Ball-and-beam laboratory system controlled by Simulink model through dedicated microcontrolled-Matlab data exchange protocol

Krzysztof Nowopolski
Poznań University of Technology
60-965 Poznań, ul. Piotrowo 3a, e-mail: Krzysztof.Nowopolski@put.poznan.pl

A ball balancing on a beam is a perfect platform for demonstration various control strategies. The task is to place ball rolling freely, in a reference position along the beam driven by a motor, rejecting disturbances. The research system consists of mechanical platform with stepper motor and two infrared ball position sensors, power supply and electronic controller based on a 32-bit microcontroller connected to Matlab/Simulink software running the main control algorithm. Designed in Simulink input and output subsystem blocks contain all functions required to provide data exchange with microprocessor controller of the mechanical system. Developed data exchange protocol and its implementation both in microcontroller software and Simulink subsystems allow to use this complete system for educational purposes: the controller designer is free to develop supervisory control layer, not going into details of system operation. The results of system operation with P, PD and LQR control laws are shown at the end of this paper.

KEYWORDS: beam-and-ball, interface, MATLAB, Simulink, stepper motor, SimMechanics, linear position control, microcontroller, 32-bit microcontroller, LQR control law

1. Introduction

Modern university courses of control theory should not focus only on control systems simulations and textbook computational exercises. There is a rising demand on impressive laboratory systems that can be easily configured and reprogrammed for work with user-defined control strategies to develop students engineering abilities. An objective of the design and research work described in this paper is to develop a laboratory control system with an interface to Matlab computing environment. The main requirements were: interface that allows to implement control laws in an easy manner (graphical user interface is preferred), compact size, low cost, simple design, minimalized number of sensors, utility for education purpose. There are only a few classical systems, that can satisfy this conditions, one of them is the presented ball-and-beam laboratory system.

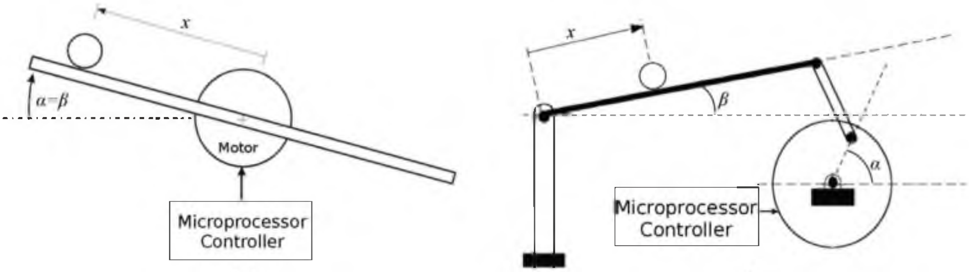
The system consists of a ball rolling freely on a beam, which is driven by an electric motor. The task is to keep the ball in one reference position rejecting disturbances, or to track the reference position signal, tending to keep the control error at negligible level. Both accuracy and dynamic aspects of the control need to be taken into consideration during design of the control strategy.

The system is unstable, underactuated and nonlinear and therefore it is treated as an appreciated benchmark for various control strategies [1]. Owing to the presented in the article microcontroller-Matlab data exchange interface, this laboratory system can be easily adapted as control theory laboratory practice for Control Engineering, Computer Science or Electrical Engineering students.

2. Laboratory system

2.1. Mechanical system

There are two basic mechanical layouts of the beam-and-ball system. First one, shown in the Figure 1a, consists of an electric motor, connected directly through its shaft to the beam, which center is located in the motor shaft axis. The main principle of the second layout is shown in the Figure 1b. One end of the beam is attached to a motionless rod, and the second one is connected to a moveable rod (length *L*) attached to a disc (diameter *2R*) driven by the motor. All mentioned joints are fitted with ball bearings.



