Turkish Journal of Electrical Engineering and Computer Sciences

Volume 24 | Number 4 Article 29

1-1-2016

Investigation of an object follower system

ELİF ERZAN TOPÇU

AHMET DEMİRKESEN

İBRAHİM YÜKSEL

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 2412 – 2428

doi:10.3906/elk-1402-48

Investigation of an object follower system

Elif ERZAN TOPÇU[1], Ahmet DEMİRKESEN[2], İbrahim YÜKSEL[1,*]
[1]Department of Mechanical Engineering, Faculty of Engineering, Uludağ University, Bursa, Turkey
[2]Graduate School of Natural and Advanced Sciences, Uludağ University, Bursa, Turkey

Received: 05.02.2014 • **Accepted/Published Online:** 13.08.2014 • **Final Version:** 15.04.2016

Abstract: In this work, an example of a mechatronic system that involves all elements of model-based design, called an object follower system, has been developed. This system is a servo system that is able to recognize within its field of view an object that has a certain color and space, and it is able to control the camera's point of view to center the object through the view itself. With the help of the model-based design to control the system design embedded algorithms have been developed for this system and used in a simulation environment and experimental space. Various PID control techniques are applied to the system to control the speed and position of the movement subsystem. The performance of the object follower system has been tested.

Key words: Model-based design, xPC Target, object follower system, position control, rapid control prototyping

1. Introduction

Techniques oriented toward perception and trailing of objects in vision are used in many different areas. Visual systems are used in many areas involving computer visioning, control theory, mechanical engineering, and optical and industrial automation that work together. Examples can be given as industrial robotics, navigation systems in auto devices, perception of visual observation, and automatic examination of production lines.

When we look at the works that have been done in last few years, Chaumette and Hutchinson [1] examined the systems used in the servo control of image processing of robots, or, in other words, visual servo control. Lee and Wohn [2] developed a work about trailing moving objects by a mobile camera. Kim et al. [3] proposed a theory on perception of movement and trailing of the movement and they actualized this theory with a mobile robot that uses a camera. Mir-Nasiri [4] developed a mobile trailing robot that trails objects in 3D. Lang et al. [5] worked on a mobile robot focused on vision that defines objects and trails them. Kim et al. [6] proposed a real-time solution for modeling and tracking multiple 3D objects in unknown environments for augmented reality. Mohammed et al. [7] investigated a trajectory tracking control system theoretically. They suggested an optimal preview control method to track a trajectory in a 3-axis servo table system. Pomares et al. [8] worked on a 4-axis robot tool based on a joint structure to perform complex machining shapes in some noncontact processes. A new dynamic visual controller was proposed in order to control this structure. Nasisi and Carell [9] developed an adaptive controller for robot positioning and tracking using direct visual feedback with a camera-in-hand configuration. Cubber et al. [10] presented a model-driven approach to derive a description of the motion of a target object. This developed technique can be applied to any pan-tilt zoom camera mounted on a mobile vehicle as well as to a static camera. Huang and Lin [11] constructed a low-cost PC-based control

*Correspondence: ibrahim@uludag.edu.tr

$X-Y$ table with AC servo motors and ball screw mechanisms to achieve long-range micropositioning. This $X-Y$ mechatronics system was integrated with a machine vision system to construct a visual-based motion control system for industrial positioning applications. Jaradat et al. [12] developed a new strategy for automating three-dimensional miniaturized manipulators using visual feedback. This automation scheme can be used to handle the challenges associated with manipulation of soft components. The developed vision control system combines depth and planar motion control using a single camera and an autofocus algorithm.

Works based on model-based methods are also examined. Orehek and Robl [13] designed a truck hydrostatic front wheel driver controller in a model-based design. Kirby and Kang [14] developed a work on safeguard relays of power systems, introduced an innovative work technique on a model-based design, and improved the design cycle. Other studies [15,16] examined a model-based design method and object follower system behavior by using different control algorithms.

In this work, working with the model-based design approach, which is a mathematical and visual method to overcome the design of embedded software and complex control systems, is discussed in the design and implementation stages of mechatronic systems. As an example of the application an object tracking system is investigated.

This object follower system is a servo system that is able to recognize an object that has a certain color and dimensions through its visionary field and is able to control the angular movements of the camera to center the object in the vision. PID control techniques are applied to the system to control the speed and position of the movement subsystem. As a test method, rapid control prototyping and software in loop simulations are used. The performance of the system has been tested.

In this study, it was shown that simulation studies validated the design before working with actual physical prototypes. The compliance to the requirements of the modeled system's behavior could be tested through simulations in the early stages of the design, automatic code conversion coding errors could be canceled, and rapid control prototyping with different design ideas could be quickly tested on real hardware.

2. Model-based design

Model-based designing is a method for overcoming difficulties in the design of complex control systems and embedded software. In this method a mathematical model of the system can be developed easily in a graphical block diagram environment (such as MATLAB/Simulink). Figure 1 shows a generalized workflow of a model-based design. In the model-based design, there is a system model, which is the center of the main subject, instead of physical prototypes and executable text commands.

Control engineers who work with traditional methods can get a working prototype of the product at the end of the development stage. However, they can only realize whether the prototype works or not after the system is integrated with its components. In the stages of design, realization, and integration of the system, feasible errors can only be perceived in the last stage. The main advantage of the model-based method is to facilitate tests and verifications at all stages of design. The seamless tests and verifications facilitate testing and verification of errors in early design stages.

As shown in Figure 1, in the model-based design it is possible to create codes automatically from the block diagrams representing the mathematical model of the system. The created codes are ready to be deployed to a real-time target computer, microcontrollers, DSP, and FPGA and to be tested in real time. It is possible to classify rapid prototyping, testing a processor in loop, testing software in loop, and testing hardware in loop as commonly known real-time applications.

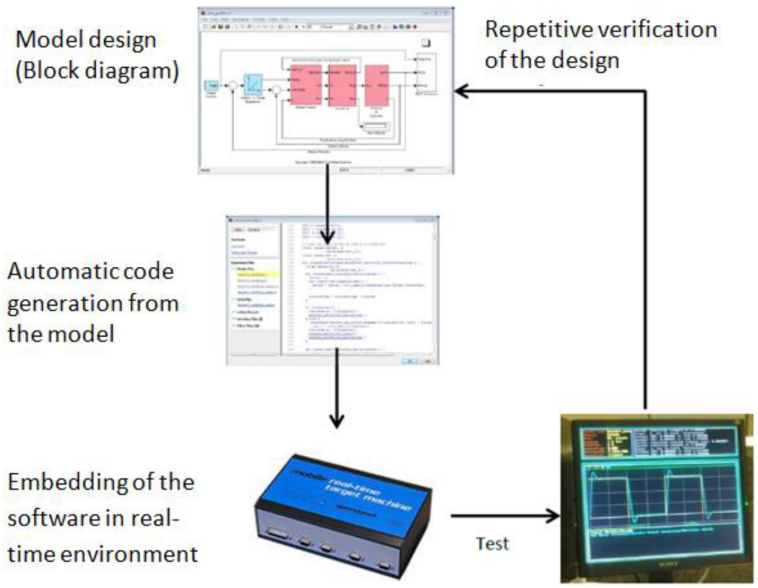


Figure 1. Work flow of the model-based design.

A real-time test is the operation of a simulation model running under normal working conditions, testing, and verification. The control algorithms are designed and analyzed related to the controlled system. The control algorithms must be integrated and tested on the real control system.

