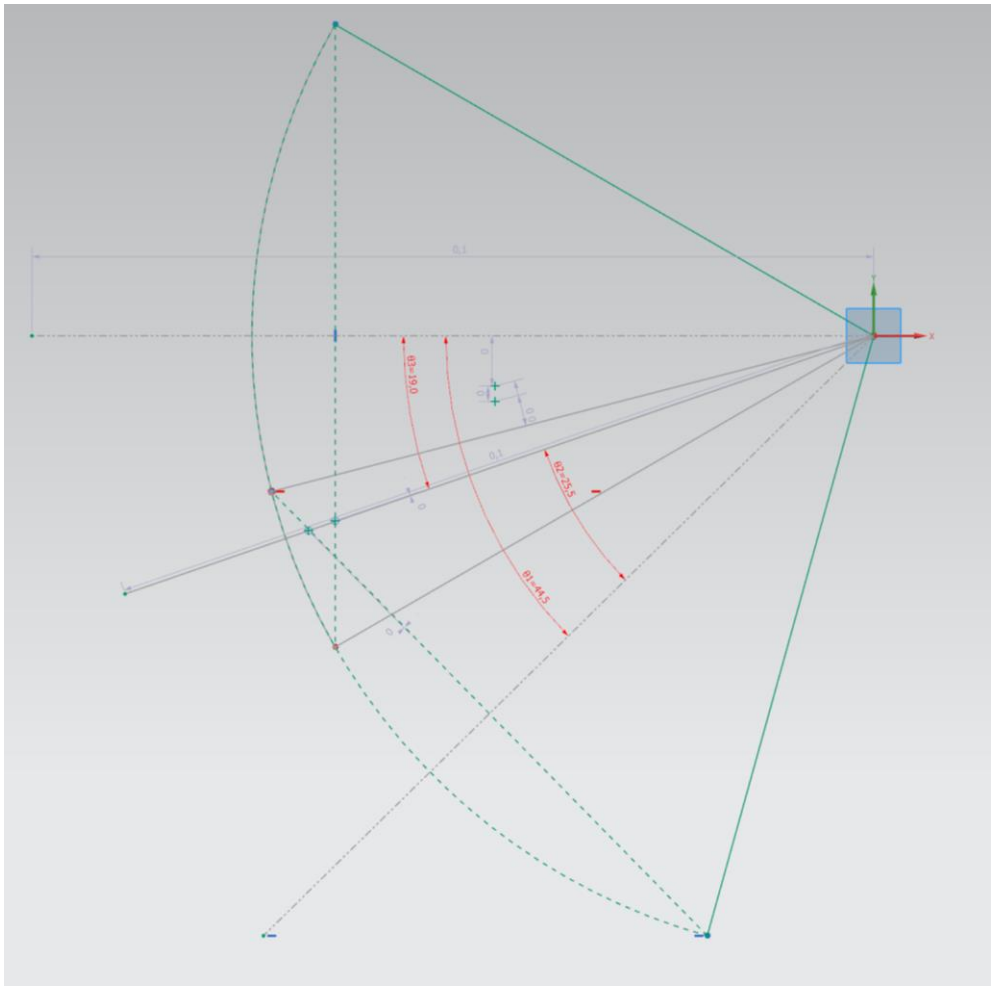


Fig.6 Geometric Analysis

IV. Estimation program architecture in ROS

Fig. 7 and Fig. 8 depict how launch files are combined in the ROS environment to fulfill the task from object detection to position estimation of robot and target.