(a) (b)

Fig. 1. Two basic layouts of ball-and-beam system: a) beam driven by a rod, b) motor shaft placed in the center of beam

Since the composition of mechanical parts of second layout creates a mechanical gear, this solution ensures much better accuracy of the beam positioning than a layout with a motor shaft placed in the center of the beam. The correlated result of the gear application is decreased demand for electric motor power. On the other hand, for first layout much less mechanical parts are needed. It brings two main advantages: the costs of the whole construction decrease, and the kinematic chain complexity is lower. In first layout there is only one angular position that needs to be taken into consideration. In the second one, there are three angle values that need to be processed to find system state accurately. Due to this fact, the number of trigonometric functions that need to be computed is enlarged, so more powerful controller is required to manage with this task. Although, for small tilt angles some simplifications may be introduced. Under condition, that the motor angle tends to 0, it can be assumed that moveable rod is perpendicular to the beam, so trigonometric transformations are negligible in this joint. Therefore, the relationship between motor shaft angular position α and beam angular position β is approximated by formula 1.

$$\beta = \frac{R}{L}\alpha \tag{1}$$

In order to increase the accuracy of angular positioning of the beam, an additional gear is added between motor shaft and the disc.

Before the ball-and-beam mechanism was constructed, complex analysis of the system operation, based on a computer model, was done. The model of ball movement was derived explicitly from the Lagrange-Euler equations, presented also in [2, 3]. Formula (2) describes relationship between ball linear position x and beam inclination β. Ball mass, radius and momentum of inertia are represented by parameters m, r and J respectively.

$$\ddot{x}(t) = \frac{m \cdot g}{m + \frac{J}{r^2}} \cdot \sin(\beta(t)) \tag{2}$$

Due to mentioned complicated structure of the mechanical construction, the system driving the beam (kinematic chain between motor and the beam) was modeled by using Simscape SimMechanics toolbox. This powerful environment allows to analyze kinematic properties of the modeled system after providing basic parameters of bodies and joints that it contains, not going into details of trigonometric transformations between parts of the system.

The assumed dynamics of the system was rather low, hence the application of a stepper motor was acceptable. This way motor shaft position control is kept at extremely simple level, since it can be run in an open loop, with no additional encoders or other angular position sensors. The electrical stepper motor driving the beam, was simulated using the following equations [4]:

$$\begin{cases} T(t) = K \cdot \left[-\sin(N\theta) \quad \cos(N\theta) \right] \cdot \mathbf{i}(t) \\ \dfrac{d\mathbf{i}(t)}{dt} = -\dfrac{R}{L} \cdot \mathbf{i}(t) + \dfrac{1}{L} \cdot \mathbf{u}(t) + \dfrac{K}{L} \cdot \omega(t) \cdot \begin{bmatrix} \sin(N\theta) \\ -\cos(N\theta) \end{bmatrix} \end{cases} \tag{3}$$

Two-element vectors **i** and **u** represent respectively currents and voltages in stepper motor windings. Basic motor parameters that have been taken into consideration are: winding resistance R and inductance L, and motor constant K. The motor constant is proportional to number of phases, stator poles and magnetic flux of each pole. The output of the motor model is torque signal, that is multiplied by the gear coefficient n and then feeds the SimMechanics system model. The

SimMechanics model output values are motor angular velocity ω and position θ. These two values are multiplied by gear coefficient n to calculate the motor shaft angular position α and the beam inclination β. Interconnection between elements of simulation model is shown in the Figure 2.