3. Structure of the object follower system

The object follower system is a moving camera that can make rotational motions in the horizontal plane and can follow a still or moving object. It is visual servo system that can update the horizontal position of the camera in such a way that it equalizes the center of the object with the center of the image. Hardware and software components related to the experimental setup are shown in Figure 2 and a photograph showing the general view of the experimental setup is given in Figure 3.

As can be seen in Figures 2 and 3, the moving camera consists of a DC motor/gear set and a camera. The DC motor is driven by a full H-bridge circuit.

Algorithms of image processing and position control run in real time on a target computer, which has Intel Core 2 Duo microprocessors and 2 GB RAM. This computer has a multifunction Humusoft I/O card to communicate with environmental components such as sensors and actuators. The position of the camera is sensed with an encoder, which is integrated into the DC motor. The speed of the camera is estimated from the position values. The HP HD-3100 web camera is chosen for imaging purposes.

MATLAB/Simulink and necessary toolboxes are used in design of the controller, development of a real-time program, and analysis. The real-time operation principle is realized by a real-time program that runs on the target computer. The real-time program was designed in the MATLAB/Simulink environment and realized with the xPC Target program. The xPC Target program was used as a real-time software program environment for the model developed in the Simulink environment. The Simulink model of the real-time program is given in Figure 4.

There is a configuration of target–host computer communications for an xPC Target application. The general view of the configuration is shown in Figure 5.

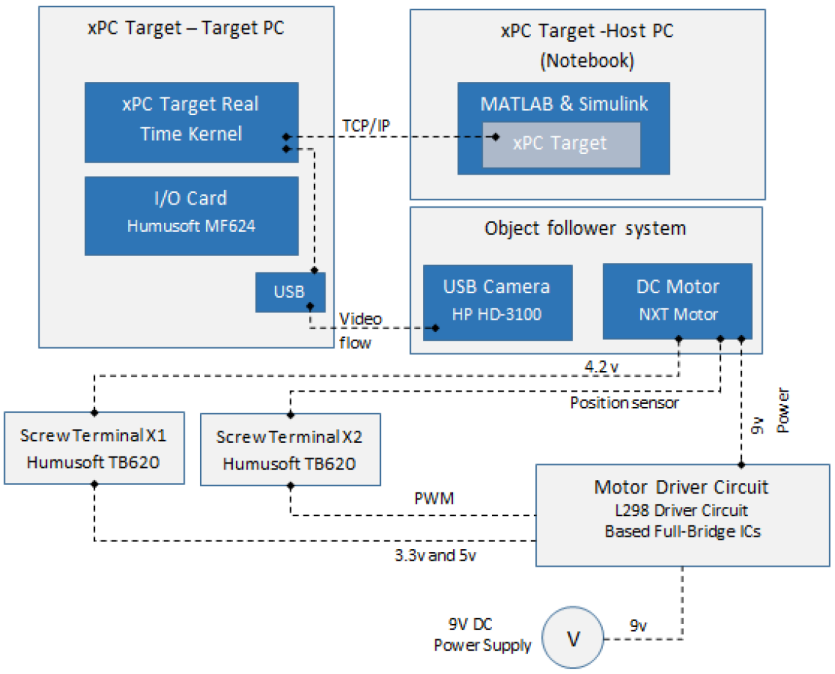


Figure 2. Hardware and software components related to the test bench.

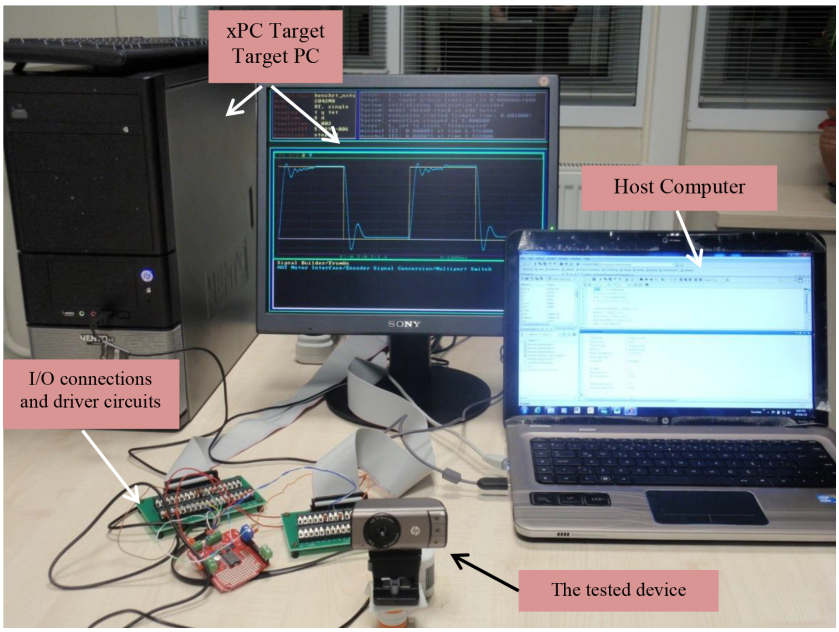


Figure 3. A general view of the experimental system and its components.

xPC Target configuration requires a target computer where the real-time applications run. The target computer operates with a real-time operation kernel and does not need any operation system such as Windows, Linux, or DOS. The real-time applications built in Simulink models run on the kernel. The target computer is controlled by the host computer. The host computer is used for development of Simulink models. The real-time applications are developed on the host computer and then downloaded to the target computer. Before and

during the run, online parameter tunings of the real-time applications are possible. The signals taken by the target computer can be transferred to the host computer. Communication of host–target computers can be done by TCP/IP protocol through a LAN.

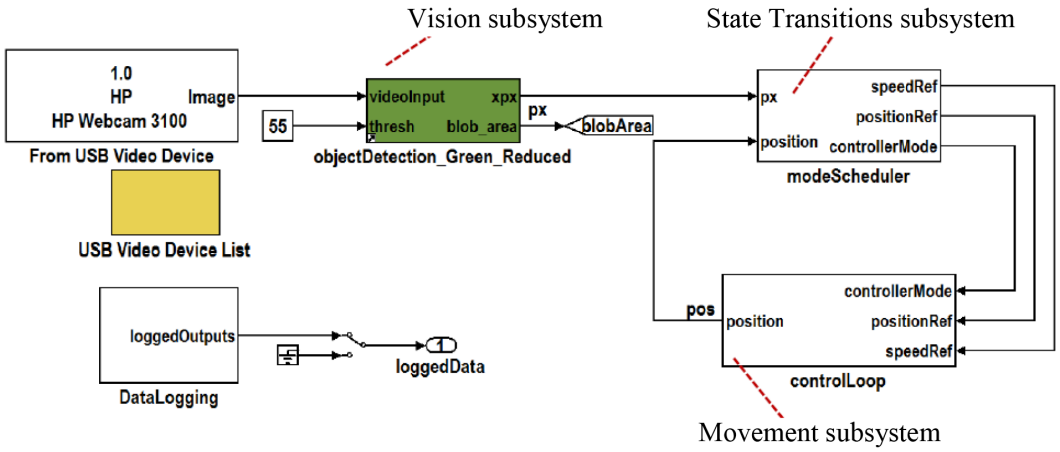


Figure 4. MATLAB/Simulink model of the real-time program.