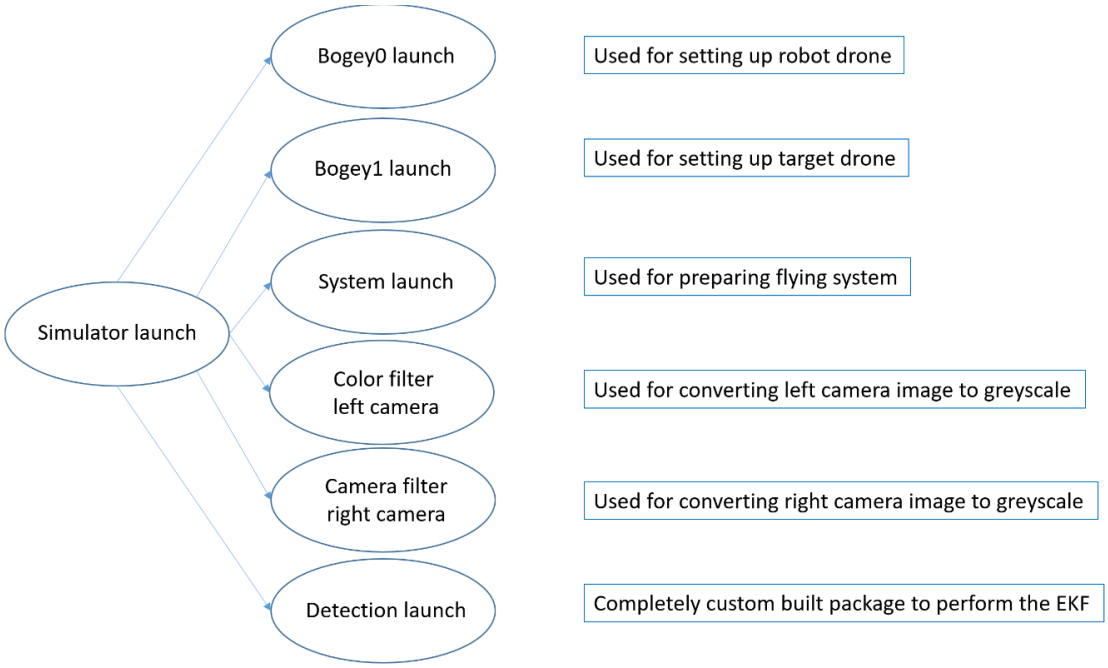


Fig. 7 System architecture

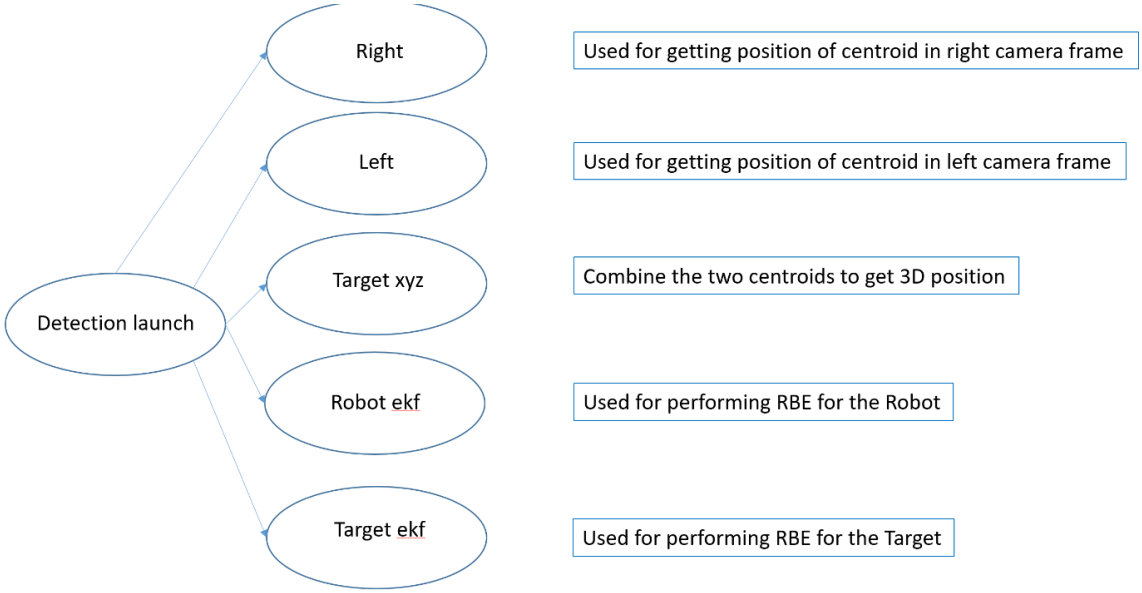


Fig. 8 Detection architecture

V. Experimental Results

In order to validate the presented approach, various scenarios were simulated in Gazebo. Our main goal was to approximate the MBZIRC challenge and check the robustness of both the code and the method.

The steps necessary to do that were to determine the parameters that characterize the problem and change the values of those parameters or give them a fixed value and then perform testing. First parameter was the velocity of the target drone, second the velocity of the chasing drone and then their corresponding flight height (z axis) and general trajectory.

The fixed parameters were the last two since we established a fixed height of 6 meters for the target drone and 7 meters for the chasing drone. Keep in mind that we are chasing the ball that is hanging from the target drone so the height for the first case is 4 meters. After that, we must also add that the trajectory is a straight line in the y axis. Therefore, the parameter we set for every different case is only the velocity in the x axis. As it will also be evident in the next graphs, y and z displacements are also there since the drones have dynamics that create also these trajectories but we do not define them somehow ourselves. These are built in by the autopilot of the drones.

Also, the errors are found by comparing the estimated values with the actual. The actual is in each case given by the GPS measurements for target and robot.

In short, the following are defined for the axes shown in Fig. 9, since our input is only velocity in the X axis.

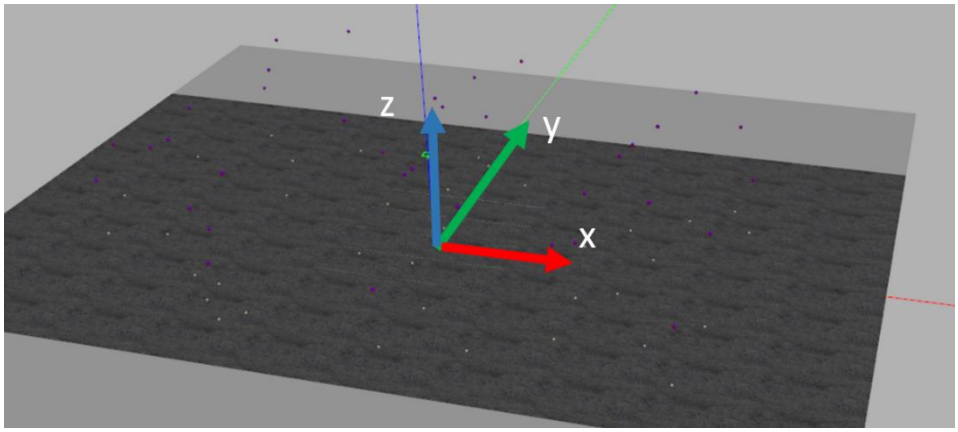


Fig. 9 Defined axis frame in Gazebo

Table 2: Starting positions of the target and robot

Parameter set	**Target drone**	**Chasing drone**
Varying	X	X
Fixed	Y= 0 m	Y= 0 m
Fixed	Z= 4 m	Z= 7 m

The different scenarios are represented below. **All scenarios have the same organization flow.**

Scenario I: (opposite direction)

- Target is moving with 1 m/s velocity
- Robot is moving with -1 m/s velocity

The results of robot estimation in Scenario I is shown in Fig 10. The graphs on the left show the predicted and corrected position in x, y, z axis. On the right we have the corrected and actual position in the three axes again.

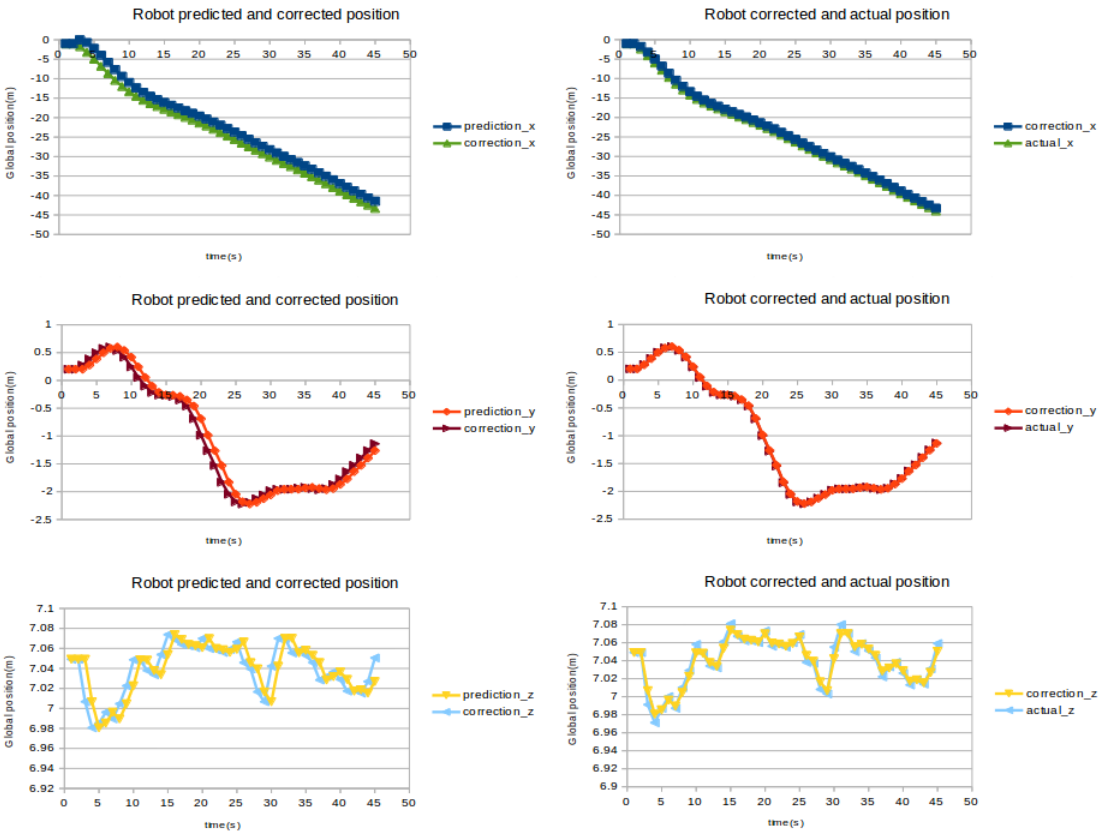


Fig. 10 Robot estimation, Scenario I

The results of target estimation in Scenario I is shown in Fig 11. The graphs are in the same structure as Fig. 10.