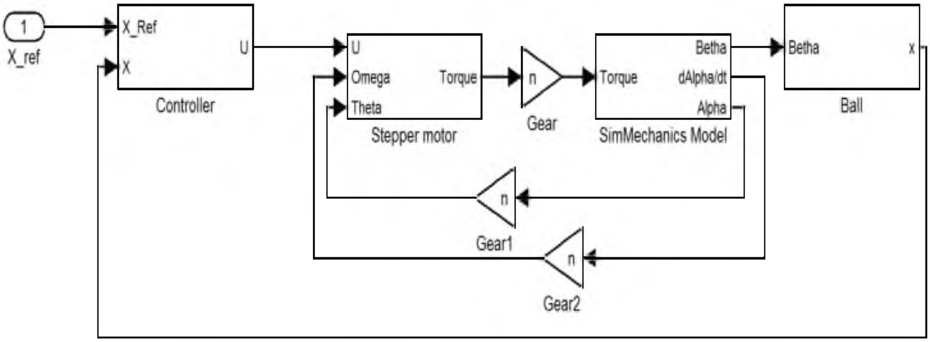


Fig. 2. Overview of computer model of the beam-and-ball mechanism

The simulation model was derived and designed for three general reasons: to analyze the behavior of the system, to choose the basic mechanical parameters of the system (such as dimensions and masses of the beam, rod and disc, required torque), and preliminary examine some basic and most promising control strategies. P, PD and LQR control laws were implemented as the controller of the simulation model. One of the basic conclusions of this simulation research was: it is essential to bring the velocity to zero, not only control the ball position. Therefore the most simple proportional controller will not allow to stabilize the ball in a reference position.

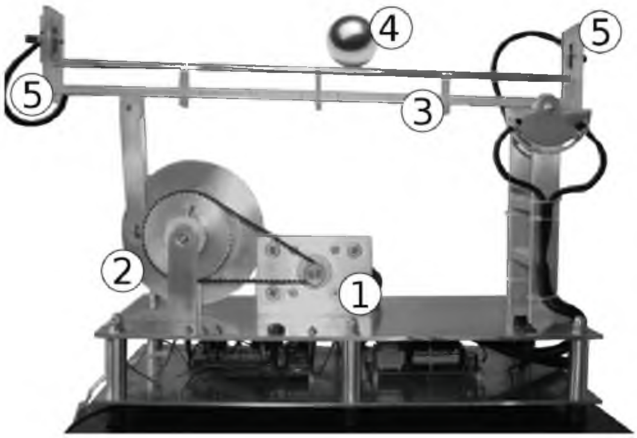


Fig. 3. Overview of the built laboratory system: 1 – stepper motor (behind a metal plate), 2 – disc connected to the beam by a rod, 3 – beam, 4 – ball, 5 – infra-red position sensors

Taking into consideration results of the described analysis, the mechanical construction of the laboratory system was designed and built. The model was used to choose an optimal electrical motor for this system, as well. The overview of the construction is shown in the Figure 3.

2.2. Microprocessor controller

The heart of the microprocessor controller of the laboratory system is STM100RB microcontroller. The following operations are the main tasks of this 32-bit ARM architecture device: driving the stepper motor, measurement of the ball position and communication with supervisory control layer, i. e. computer with running Matlab/Simulink model. The microcontroller drives the motor by A4988 integrated circuit. This chip contains in one package both signal translator and two full transistor bridges, each for one phases of motor windings. Therefore, all signals that microcontroller sends to the A4988 driver are: selection of the microstep precision (configured only once at the system startup, 3 lines), direction signal (1 line) and step impulses (1 line). Both switching between phases and motor current control are provided by the A4988 driver. The microstep precision in stepper motor drive is a very desired feature, since it ensures smooth movement of the shaft even at low velocities, avoiding resonances that may occur in the mechanical construction and within the motor structure itself. On the other hand, reducing the microstep size may decrease also the repeatability of the drive. The selected microstep size is 1/16 of a full step and the maximum velocity is limited by microcontroller software to 28 deg/s to avoid losing steps.

The ball position measurement is based on two GP2Y0A41SK0F infra-red analogue position sensors placed at the both ends of the beam. This redundancy within the number of sensors is related with small operation distance of these devices, limited to about 30 cm (about 1 ft). The beam is 0,5 m long (1,7 ft), so one sensor is not able to measure the distance along whole beam. The other disadvantages of these infra-red sensors, such as long reaction time (about 30 ms) and small precision of resistive photoelement (low-budget substitute of CCD) are compensated by very low price. Voltage is the output signal of the position sensor. Since the signal power is very small, it is susceptible to interference. Therefore both analogue RC filter placed before AD converter of the microcontroller and digital filter implemented in microcontroller software are required to obtain useful signal. Since the response time of the sensor is so long, the filters can also have large time constants. In the next stage the infra-red sensors will be replaced by vision system.