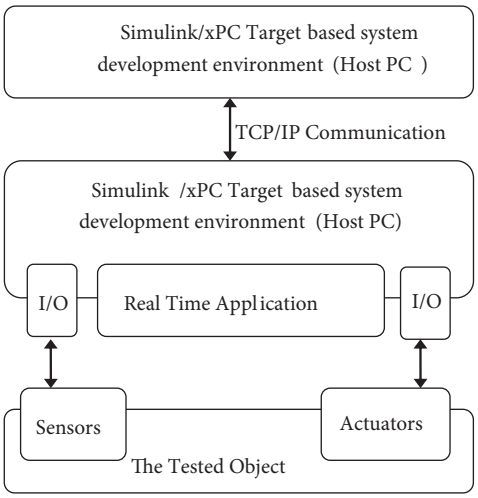


Figure 5. The structure of the real-time test with xPC Target.

The flow diagram of the program is given in Figure 6. As shown in this figure, vision, state, and movement subsystems forming the object follower system can be summarized as follows. An object is sensed as a processed image taken by the camera and the position of the object in the image is estimated. Depending on information on the object's position and presence/absence of the object, the object follower system determines the type of controller mode to run. The determined controller mode, which can be either velocity control or position control, is realized.

3.1. Vision subsystem

In this study, the considered object-based vision servo system needs a vision subsystem for taking images of the environment, sensing and selecting the following object from images, estimating the position error of the

object (distance from the center), and feeding back the error to the vision subsystem. Therefore, the vision subsystem consists of hardware and software components that can perform the mentioned operations. While the software, which can interpret the images, runs on the target computer in real time, the data of images can be taken by the web camera. The general flow diagram of the software developed for the vision subsystem is shown in Figure 7 [15]. Methods of threshold evaluation and labeling of the bounded component respectively are applied and then the elements related to the object are estimated by blob analysis.

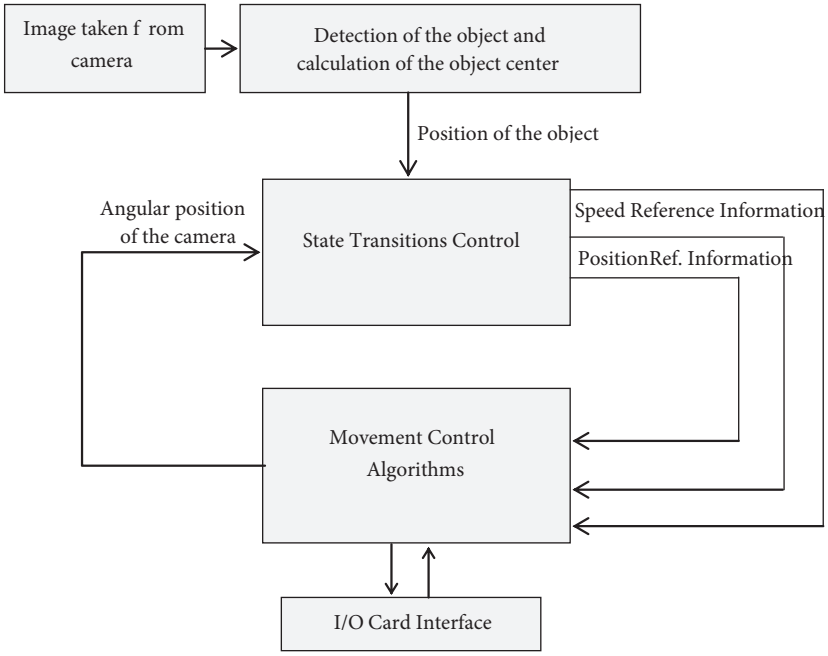


Figure 6. The working principle of the real-time program.

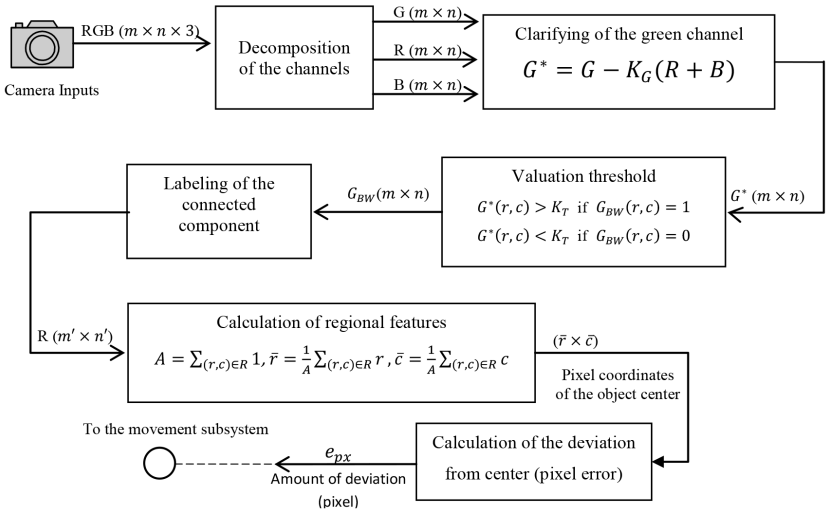


Figure 7. Functional diagram of the vision subsystem.

In tracing of the object by the system, the difference between the horizontal center of the object in the image, $(\bar{c})$, with center of the image $(m/2)$, is realized by reducing it to a minimum. The error, e_{px}=$\bar{c}$−$m/2$ as

pixels, is compensated with the horizontal rotational movement of the camera. In practice, as matching of the centers of the image and the object is not completely possible, the estimation of error is realized with a specific tolerance, $\mp\Delta\bar{c}$. Geometric notations on the taken image are given in Figure 8.

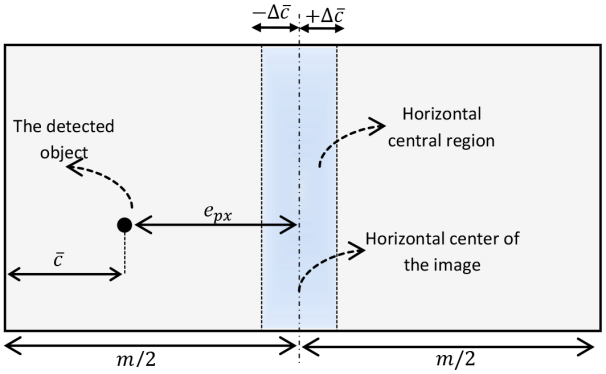


Figure 8. Geometric representations on the image.

When the object is detected in the vision area (and outside of the image center), the camera will turn towards the object with a constant velocity. When the image centers are equalized, the camera stops at the angular position of the present situation. This type of control can be realized with the state transitions described in Figure 9. Figure 10 shows a screen image obtained during the tests of the designed sight subsystem.

