OPTIMAL CONTROL

INTEGRATED MASTER DEGREE IN MECHANICAL ENGINEERING

SCIENTIFIC AREA OF CONTROL, AUTOMATION, AND INDUSTRIAL INFORMATICS

Target Tracking with Pan-Tilt

Author:
Luís MIRANDA, 81089
Diogo CATARINO, 81772
Reon FUJIEDA, 89120

Instructor:
Prof. Paulo OLIVEIRA

April 8, 2018

Contents

1 Introduction

Our aim is to model and control an acquisition and tracking system of a static target, in an optimal way. This system is built on top of a mobile robot "Rasteirinho" and will track the target using a combination of horizontal and vertical movement (Pan-Tilt). These movements are executed with the help of two servomotor through the acquisition of data from an end-effector camera. The main purpose of the proposed system is to keep the camera's central point fixed on the target.

In a first phase, the systems and sub-systems should be studied and modelled. After the control problem is formulated, the systems characteristics such as stability, controlability and observability will be analyzed and discussed. A Simulink model should be implemented as well. Lastly, the frequency and time responses, both in open-loop and in closed-loop, should be verified.

2 System dynamics

To solve the problem in hand the angular position values for the Pan-Tilt system must be obtained from the information obtained by the camera. Given the relative position of the target to the "rasteirinho", the desired angles to keep the target centered are easily taken as it is shown on figure 1.

$$\alpha_{pan} = \arctan(\frac{x_R}{Y_R}) \tag{1}$$

$$\alpha_{tilt} = \arctan(\frac{Z_R}{\sqrt{X_R^2 + Y_R^2}}) \tag{2}$$

Using the camera we can calculate the coordenates needed by measuring, in pixels, the distance from the extremities of a L-sized square to the center of the image. Observing how the camera works in figure 2 the equations for transforming from virtual to real coordenates are (in which f is the focal distance of the camera):

$$u = f\frac{Y_C}{X_C} \tag{3}$$

$$v = f\frac{Z_C}{X_C} \tag{4}$$

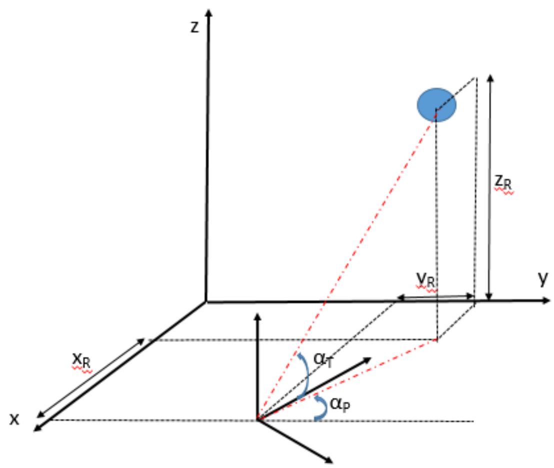

Figure 1: Target and "rasteirinho" coordinates

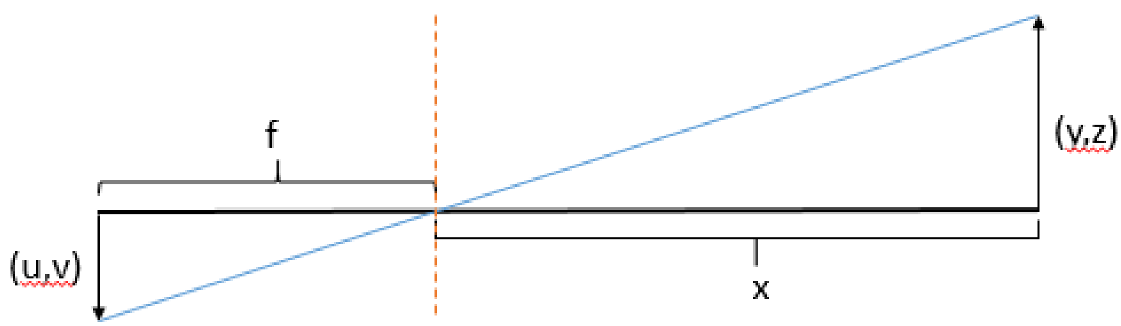

Figure 2: Camera functionality

Relating the camera reference to the "rasteirinho" reference according to the transformation seen in the Pan-Tilt Model section, the coordinates (x_C, y_C, z_C) can be calculated from the coordinates (x_R, y_R, z_R) and replacing on equations 3 and 4 the extremities of the target on the image will be given by:

$$\begin{cases} \begin{bmatrix} 0 \\ \frac{L}{2} \end{bmatrix} = \begin{bmatrix} x_R \\ z_R \end{bmatrix} + \begin{bmatrix} c_{\alpha_T} & -s_{\alpha_T} \\ s_{\alpha_T} & c_{\alpha_T} \end{bmatrix} \begin{bmatrix} x_{C_1} \\ z_{C_1} \end{bmatrix} \\ \begin{bmatrix} 0 \\ -\frac{L}{2} \end{bmatrix} = \begin{bmatrix} x_R \\ z_R \end{bmatrix} + \begin{bmatrix} c_{\alpha_T} & -s_{\alpha_T} \\ s_{\alpha_T} & c_{\alpha_T} \end{bmatrix} \begin{bmatrix} x_{C_2} \\ z_{C_2} \end{bmatrix} \end{cases} \tag{5}$$

$$\begin{cases} u_1 = f \frac{x_R s(\alpha_P - \alpha_R) + \frac{L}{2} c(\alpha_P - \alpha_R) - y_R c(\alpha_P - \alpha_R)}{-x_R c(\alpha_P - \alpha_R) + \frac{L}{2} s(\alpha_P - \alpha_R) - y_R s(\alpha_P - \alpha_R)} \\ u_2 = f \frac{x_R s(\alpha_P - \alpha_R) - \frac{L}{2} c(\alpha_P - \alpha_R) - y_R c(\alpha_P - \alpha_R)}{-x_R c(\alpha_P - \alpha_R) - \frac{L}{2} s(\alpha_P - \alpha_R) - y_R s(\alpha_P - \alpha_R)} \end{cases} \tag{6}$$

$$\begin{cases} v_1 = f \frac{x_R s_{\alpha_T} + \frac{L}{2} c_{\alpha_T} - z_R c_{\alpha_T}}{-x_R c_{\alpha_T} + \frac{L}{2} s_{\alpha_T} - z_R s_{\alpha_T}} \\ v_2 = f \frac{x_R s_{\alpha_T} - \frac{L}{2} c_{\alpha_T} - z_R c_{\alpha_T}}{-x_R c_{\alpha_T} - \frac{L}{2} s_{\alpha_T} - z_R s_{\alpha_T}} \end{cases} \tag{7}$$

At last, rearranging the previous equations, (x_R, y_R, z_R) can be calculated based on u1, u2, v1, v2 and the current angular position of the camera (α_{pan}, α_{tilt} and rotation of the "rasteririnho") which are used in equations 1 and 2 to calculate the desired angles of pan and tilt. Note that u_1 and u_2 are obtained in the same way as v_1 and v_2 only replacing α_{pan} by α_{tilt}

$$\begin{cases} x_R = \frac{(-f c_{\alpha_p} - u_2 s_{\alpha_p})(f^2 c_{\alpha_p} s_{\alpha_p} + f(c^2_{\alpha_p}(u_1+u_2) - u_1) - s_{\alpha_p} c_{\alpha_p} u_1 u_2) L}{f(f s_{\alpha_p} + u_2 c_{\alpha_p})(u_1 - u_2)} \\ y_R = \frac{-(f^2 s_{2\alpha_p} + f(2c^2_{\alpha_p} - 1)(u_1+u_2) - s_{2\alpha_p} u_1 u_2) L}{2f(u_1 - u_2)} \\ z_R = \frac{-(f^2 s_{2\alpha_t} + f(2c^2_{\alpha_t} - 1)(v_1+v_2) - s_{2\alpha_t} v_1 v_2) L}{2f(v_1 - v_2)} \end{cases} \tag{8}$$

3 Pan-Tilt Model

The camera model has 2 rotation degrees of freedom: rotation around the vertical and horizontal axis, which are called *pan* and *tilt*, correspondingly. The pan axis is collinear with the camera Z-axis, for a zero tilt angle. The tilt axis is collinear with the Y-axis, which is parallel to the ground.

By considering α_{pan} as the pan angle and α_{tilt} as the tilt angle, we can write the tilt transformation matrix as the following,

$$T_{tilt}(\alpha_{tilt}) = Rot_Y(\alpha_{tilt}) = \begin{bmatrix} cos(\alpha_{tilt}) & 0 & sin(\alpha_{tilt}) \\ 0 & 1 & 0 \\ -sin(\alpha_{tilt}) & 0 & cos(\alpha_{tilt}) \end{bmatrix} \tag{9}$$