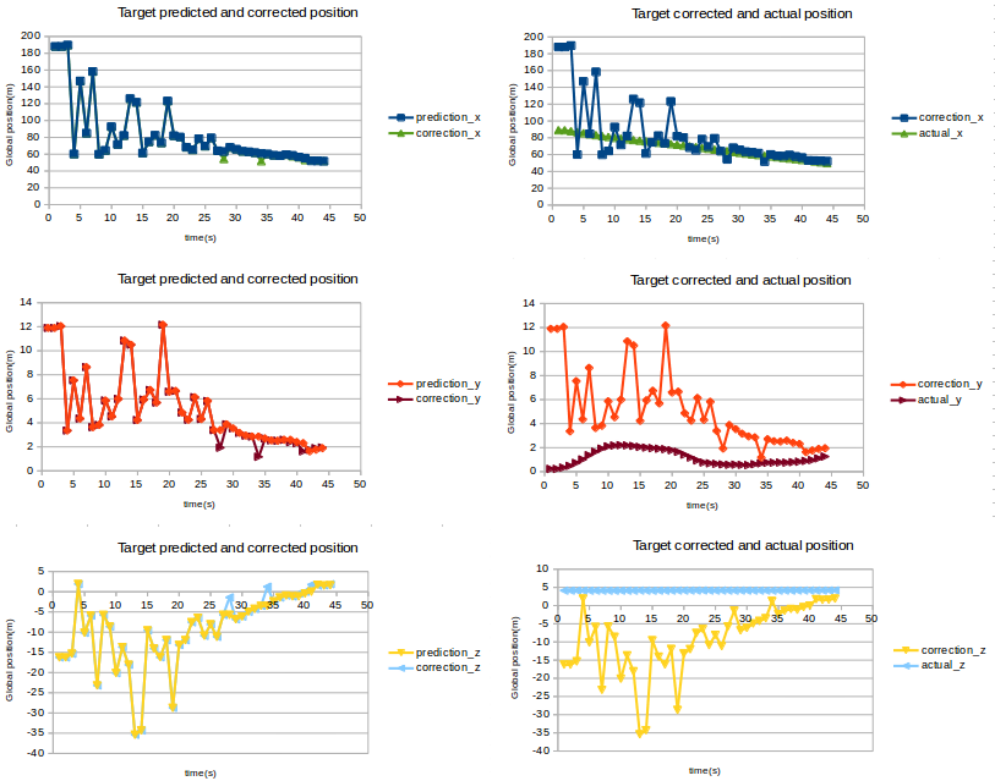


Fig.11 Target Estimation, Scenario I

Fig.12 shows the measurements of the target positions in the three axes as seen in the sensor frame.

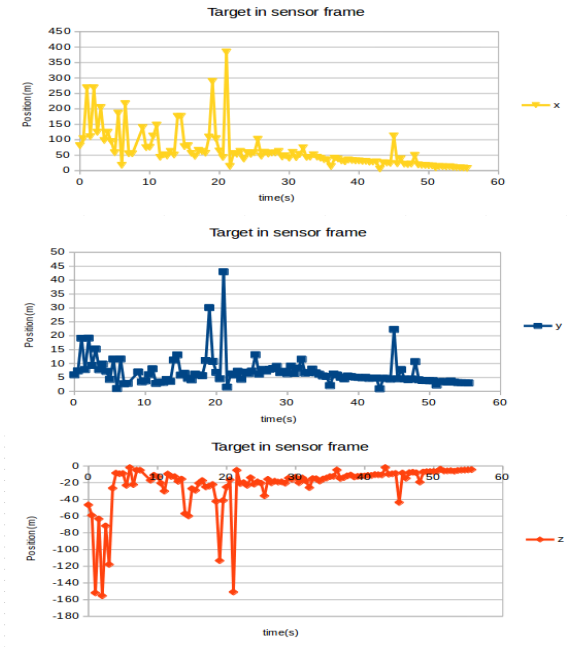


Fig. 12 Target in sensor frame, Scenario I

Scenario II: (only one is moving)