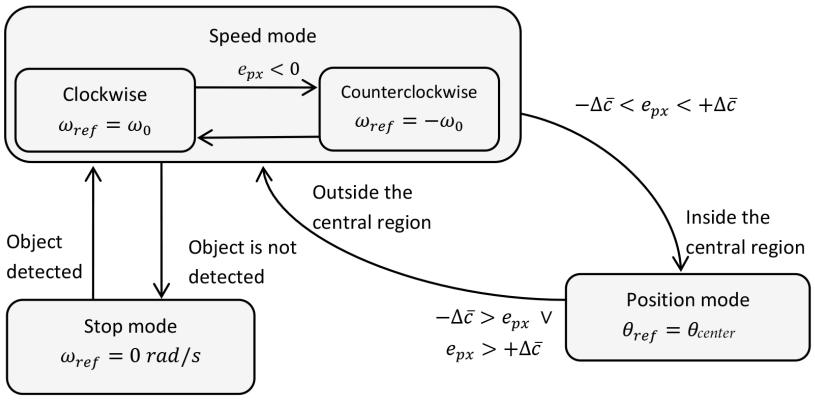
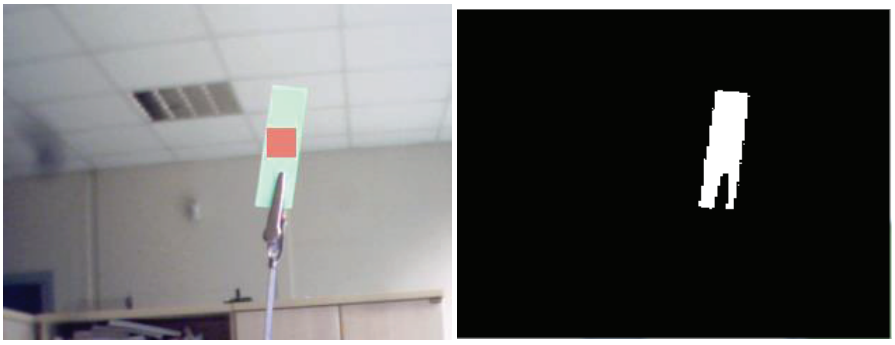


Figure 9. State transitions controls.



(a) Detected and calculated center object (b) Calculated blob

Figure 10. The test of the vision subsystem.

3.2. Subsystem of state transitions control

Depending on the operation principle of the object follower system, the system has different operation modes [15]. The object is either inside or outside of the image center and the object is either inside or outside of the vision area, and the system accordingly has different states. Figure 9 shows the operation states and the transition conditions of the system. The subsystem of the state transitions determines whether the controller realizes velocity or position control, depending on the different operation states. State transition controls of the given algorithm, shown in Figure 9, were designed in the MATLAB/Stateflow environment.

As can be seen from Figures 9 and 11, the angular position of the system can be realized with either velocity control or position control. Accordingly, when the system is initiated for the first time, the starting of the camera in the system is delayed for 2 s for giving time to the hardware. If the deviation is $-10 < e_{px} < 10$, the system turns out to be position control. Position reference of the controller will be the angular position of the camera at that moment ($\theta_{ref} = \theta$). When the object is outside of the horizontal image center, the system turns to velocity mode and approaches the object with a constant velocity. In this situation, depending on either $e_{px} > 0$ or $e_{px} < 0$, the velocity reference of the controller will be $\omega_{ref} = \pm 1$ rad/s.

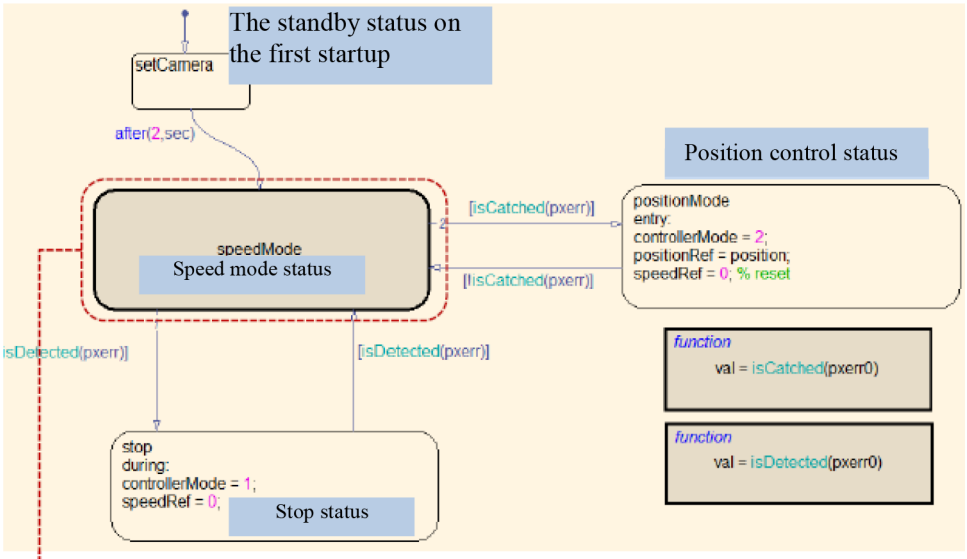


Figure 11. Stateflow scheme on state transitions status.

3.3. Motion subsystem

The angular movement of the camera used in the object follower system is realized by the motion subsystem. The motion subsystem consists of a DC motor/gear group for hardware and algorithms of position and velocity controllers for software.

Figure 12 shows the model of the control system related to the motion subsystem. In the design of the control system related to the motion subsystem, a procedure is followed as given in the following subsections.

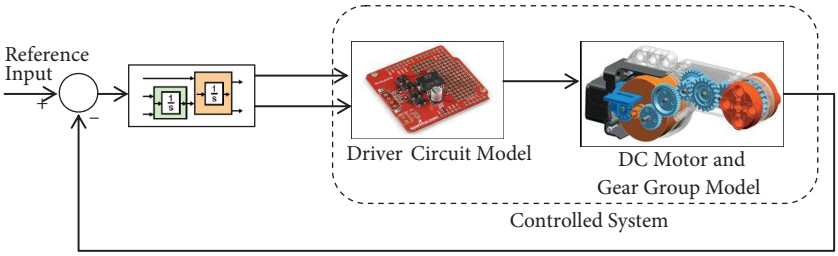


Figure 12. Control system model on movement subsystem.

3.3.1. Modeling of the controlled system

In the considered system, as the design of the controller is based on the system model, first modeling of the controlled system is needed. Therefore, a mathematical model of the components (DC motor/gear group and driving electronic) of the electromechanical system is obtained and then a Simulink model is built. Some unknown parameters such as viscous friction constant and inertia moment are estimated by using a parameter prediction method based on measured data. Defined parameters of the DC motor are given in the Table.

Table. DC motor parameters.

Parameter	Value
R_a- Armature resistance	6.69 Ω
L_a- Armature inductance	1.4×10^{-4}H
K_b- Opposite electromotive force coefficient	0.468 V/(rad/s)
K_t - Equivalent torque coefficient	0.317 Nm/A

3.3.2. Controller design

After obtaining a controlled system model verified by experimental data, a controller was designed for the system.

3.3.3. Implementing real-time tests

After designing and verifying the control system built in the simulation environment, control algorithms are tested as a real-time operation. In this study, the developed control algorithms are tested in xPC Target with a real electromechanical system as a real-time run. As a test method, rapid control prototyping and software in loop simulations are used. This is a recurrence producer. In the designed control system, model parameters are modified and real-time tests are repeated until the required performance is satisfied. After the performance criteria are satisfied, the most suitable parameters are used in the work.

4. Modeling of the controlled system and design of the controller algorithms

4.1. Modeling of controlled system

In Figures 13a and 13b, physical and block diagrams of the electromechanical subsystem are shown. As can be seen from Figure 13, the electrical system consists of resistance and inductance elements and the mechanical system consists of inertia mass and angular viscous friction elements. In the mathematical model of the rotational mechanic system, a voltage difference applied to the armature ends results in an angular motion in the mechanical system. In this system, as the rotor is forced to rotate in varying magnetic fields, an opposite electromotor force occurs against the current flow in the armature.