The pan movement is the rotation about the Z-axis, for zero tilt angle. If the tilt angle is not zero, then the rotation about the vertical axis become composed by three rotations. The first one is the rotation of -α_{tilt} about the Y-axis aligning the Z-axis with the vertical axis. Then, the camera rotates with α_{pan} about the Z-axis. At last, a rotation of α_{tilt} about the Y-axis is made in order to return the camera to its tilt position. The representation of this pan transformation matrix is the following,

$$T_{pan}(\alpha_{pan}) = Rot_Y(\alpha_{tilt}) \cdot Rot_Z(\alpha_{pan}) \cdot Rot_Y(-\alpha_{tilt}) =$$

$$\begin{bmatrix} cos(\alpha_{tilt}) & 0 & sin(\alpha_{tilt}) \\ 0 & 1 & 0 \\ -sin(\alpha_{tilt}) & 0 & cos(\alpha_{tilt}) \end{bmatrix} \begin{bmatrix} cos(\alpha_{pan}) & -sin(\alpha_{pan}) & 0 \\ sin(\alpha_{pan}) & cos(\alpha_{pan}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\alpha_{tilt}) & 0 & -sin(\alpha_{tilt}) \\ 0 & 1 & 0 \\ sin(\alpha_{tilt}) & 0 & cos(\alpha_{tilt}) \end{bmatrix} \tag{10}$$

The camera pose can be described by the pair of angles α_{pan} and α_{tilt} with respect to the initial position. Due to the relative movement, the world seen by the camera moves in the opposite. Therefor, the transformation matrix from the initial camera coordinates to the new ones is $T_{world}(\alpha_{tilt}; \alpha_{pan}) = Rot_Y(-\alpha_{tilt}) \cdot Rot_Z(-\alpha_{pan})$

$$T_{world}(\alpha_{tilt}; \alpha_{pan}) =$$

$$\begin{bmatrix} cos(\alpha_{tilt})cos(\alpha_{pan}) & cos(\alpha_{tilt})sin(\alpha_{pan}) & -sin(\alpha_{tilt}) \\ -sin(\alpha_{pan}) & cos(\alpha_{pan}) & 0 \\ sin(\alpha_{tilt})cos(\alpha_{pan}) & sin(\alpha_{tilt})sin(\alpha_{pan}) & cos(\alpha_{tilt}) \end{bmatrix} \tag{11}$$

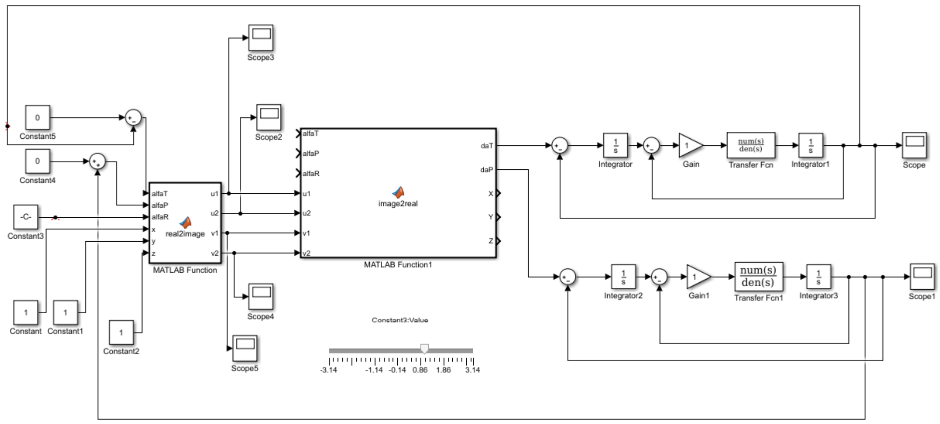

Figure 3: "Rasteirinho" and camera

4 Simulink

From the variables $\alpha_{tilt}, \alpha_{pan}, \alpha_R$ and the spacial coordinates of rasteirinho it was possible to create a Simulink file to model the rasteirinho, as can be seen on figure 3.

- The first MATLAB box, real2image, contains equations (3) and (4): given the five variables above mentioned, it is possible to obtain the coordinates u1, u2, v1 and v2 from the target.
- The second MATLAB box, image2real, contains equation (8): from the image position (u,v) calculated before, it is possible to obtain the rasteirinho position (x_R, y_R, z_R) and an update at angles α_{tilt} and α_{pan}

Concerning the servo motors, another Simulink was created, as can be seen on figure 4, whose inputs are the angles calculated above.

A step input was simulated, and the servo response and servo response error can be seen on figures 5 and 6.