- Target not moving
- Robot is moving with -1 m/s velocity

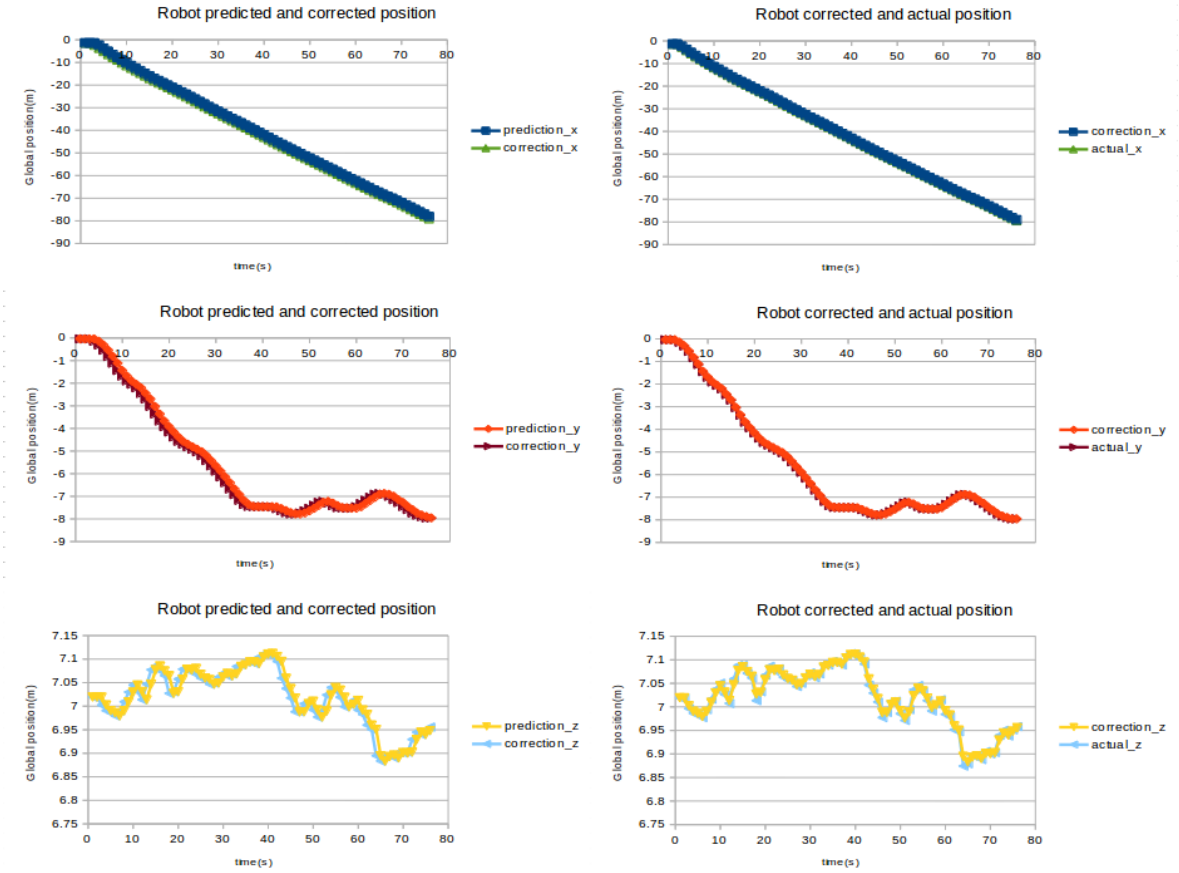


Fig. 13: Robot estimation, Scenario II

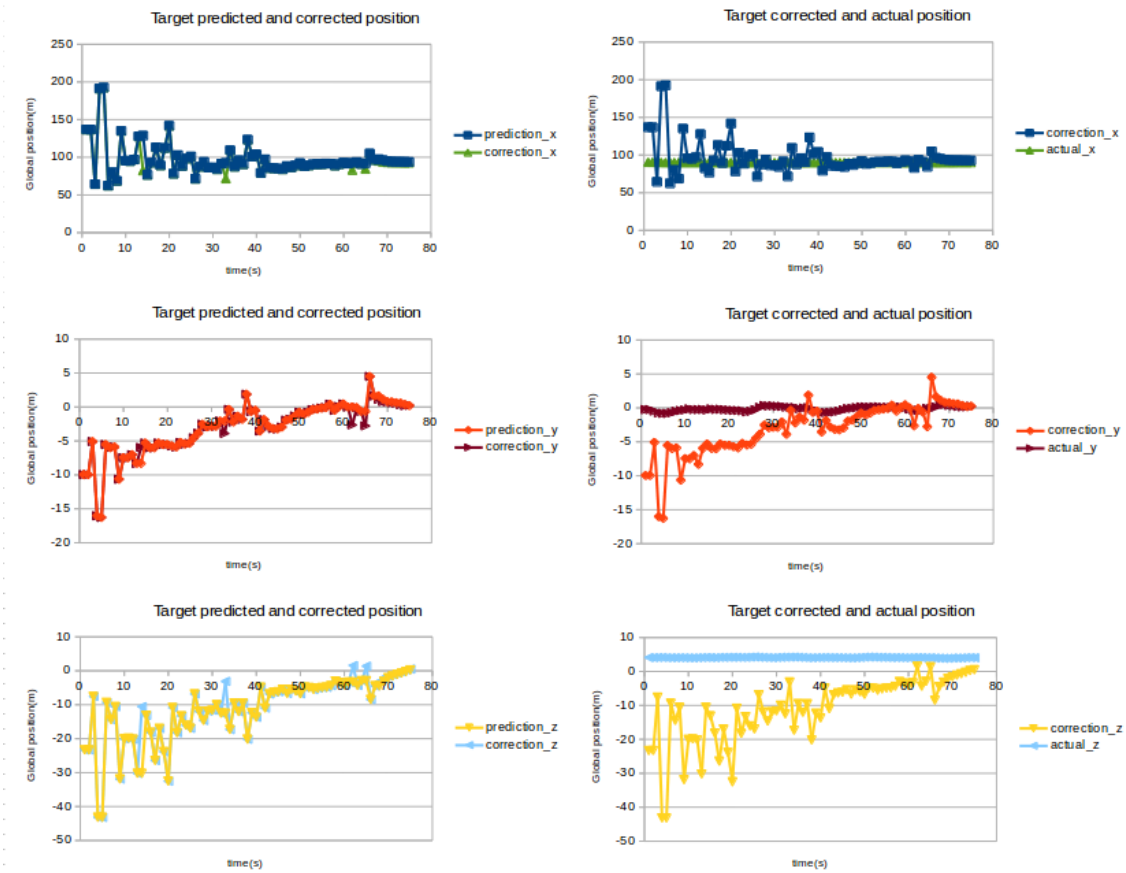


Fig. 14 Target Estimation, Scenario II

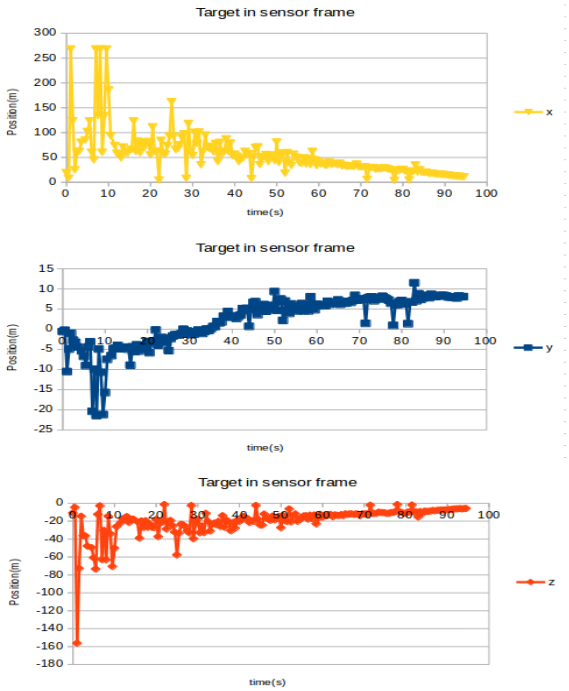


Figure 15: Target in sensor frame, Scenario II

Scenario III: (both moving in the same direction)