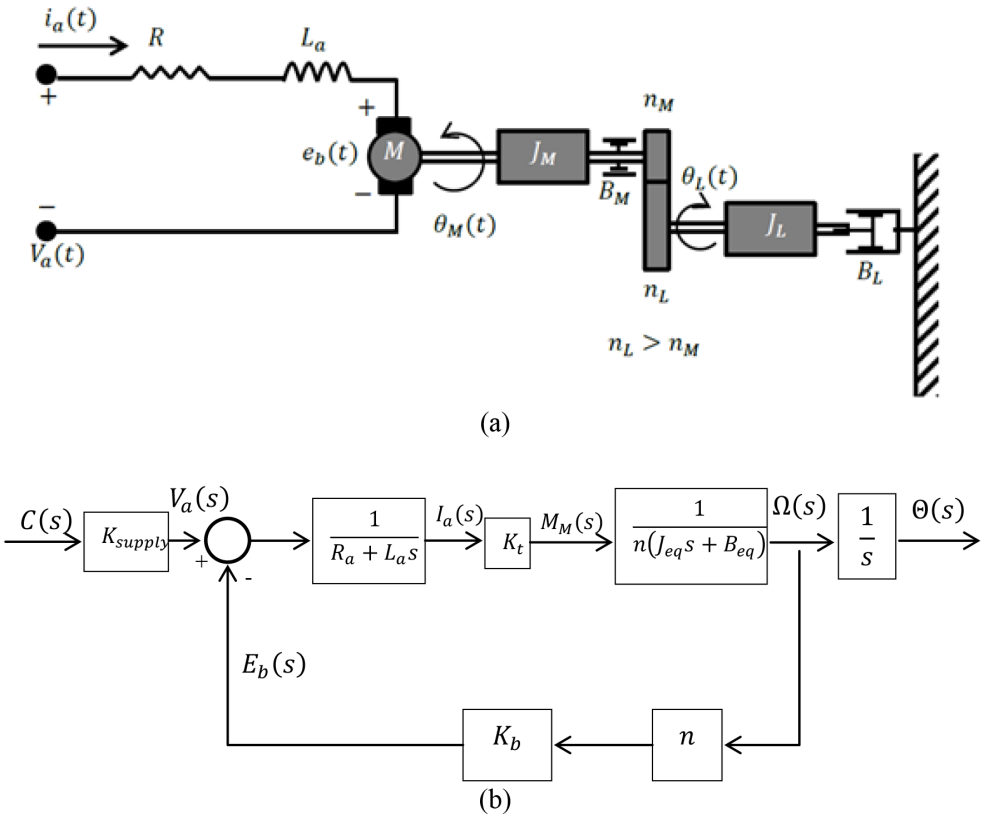


Figure 13. Block scheme of the electromechanical subsystem.

Figure 13b shows the block diagram of the DC motor, gear box, and driving circuit of the controlled system. Here C(s) defines the control signal. Θ(s) is the angular position of the load shaft and Ω(s) is the angular velocity of the load shaft.

Different approaches are used for modeling of the driving circuit. In the first approach, MATLAB/Simscape and MATLAB/SimElectronic libraries of the Simulink environment are used for modeling. In the second approach, a simplified mathematical model is obtained. In this study, the simplified model is analyzed.

In the modeling of the mechanical part of the electromechanical subsystem Newton's second law of motion is used and in the electrical part of it Kirchhoff's laws are used. Referring to the block diagram (Figure 13b),

the transfer function for the velocity and position output of the system are respectively as follows.

$$\frac{\Omega(s)}{C(s)} = \frac{K_{supply}K_t}{n(J_{eq}L_a s^2 + (B_{eq}L_a + R_a J_{eq})\, s + B_{eq}R_a + K_t K_b)} \tag{1}$$

$$\frac{\theta(s)}{C(s)} = \frac{K_{supply}K_t}{ns(J_{eq}L_a s^2 + (B_{eq}L_a + R_a J_{eq})\, s + B_{eq}R_a + K_t K_b)} \tag{2}$$

Here, L_a is neglected as the value is small compared to the other parameters. Therefore, a simplified transfer function for design of the velocity controller can be defined as below.

$$\left(\frac{\Omega(s)}{C(s)}\right)^* = \frac{K_{sys}}{(\tau s + 1)} \tag{3}$$

Here, time constant τ and gain of the system K_{sys} can be defined as follows.

$$K_{sys} = \frac{K_{supply}K_t}{n(B_{eq}R_a + K_t K_b)} \tag{4}$$

$$\tau = \frac{R_a J_{eq}}{B_{eq}R_a + K_t K_b} \tag{5}$$

The equivalent viscous friction constant, $B_{eq}(=B_m + \frac{B_L}{n^2})$, and the equivalent inertia moment, J_{eq} $(=J_m + \frac{J_L}{n^2})$, are estimated using the least square method with MATLAB. Here B_m and B_L are viscous friction constants of the motor and load, respectively. J_m and J_L are inertia moments of motor and load. In the program, the estimation was performed by parameter prediction based on measuring data. Experimental data required by parameter prediction were obtained through xPC Target via collecting shaft position data of the electromechanical subsystem. As a result of estimation, a value of B_{eq} of 4.031×10^{-8} Nm/(rad/s) and value of J_{eq} of 0.0084 kgm^2 were obtained.

4.2. Design of motion control algorithm

For the object follower system considered in this study, depending on the operation of the system, the system is either in velocity or position state. Therefore, the velocity control design of the DC motor is required. In the design of the velocity control design, the required criteria were taken as zero steady-state error for step input, maximum overshot of 5%, and settling time of less than 1 s.

P control was not enough to make the steady-state error zero. Therefore, it was decided to use a PI control for steady-state error. The control system with PI control is shown in Figure 14.

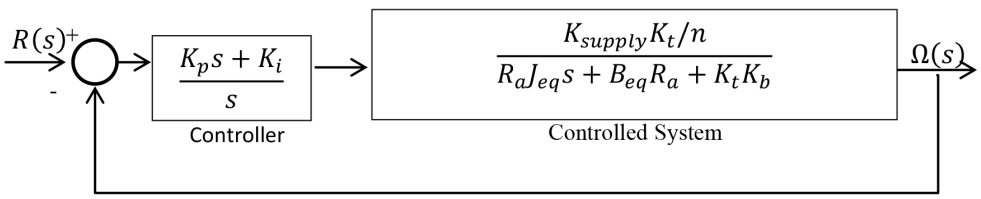


Figure 14. Block scheme of the system with PI controller.

In the design of the controller, root-locus and step response characteristics of the system were used. The design of the controller was mainly based on a linear model. In the meantime, a nonlinear model including friction force was considered. As a result of the estimation done to eliminate the delay caused by nonlinearity of the system at the beginning of response, proportional gain K_p of the PI control of 0.533 and integral gain K_i of 1.48 were determined. In Figures 15 and 16, the unit step response of the linear and nonlinear system for different PI configurations is shown. Simulation and experimental unit step responses of the system with the PI-x4 controller are given in Figure 17.

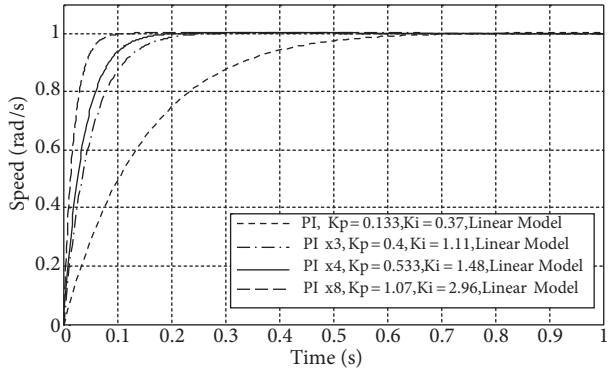


Figure 15. Step response of the linearized control system with various PI controllers.