Communication with computer is running by UART interface of the microcontroller. Since the vast majority of modern computers is not equipped with any RS-232C port, the common translator MAX232 (converting UART voltage levels to RS standard) was replaced by FT232RL integrated circuit. This device uses physical layer of USB interface, but reports in operating system of the computer as a virtual serial port (see Fig. 4). This way the simplicity of UART usage and its very good Matlab implementation may be combined with universality of the USB interface.

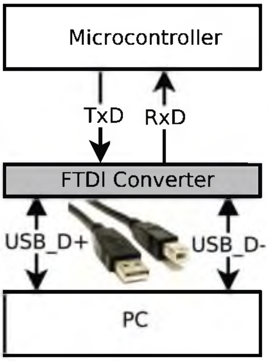


Fig. 4. Diagram of physical layers interconnection

3. Data exchange protocol

3.1. Frame of data and data exchange cycle

As it was mentioned in the section 2.2, the microcontrollers tasks are focused on the acquisition of ball position and stepper motor positioning. Whole algorithm of ball positioning and control law may be easily implemented and changed in real time in the running computer simulation, using GUI. This user-friendly way of operation is ensured by the data exchange protocol described below.

The data exchange protocol is very simple, but effective. Two sides of the communication process are: microcontroller that works as a server and a PC which is a client. There are only three different types of frames included in the protocol.

Each communication cycle is initiated by the client – the computer software. The initiation frame contains two bytes. These values is characteristic for transmission initiation and threaten in any other frame as inadmissible.

The microcontroller response contains six bytes: two integer values representing measurements of voltage outputs of the ball position sensors, and the third integer value – momentary position of the stepper motor shaft. The shaft position is estimated by counting the number of step impulses, assuming, that conditions of motor operation (limited rotational speed, small torques required to pull up and hold the beam) do not allow to lose steps. If there is no server (microcontroller) response, the initialization is repeated periodically.

After receiving the microcontroller response frame, Matlab control algorithm computes new reference motor shaft position. This information is sent to microprocessor system as a two-byte integer number, equal to the number of microsteps of the motor shaft to be done. Simple diagram in the Fig. 5 recapitulates the data exchange order.

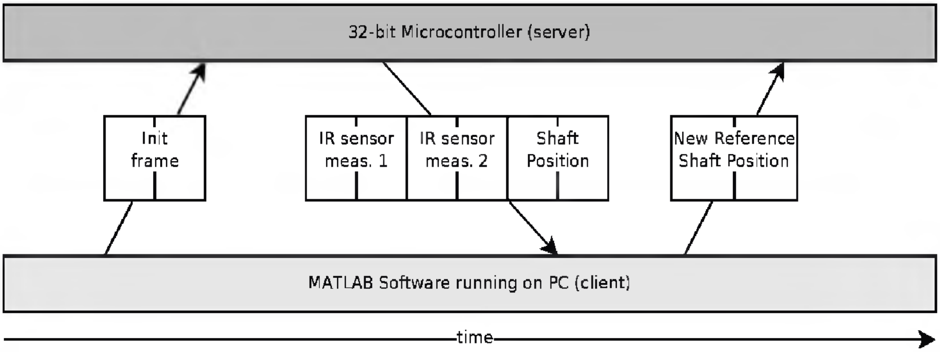


Fig. 5. Diagram of data exchange protocol

3.2. Matlab implementation

The Matlab implementation of the interface focuses on the usage of Simulink blocks from Instrumentation Control Toolbox. Incoming data are received by *Query Instrument* block, while the new reference shaft position is sent by *To Instrument* block. The basic configuration of both of these blocks is similar: a valid serial port number and baud-rate have to be selected. An important thing is to set a small received and sent data buffer sizes. If the buffer size is greater than 20 bytes, the software tends to fill the buffer firstly, rather than provide new data to the serial interface. This makes the communication too slow to control the ball position, since transmission of new reference values to microcontroller becomes strongly delayed.