- Target is moving with -1 m/s velocity
- Robot is moving with -1 m/s velocity

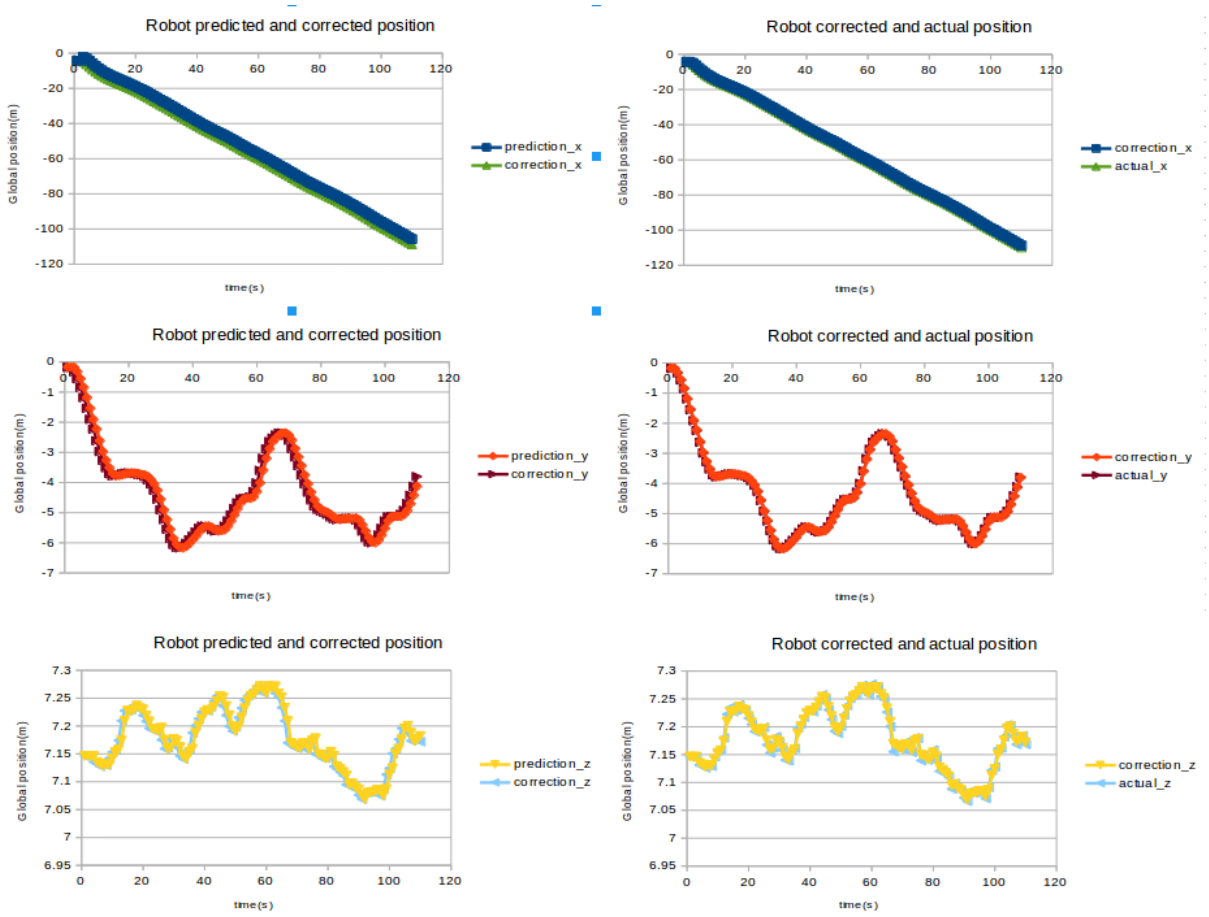


Fig.16 Robot estimation, Scenario III

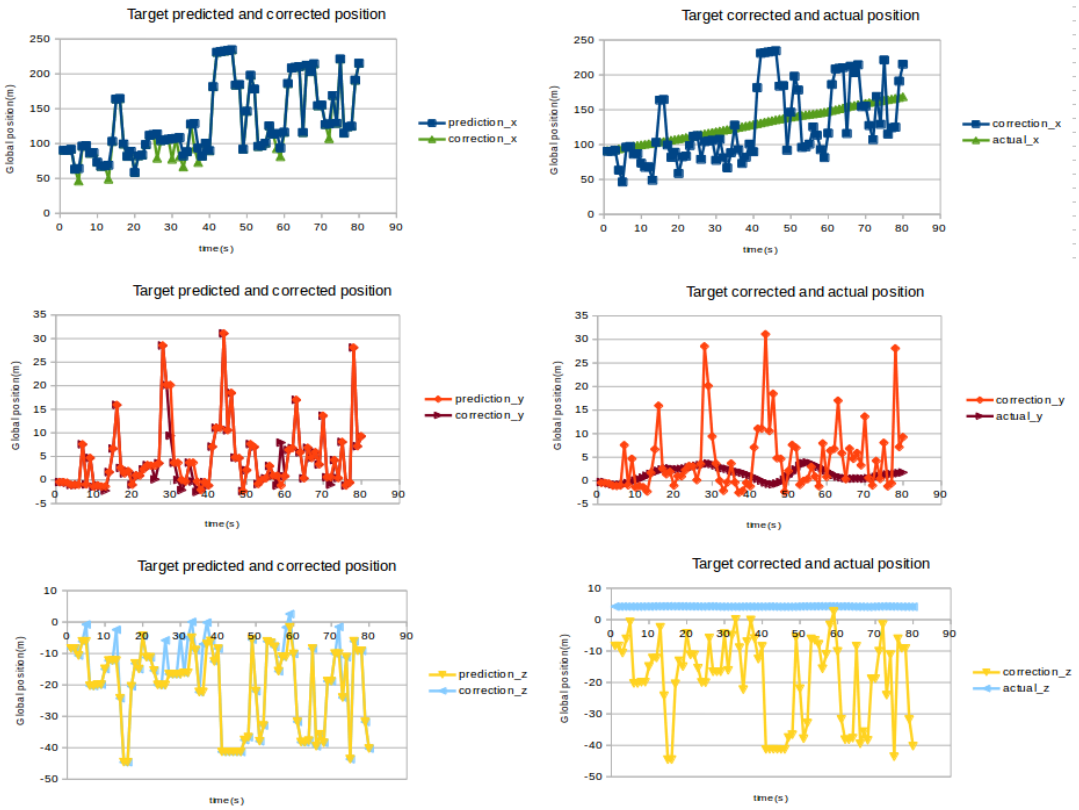


Fig. 17 Target Estimation, Scenario III

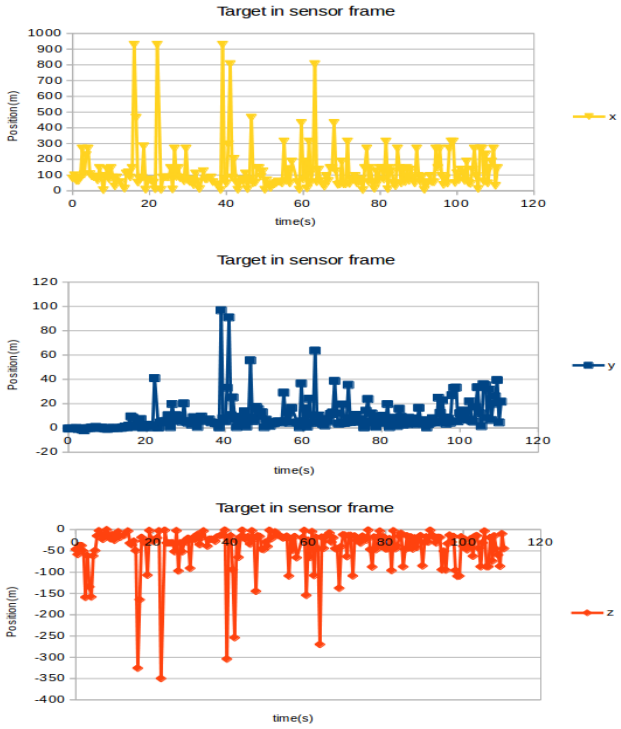


Fig. 18 Target in sensor frame, Scenario III

Scenario IV: (both moving in the same direction)