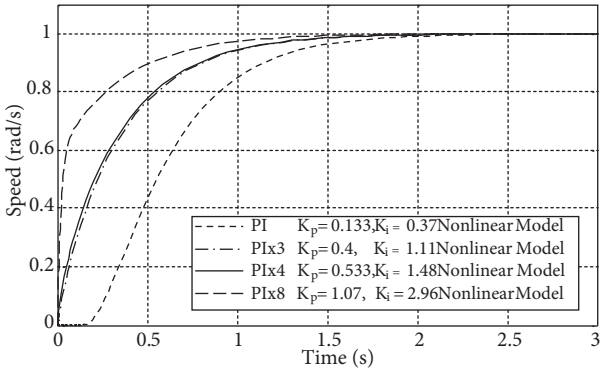


Figure 16. Step response of the nonlinear control system with various PI controllers.

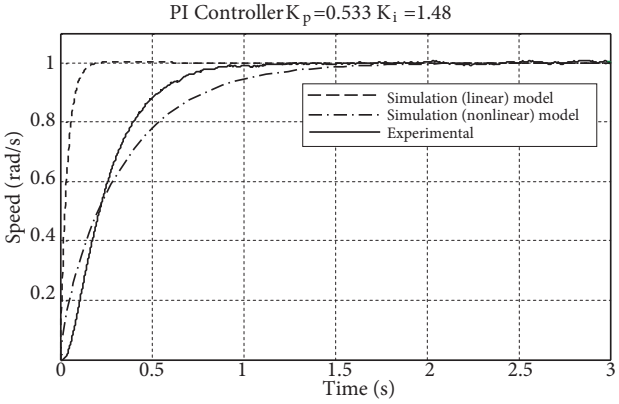


Figure 17. Simulation and experimental step responses of the system with PI-x4 controller.

In the position control of the system, PID control was aimed at for reducing the overshoot and reducing the effects of disturbances. The adjustment of necessary parameters was done with Simulink Control Design.

5. Results and discussion

In the previous sections of this work, the subsystems constituting the object follower system, which includes all design stages of a mechatronic system, were separately investigated.

In this section, the system is taken into consideration as whole and performance tests are applied to the system with the results examined.

For testing the performance of the object follower system an experimental grid was built on the test table (Figure 18). Seven different points were established on the grid. For the following performance, the object was

moved with relatively low and constant velocity on a predefined trace in the grid. For testing the following performance, two different test types were applied. Reaction times of the system were measured with a "still objects" test and the following performance of the system was tested with a "moving objects" test. Tests were repeated with different controller configurations and the effect of the velocity control was presented on the general system performance.

While the tests were being implemented, the position of the object as pixels in the image of the object taken by the system, angular position of the camera, mode of the controller, and reference information of position control were collected, recorded, and visualized. Interpretations of experimental results were based on the above information.

5.1. Tests of still objects

Reaction time of the system was determined with the tests of still objects. Before the tests were started, the position of the camera was set to 0° and the object was left at point C (Figure 19). As the system started to work, the camera oriented towards the object. Tests were repeated with PI-x1 and PI-x4. With PI-x1 control, the arrival time of the object to the image center was 1.851 s and it was 1.106 s with PI-x4 controller configurations. Figure 20 shows test results of the PI-x4 controller. The following conclusions were reached from the investigation of the results in Figure 20.

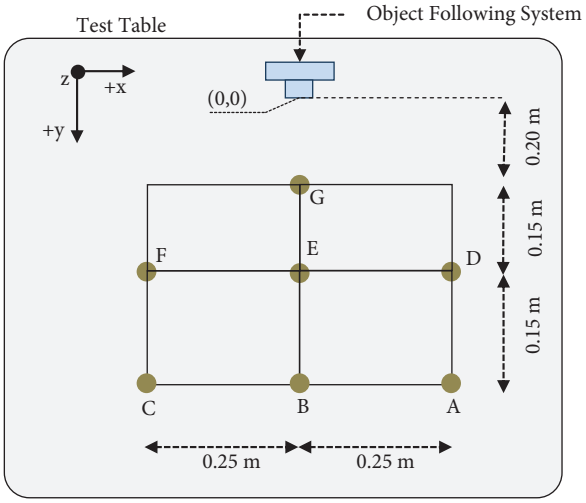


Figure 18. Experiment grid.

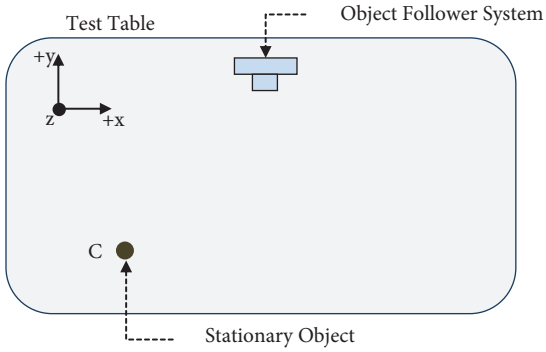


Figure 19. The test stand and the position of stationary objects.

- The object was detected in 0.111 s after the system was activated and the camera could start to move without any delay. This is because the configuration of the controller compensates friction torques. Therefore, reaction delays were eliminated.

- Detection of the object and arrival of the object to the image center takes 1.106 s (3.217–2.111).

- From t = 3.217 s, the center of the object in the image is the image center area. Following of the still object was successful.