Incoming data received by the *Query Instrument* blocks form a buffer, that needs to be split into particular values. It is done by *Unbuffer* blocks (see Fig. 6), that divide each frame into three values described in section 3.1. The motor shaft position can be easily converted to the angular position in degrees, but the ball position needs more specific operations to be found, since the voltage vs. position characteristic is strongly non-linear (Fig. 7). Moreover, the characteristic is ambiguous – therefore, all the measurements are treated as larger than 4 cm. The best results of the ball position x approximation gives inversion of position sensor sample v combined with 2nd degree polynomial (formula 4).

$$x(v) = \sum_{i=0}^{2} a_i \cdot \left(\frac{1}{v}\right)^i \tag{4}$$

Data received from two sensors are compared, and the measurement with higher voltage value is taken into consideration, since it indicates, that the ball is closer to this sensor and this measurement is more reliable. All the mentioned preprocessing operations are gathered in one subsystem, which structure is shown in Figure 6. This subsystem, combined with appropriately configured *To Instrument* block allows to design own control algorithms, not going into details of system operation. The examples are presented in the section 4. Real transmission speed is 125 communication cycles/s – it is over three times faster than the response time of the infra-red position sensors.

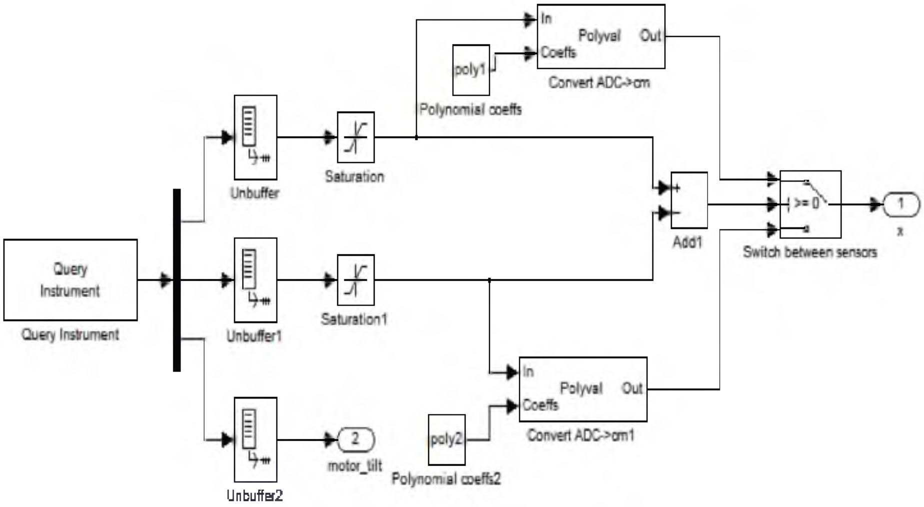


Fig. 6. Simulink subsystem for received data pre-processing

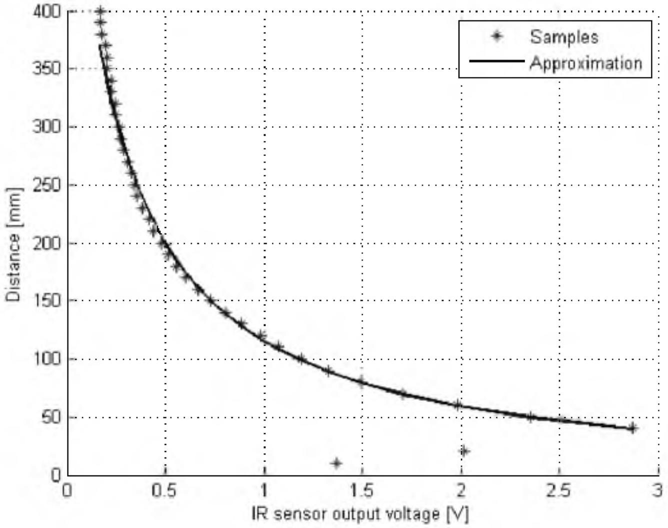


Fig. 7. Non-linear, ambiguous ball position sensor characteristic with its approximation