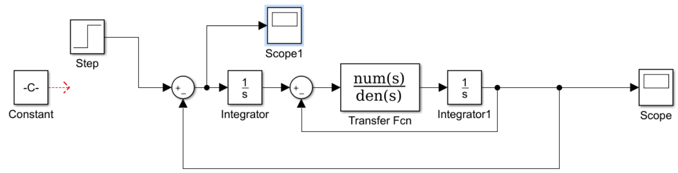

Figure 4: Servo

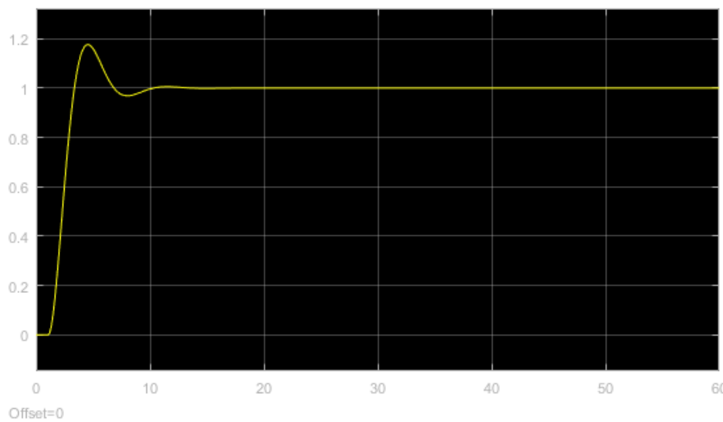

Figure 5: Servo response

Figure 6: Servo output error

5 System Characteristics

Based on previous papers on this subject, a block diagram for the servo motors was drawn as well as its transfer function.

$$G(s) = \frac{657.92}{s^3 + 40s^2 + 657.92s} \tag{12}$$

Given the transfer function it is possible to represent the system in state space. To do that it is known that:

$$\frac{Y(s)}{G(s)} = \frac{b_0 s^n + b_1 s^{n-1} + b_2 s^{n-2} + ... + b_{n-1}s + b_n}{a_0 s^n + a_1 s^{n-1} + a_2 s^{n-2} + ... + a_{n-1}s + a_n} \tag{13}$$

Calculating the a and b polynomials we can then have a state space representation of the system (in the canonical controllable form):

$$\begin{cases} a_0 = 1 \\ a_1 = 40 \\ a_2 = 657.92 \\ a_3 = 0 \end{cases} \tag{14}$$

$$\begin{cases} b_0 = 0 \\ b_1 = 0 \\ b_2 = 0 \\ b_3 = 657.92 \end{cases} \tag{15}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -657.92 & -40 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} ; y = \begin{bmatrix} 657.92 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{16}$$

The system can now be checked for controlability and observability by calculating their respective matrixes.

$$C_s = \begin{bmatrix} B & \vdots & AB & \vdots & \dots & \vdots & A^{n-1}B \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -40 \\ 1 & -40 & 942.08 \end{bmatrix} \tag{17}$$

$$O_s = \begin{bmatrix} C \\ \dots \\ CA \\ \dots \\ \vdots \\ \dots \\ CA^{n-1} \end{bmatrix} = \begin{bmatrix} 657.92 & 0 & 0 \\ 0 & 657.92 & 0 \\ 0 & 0 & 657.92 \end{bmatrix} \tag{18}$$

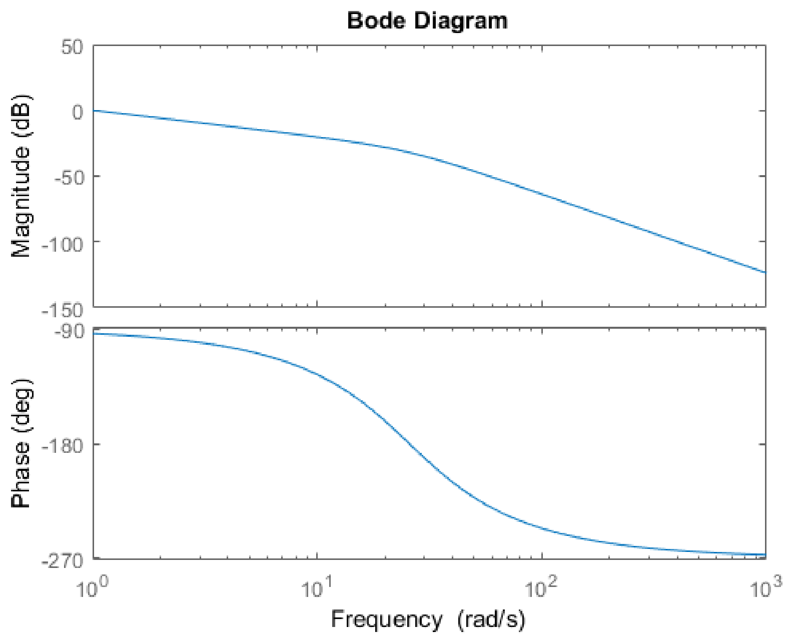

Figure 7: Open loop bode diagram

Both these matrixes are full rank (rank = 3) which means that the system is completely controlable and observable.
The systems time and frequency responses in both open loop and closed-loop can be analyzed to gather more information on the system.