- Target is moving with -1 m/s velocity
- Robot is moving with -2 m/s velocity

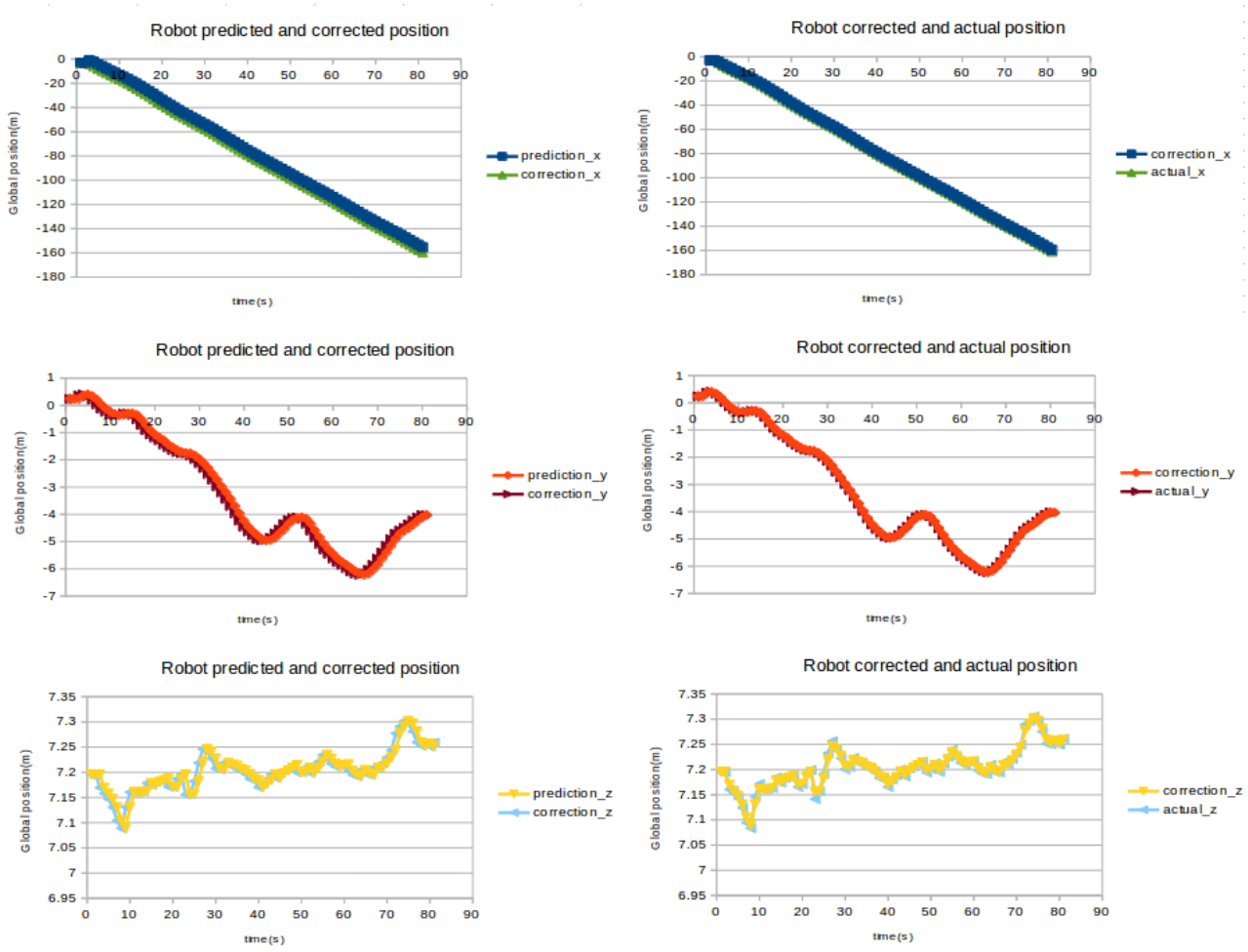


Fig. 19 Robot estimation, Scenario IV

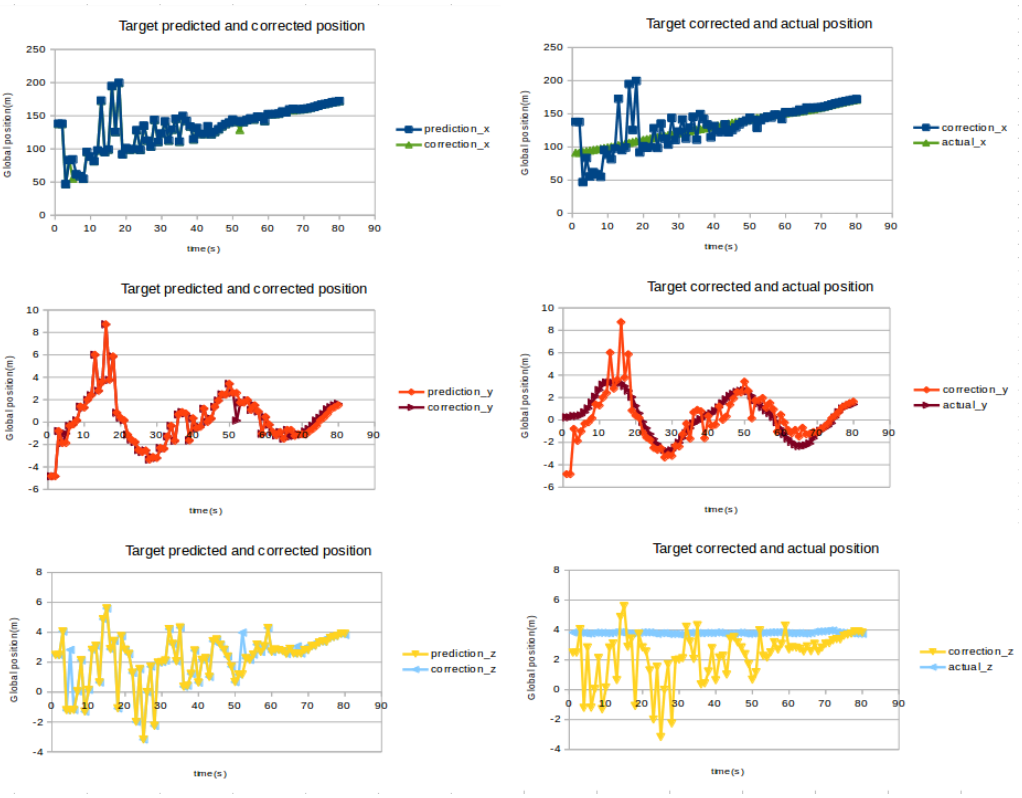


Fig. 20 Target Estimation, Scenario IV

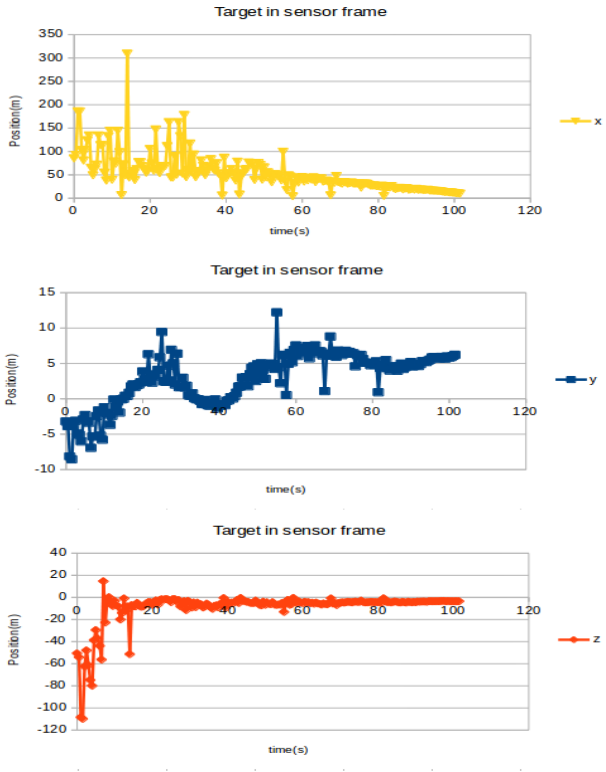


Fig.21 Target in sensor frame, Scenario IV

Scenario V: (both moving in the same direction)