4. Tests of the complete control system

The most satisfactory results of laboratory system examination were obtained using LQR control law. Two communication subsystems described in section 3.2 were utilized as a hardware abstraction layer, all other blocks are standard Simulink operations. Simulink model calculation are run in constant-step mode, only discrete-time elements are allowed within model design. Controller model diagram is shown in the Figure 8.

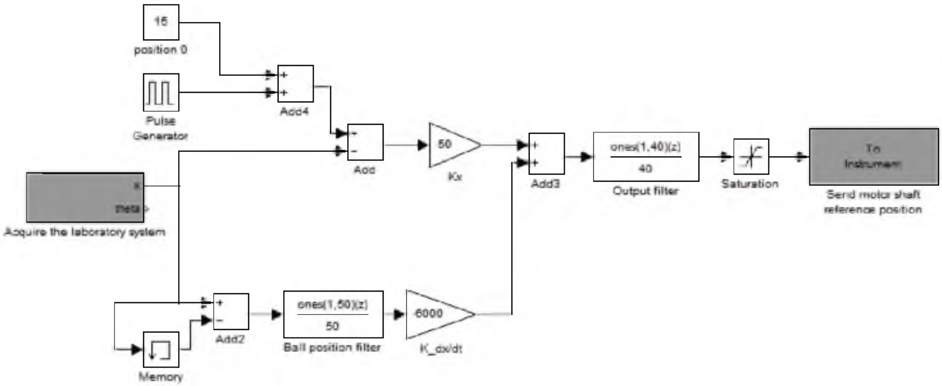


Fig. 8. Scheme of LQR controller designed for real laboratory system – two communication blocks are highlighted in grey

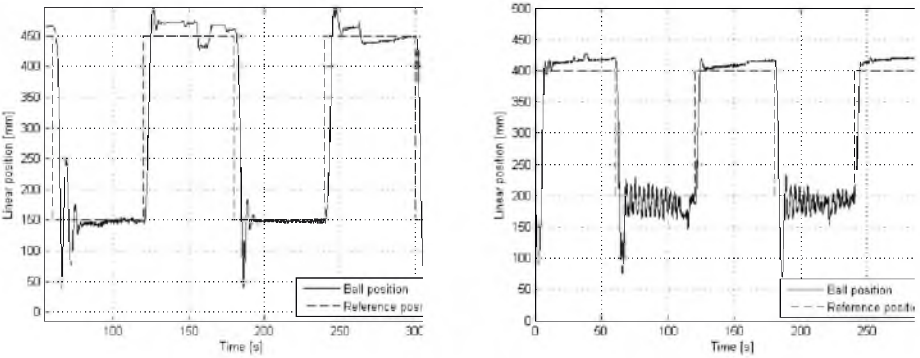
Since the linearized model of the process includes only two state variables (position and velocity of the ball, gathered in state vector **x**), the two-element **K** gain vector is required to be used in an implementation of LQR control law, given by the general equation 5 [5].

$$u = -\mathbf{K} \cdot \mathbf{x}^T \tag{5}$$

Position of ball is just multiplied by the first element of *K* vector. Velocity of the ball is computed as a difference of momentary positions, filtered by a running average filter with transfer function given by the equation 6.

$$G_{filter}(z) = \frac{1}{N} \sum_{i=0}^{N} z^{-i} \tag{6}$$

The horizon of the filter *N* is set to 50 samples, due to presence of significant disturbances in infra-red sensor measurements. These two values are summed and then filtered again before being sent. The results of operation with two **K** different values are shown in the figure 9. In the figure 9a the system operation with optimal (among tested set) controller parameters is shown. Figure 9b presents system operation when **K** vector values are too big – oscillations around reference value are observed.