As expected the step response of the system in open loop is unstable given the presence of an integrator (pole at s=0).
The system in open loop is marginally stable having 2 conjugated complex poles and another one in the origin. In closed loop (with unitary feedback) it is stable and we can see that has no stationary error because of the integrator. Parameters such as gain and phase margins and rise time can be improved according to the control methods implemented.

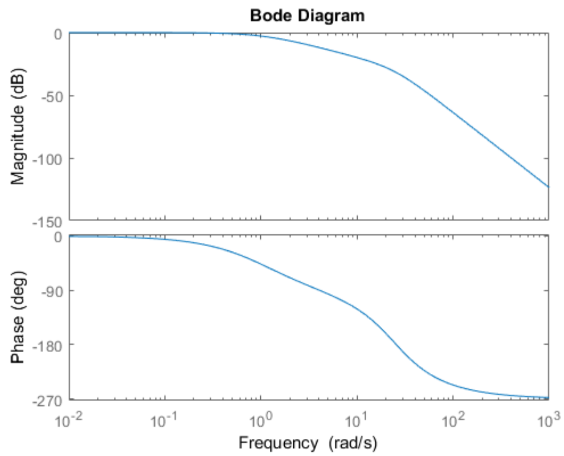

Figure 8: Closed loop bode diagram

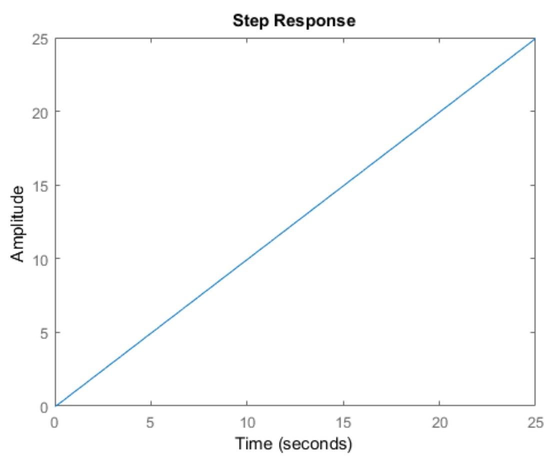

Figure 9: Open loop step response

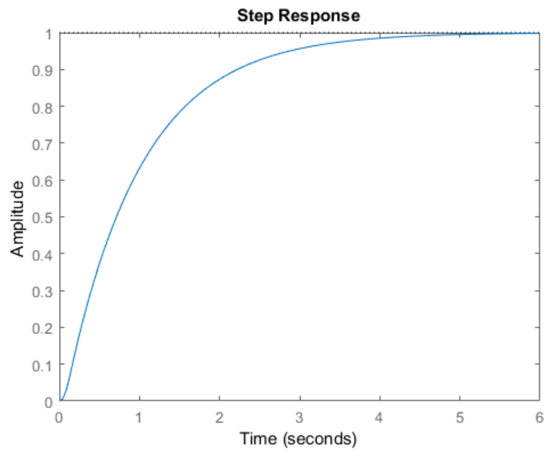

Figure 10: Closed loop step response

6 Conclusion

The explained model and discussion above about the system characteristics results in the projected system. As expected the system is linear, observable, controllable and stable in closed-loop. In the next phase, the control goals of the system to be developed need to be identified and a classical control solution need to be proposed. After that, the solution needs to be checked to see if it works and if offers a desired performance under realistic disturbances.

References

[1] Ogata, K., *Modern Control Engineering*. Prentice Hall, 5th edition, 2010.

[2] Brogan, W. L., *Modern Control Theory*. Prentice Hall, 3rd edition, 1990.

[3] Daniel, D., Armando, A., and Ramos, H., *Target Tracking with Pan-Tilt*. Instituto Superior Técnico, 2015.

[4] Park, J., Hwang, W., Bahn, W., Lee, C., Kim, T., Shaikh, M., Kim, K. and Cho, D., *Pan/Tilt Camera Control for Vision Tracking System Based on the Robot Motion and Vision Information*. IFAC 18th World Congress, 2011.

[5] Nasiri, N., *Camera-based 3D Object Tracking and Following Mobile Robot*. Swinburne University of Technology, 2006.