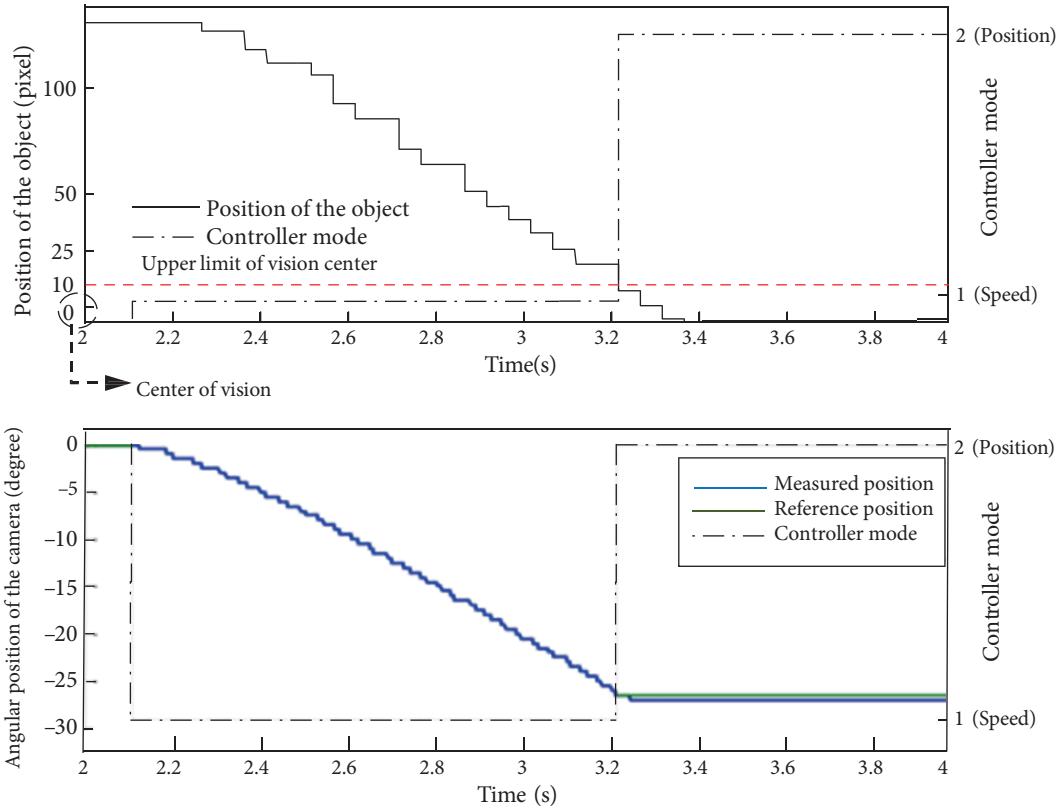


Figure 20. Experimental results of the stationary object tracking tests with PI-x4 controller.

5.2. Tests of moving objects

Following performances of moving objects were tested with this experiment. As can be seen in Figure 21, the object was moved respectively towards points B, A, G, and C with constant velocity and it was held at each point for 1 or 2 s. Because the DC motor does not have any velocity sensor, the velocity values of the motor were estimated from position data. For estimation of the velocity a "frequency measuring method" was preferred.

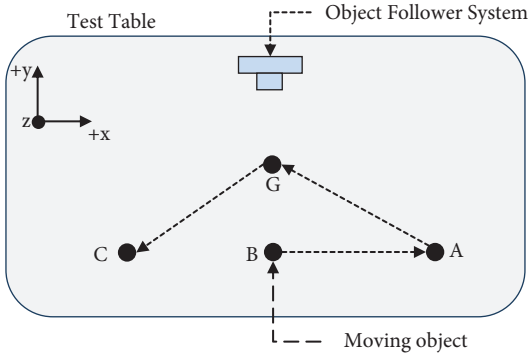


Figure 21. The test stand and the motion trajectory of the object.

In the controller configurations given in Figure 15, the tests were repeated with PI-x1 and PI-x4 for following performance of the moving objects. Thus, the effect of the relatively slower (PI-x1) and faster (PI-x4) controller on the general performance of the system was presented. With these tests, it was established that the

PI-x1 controller moves the moving object towards the boundary of the image center with more delays. With the PI-x4 controller, no delay was observed.

For evaluation of general performance of the system when PI-x4 control was applied, it will be sufficient to investigate the certain region of the graph given in Figure 22. The section of the object leaves point B and waits as point B is taken into consideration, as shown in Figure 23.

From the zoomed section in Figure 23, interpretations are as follows:

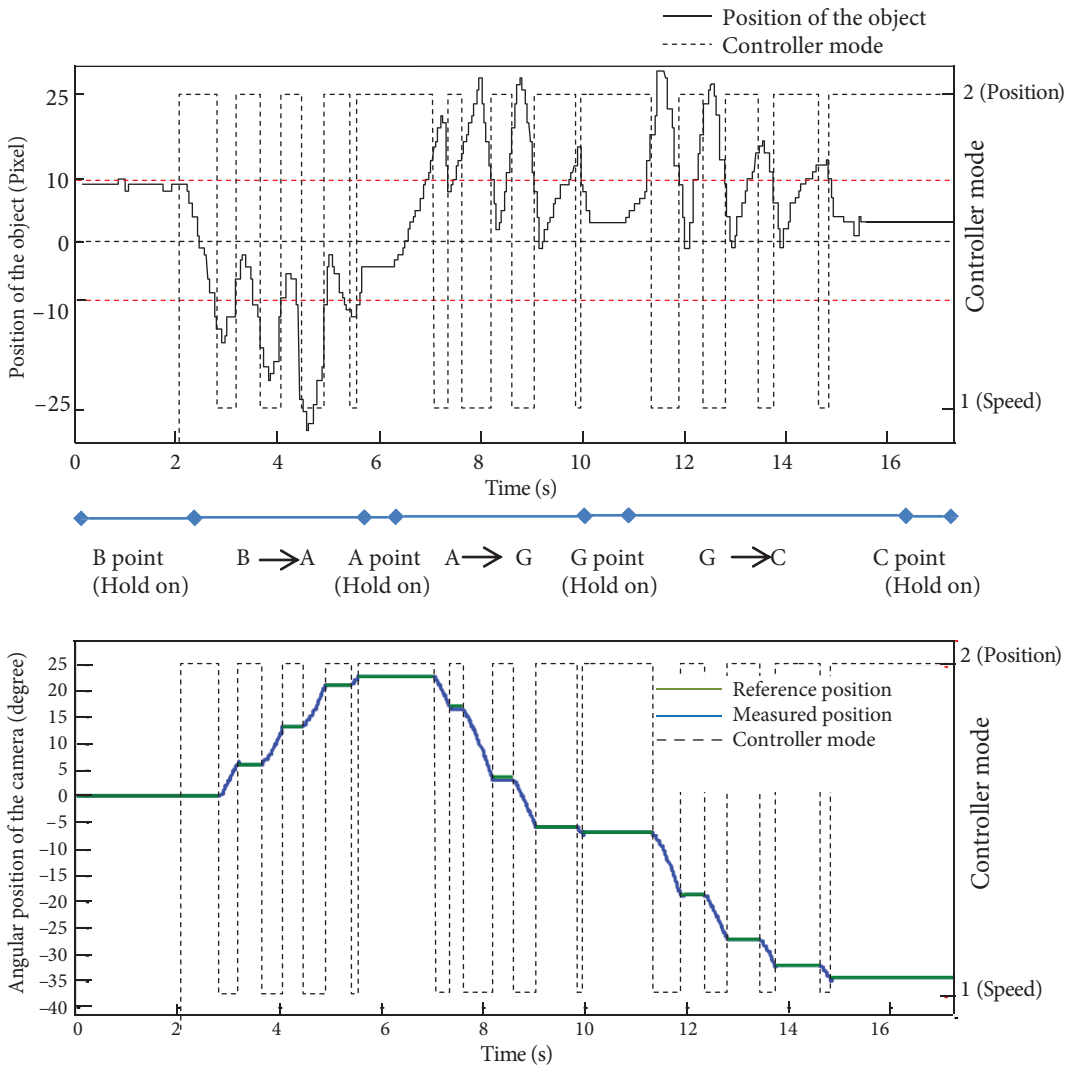


Figure 22. Experimental results of the motion tracking of the object with PI-x4 controller.

- The object leaves point B at t = 2.21 and arrives at point A at t = 5.56. Meanwhile, as the object just leaves the boundary of the image center, the system turns to "velocity mode" and until the center of the object coincides with the boundary of the image center the system carries on rotation with $\omega_{ref} = \pm 1 rad/s$ angular velocity. As a result, the object is again taken into the image center.
- The camera does not move as it arrives at point A; instead, it starts to move just as it leaves point B and starts to follow the object.