(a) (b)

Fig. 9. Test of LQR control strategy with two **K** vector values: (a) $\mathbf{K}_1=[30, 5000]^T$, (b) $\mathbf{K}_2=[40, 7000]^T$

Besides the LQR control law, proportional and proportional-derivative control algorithms were tested. As it was mentioned in section 2.1, P controller is too simple to lead the ball to the reference position. In this control algorithm the ball velocity is not taken into consideration. When the ball approaches the reference position, its velocity is too big to be reduced by the small tilt of the beam. Therefore, the control system including proportional control law remains unstable (Fig. 10).

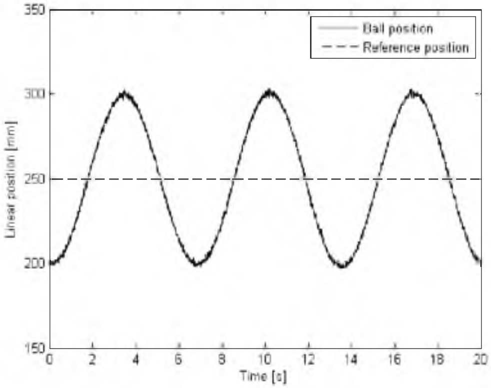


Fig. 10. Tests of basic control strategies: proportional control law

System operation with PD and LQR control laws are very similar (Fig. 11). Both of these algorithms control position as well as velocity of the ball. Application of LQR algorithm provides a second degree of freedom. It is noticeable in step switches between reference position. The PD control algorithm forms then impulses in output control signal, which cause additional oscillations and therefore need to be filtered in further processing. Beside this disadvantage, performances of these control laws are comparable.

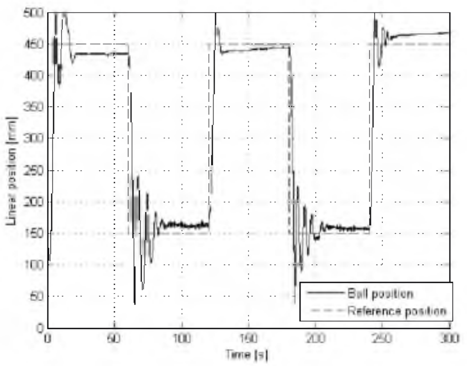


Fig. 11. Tests of basic control strategies: proportional-derivative controller

5. Conclusions

The objectives of the design and research work has been reached. The laboratory system including hardware and software parts was designed and launched. An inexpensive platform that can be utilized for educational purpose has been built and basic control algorithms have been tested. These algorithms can be utilized as one the first exercises in laboratory collage course. During system operation no problems with communication have been encountered. The main disadvantages, i.e. slow position sensor response and limited accuracy, are related with application of low-cost infra-red sensors.

References

[1] Marton L., Hodel A.S, Lantos B., Hung J.Y., *Underactuated Robot Control: Comparing LQR, Subspace Stabilization, and Combined Error Metric Approaches*, Transactions on Industrial Electronics (Vol. 55), p. 3724-3730, Oct. 2008.

[2] B. Messnerk, D. Tilbury, *Modeling the Ball and Beam Experiment*, University of Michigan 1997.

[3] Wenzhuo Chen, Xiaomei Sui, Yonghui Xing, *Modeling and Modulation of Nonlinear Ball-beam System Controller Based on Matlab,* 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), p. 2388-2391, 2012.

[4] S. Ossowski, *Modelowanie i symulacja układów i procesów dynamicznych*, Oficyna Wydawnicza Politechniki Warszawskiej 2007.

[5] J. Sheng, J. Renner, W. S. Levine, *A Ball and Curved Offset Beam Experiment*, American Control Conference, p. 402-408, 2010.