- Target is moving with -1 m/s velocity
- Robot is moving with -3 m/s velocity

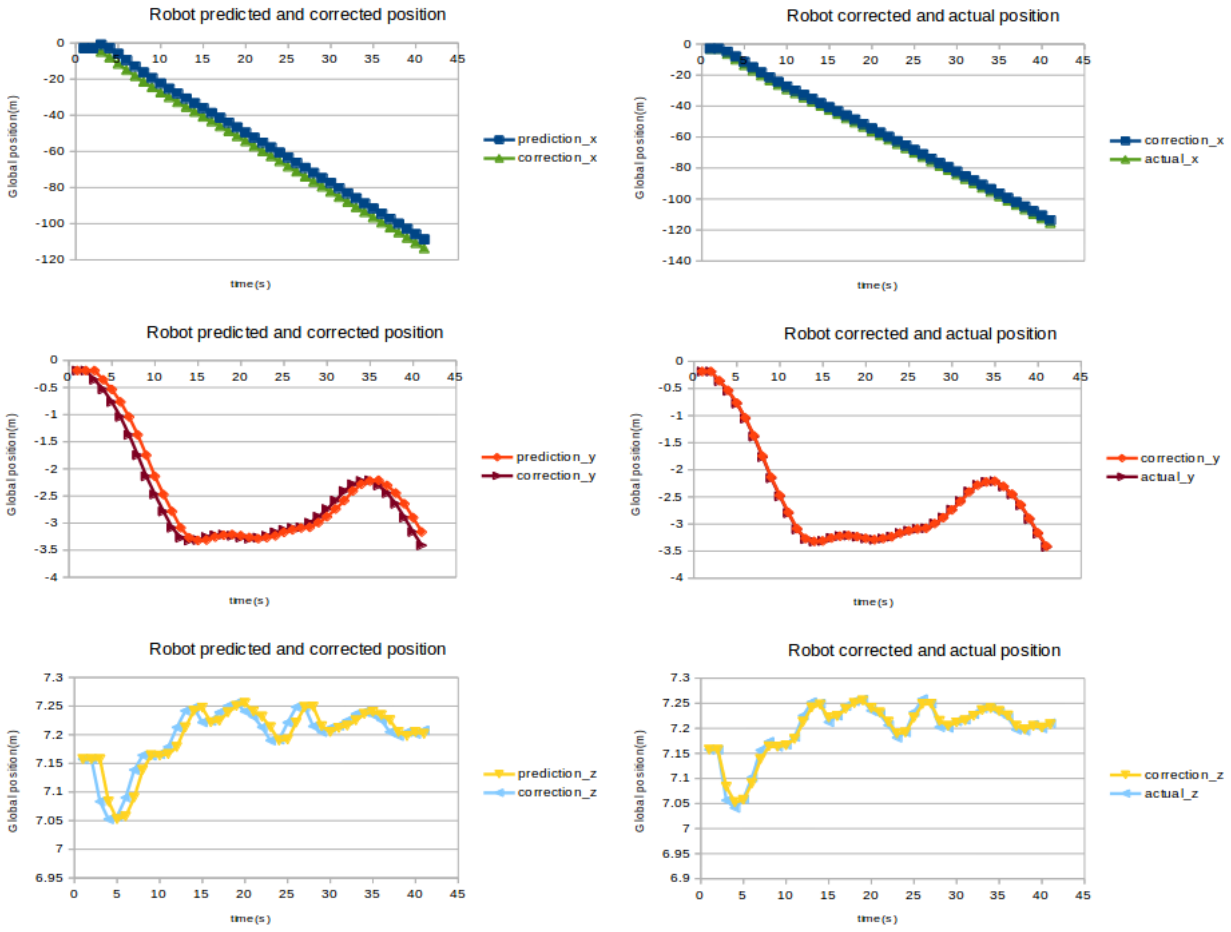


Fig.22 Robot estimation, Scenario V

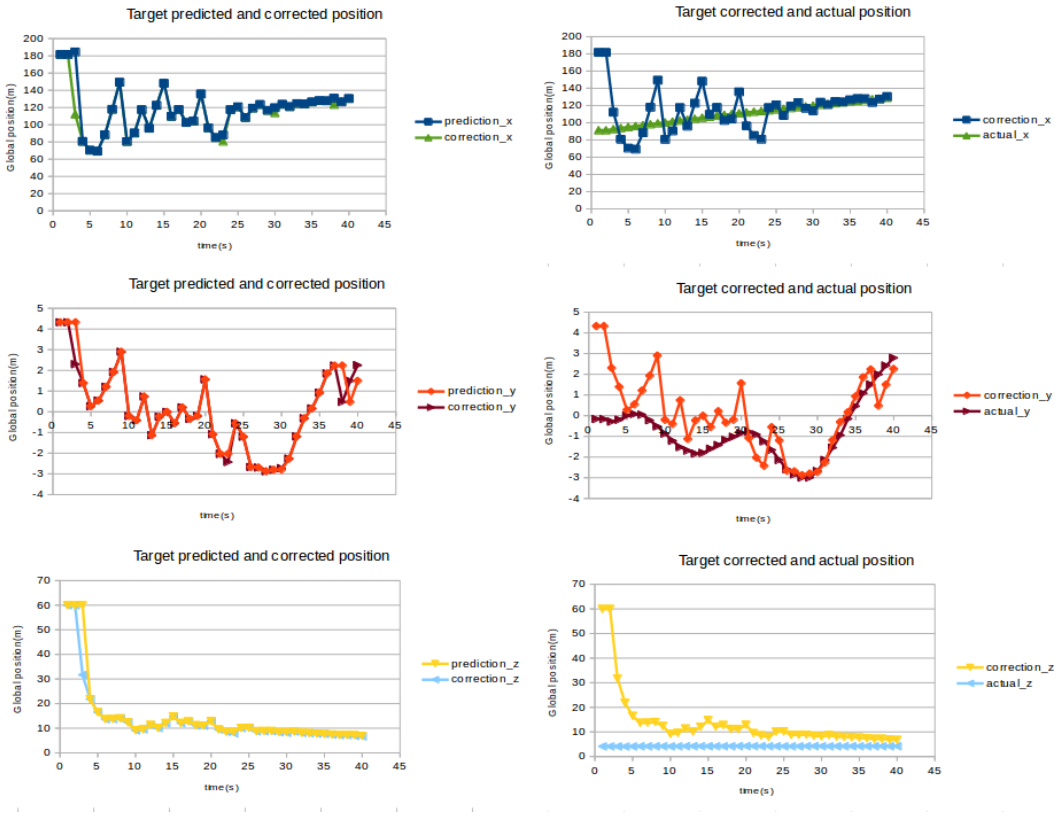


Fig. 23 Target Estimation, Scenario V

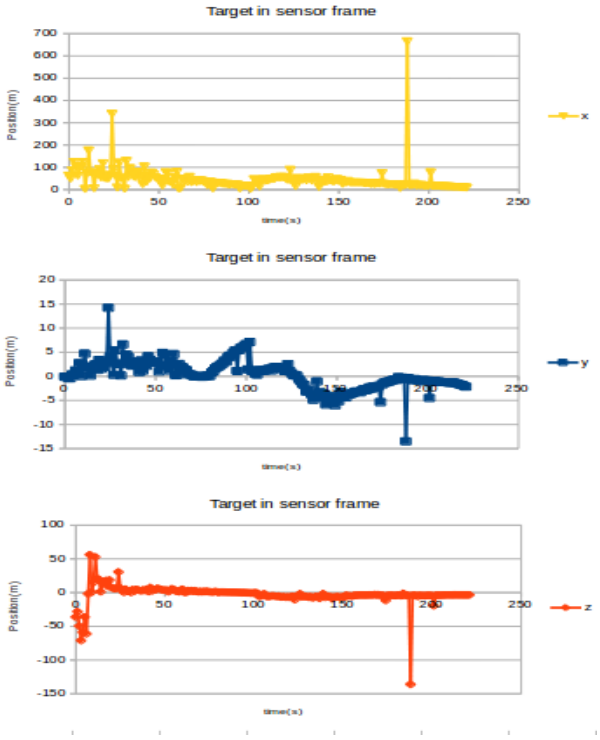


Fig. 24 Target in sensor frame, Scenario V

For viewing the efficacy of the approach in the close distance, we are presenting also the following graphs, taken from the IV scenario. By referring to the correction and actual error, we mean the difference between the correction step and the actual. By referring to the measurement and actual error, we are also referring to their difference. These are in absolute values.

It can be seen that at the steps before starting the kalman filter, these errors are the same. After initiating the RBE and considering the moving target, there are differences. In some cases, the error of the correction is smaller than the measurement which makes lots of sense.

Our conclusion is that the RBE approach we are using works efficiently and gives a small error between real and corrected value, however more work needs to be done on the estimation of the covariance in the multistage approach we are using.