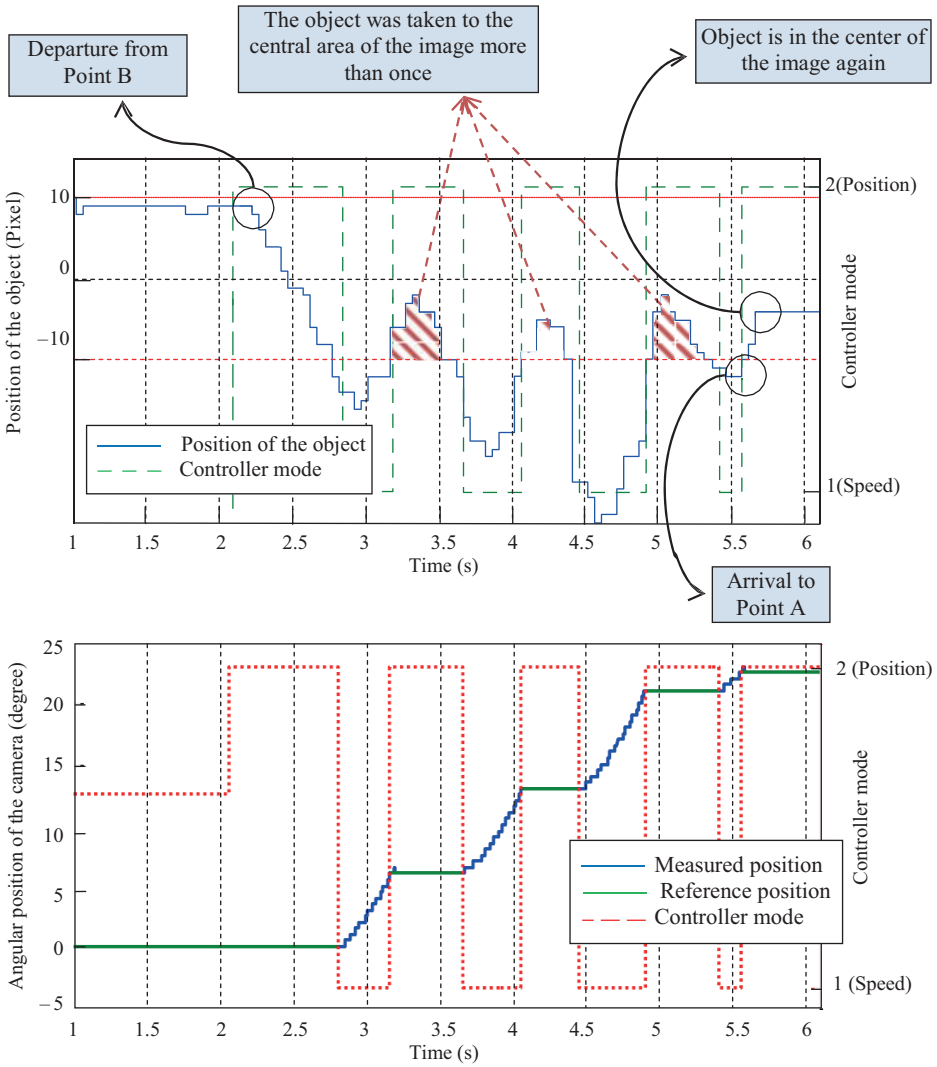


Figure 23. The zoomed graphic of the object when moving from B to A.

- According to the motion control strategy applied to the object follower system, it is required that as the object enters the boundary of the image center the movement of the system must be fixed. As can be seen in Figure 23, while the object is in motion the object is taken into the image center more than one time and it is that time the angular position of the camera sets as a reference position, and then this reference point is followed with zero steady-state error.

As a result of this, the designed system catches the object in the image center without any delay.

6. Conclusion

The design of a mechatronic system was realized with a model-based design approach.

As a results of works done on the object follower system the below conclusions were obtained:

- Detection of the object that has a specific color and a limited area in the image and the estimation of factors related to the object were accomplished.
- In the stage of control design simulation models of the system were used instead of physical prototypes. Thus, different control algorithms were rapidly tested and design of the control system was completed

rapidly. Unsuitable control strategies and errors were realized and corrected in the early stage of the design.

In future works, it is aimed to realize an object follower system with two axes by adding a vertical axis to the existing system. It is also aimed to realize a more complex object follower system that can follow object with more than one specific color and area. Different control techniques can be applied to the system.

References

[1] Chaumette F, Hutchinson S. Visual servo control part 1: basic approaches. IEEE Robot Autom Mag 2006; 1070: 82-90.

[2] Lee SW, Wohn K. Tracking Moving Objects by a Mobile Camera. Department of Computer & Information Science Technical Reports. Philadelphia, PA, USA: University of Pennsylvania, 1987.

[3] Kim KK, Cho SH, Kim HJ, Lee JY. Detecting and tracking moving object using an active camera. In: IEEE 7th International Conference on Advanced Communication Technology; 21–23 February 2005; Gangwon-Do, Korea. New York, NY, USA: IEEE. pp. 817-820.

[4] Mir-Nasiri N. Camera-based 3D object tracking and following mobile robot. In: IEEE Conference on Robotics, Automation and Mechatronics; 1–3 June 2006; Bangkok, Thailand. New York, NY, USA: IEEE. pp. 1-6.

[5] Lang H, Wang Y, Silva CW. Vision based object identification and tracking for mobile robot visual servo control. In: IEEE 8th International Conference on Control and Automation; 9–11 June 2010; Xiamen, China. New York, NY, USA: IEEE. pp. 92-96.

[6] Kim K, Lepetit V, Woo W. Real-time interactive modeling and scalable multiple object tracking for AR. Comp Graph 2012; 36: 945-954.

[7] Mohammed AMS, Yuegang T, Liang L. Research on trajectory tracking control for 3-axis servo system based on optimal preview control method. Procedia Eng 2011; 24: 297-302.

[8] Pomares J, Perea I, Jara CA, Garcia GJ, Torres F. Dynamic visual servo control of a 4-axis joint tool to track image trajectories during machining complex shapes. Robot Cim-Int Manuf 2013; 29: 261-270.

[9] Nasisi O, Carell R. Adaptive servo visual robot control. Robot Auton Syst 2003; 43: 51-78.

[10] Cubber GD, Berrabah SA, Sahli H. Color-based visual servoing under varying illumination conditions. Robot Auton Syst 2004; 47: 225-249.

[11] Huang S, Lin S. Application of visual servo-control X–Y table in color filter cross mark alignment. Sensor Actuator 2009; 152: 53-62.

[12] Jaradat MAK, Al-Fandi M, Nasir MT. Automatic control for a miniature manipulator based on 3D vision servo of soft objects. Mechatronics 2012; 22: 468-480.

[13] Orehek M, Robl C. Model based design in the development of thermodynamic systems and their electronic control units. In: Proceedings of the 2002 IEEE International Conference on Control Applications; 18–20 September 2002; Glasgow, UK. New York, NY, USA: IEEE. pp. 707-712.

[14] Kirby B, Kang H. Model based design for power systems protection relays, using MATLAB & SIMULINK. In: IET 9th International Conference on Developments in Power System Protection; 17–20 March 2008; Glasgow, UK. New York, NY, USA: IEEE. pp. 654-657.

[15] Demirkesen A. Modeling, simulation and implementation of a mechatronics system and its real-time control. MSc, Uludağ University, Bursa, Turkey (in Turkish with an abstract in English).

[16] Demirkesen A, Topçu EE, Yüksel İ. Model tabanlıbir nesne takip sisteminin incelenmesi. In: Turkish National Meeting on Automatic Control; 26–28 September 2013; Malatya, Turkey. pp. 314-319 (in Turkish).