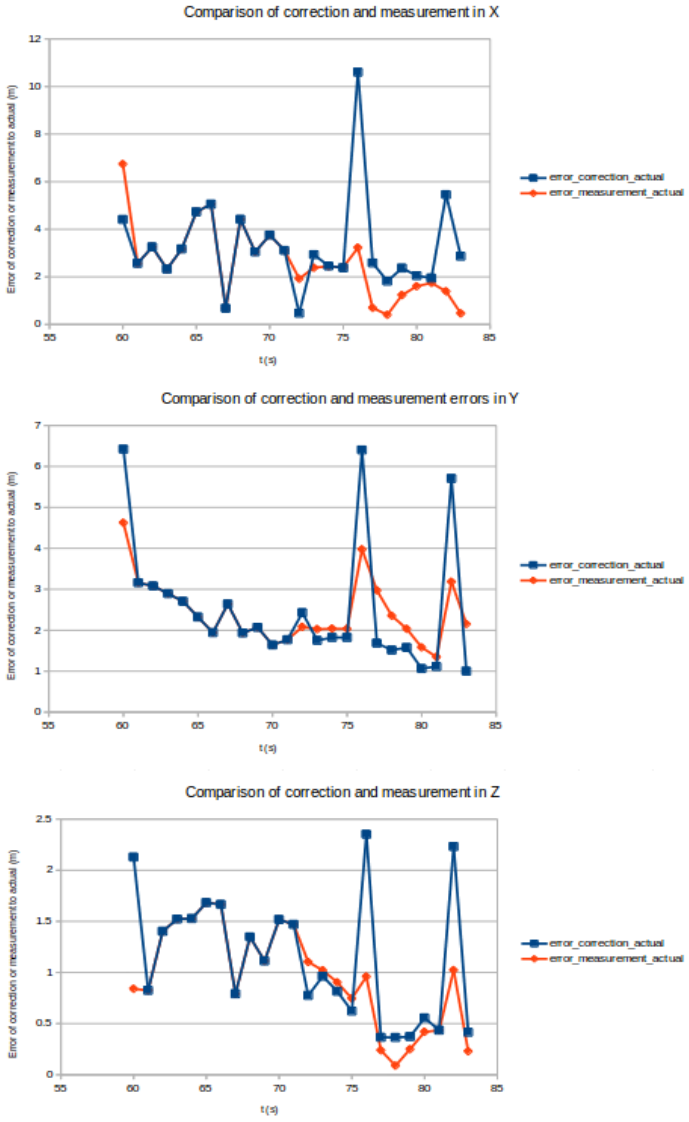


Fig. 25 Comparison of errors

VII. Conclusions and Future Work

In our work we discovered the following

- Our motion and sensors model performed well in the EKF framework.
- The two cameras system was able to adequately receive the 3D position for the target but worked better in close distances.
- The IMU and GPS provided us with precise localization and pitch angle measurements.
- Measurements were varying a lot while being far mainly due to the stereo vision system.
- Putting the Kalman filter to work when we are close produced good estimated results. We were able to estimate a covariance small when the measurements are good and that proved to work very efficiently.

Potential work in the future could include

- Perform detection of object based also on shape.
- Localize ourselves and target when we are really close to the target.
- Introduce nonlinearities in the motion model.
- Implement autonomous control for search and tracking based on observations.

Thank you for your attention!