Chapter 14
Control of a Ball and Beam System

Consider the situation represented in Fig. 14.1. It is a ball that rolls inside a channel on a beam. The beam angle is modified by a permanent magnet (PM) brushed direct current (DC) motor and this produces the ball movement by the effect of gravity. The objective of the mechanism control is to stabilize the ball at some desired position on the beam. This mechanism is a very common workbench for classical and advanced control techniques [1, 2]. As shown in the present chapter, this mechanism is unstable and the classical design methods commonly introduce an internal control loop. These are the reasons for presenting this control problem in the present chapter.

The nomenclature employed is the following:

- u is the applied voltage at the motor armature terminals.
- x is the ball position measured from the beam left end.
- θ is the beam angle measured with respect to the horizontal configuration.
- m is the ball mass.
- R and r stand respectively for the ball radius and the ball rotation radius on the channel edge.
- i is the electric current through the motor armature.
- L is the armature inductance.
- R_a is the armature resistance.
- k_e is the motor counter electromotive force constant.
- k_m is the motor torque constant.
- J_m is the motor rotor inertia.
- b_m is the motor viscous friction constant.
- J_L is the beam inertia.
- b_L is the beam viscous friction constant.
- n_1 and n_2 stand for the teeth number at the motor shaft and the beam shaft respectively, and $n = \frac{n_2}{n_1}$.

V. M. Hernández-Guzmán, R. Silva-Ortigoza, *Automatic Control with Experiments*,
Advanced Textbooks in Control and Signal Processing,
https://doi.org/10.1007/978-3-319-75804-6_14

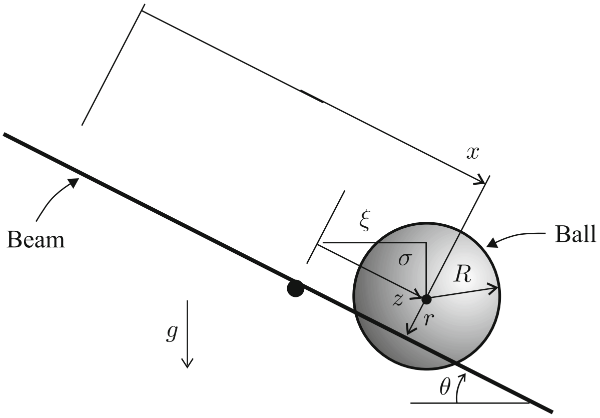


Fig. 14.1 Ball and beam system

14.1 Mathematical Model

14.1.1 Nonlinear Model

Consider Fig. 14.1. The beam length is L and suppose that the beam rotates around an horizontal axis at $x = L/2$. Note that:

$$z = x - \frac{L}{2}, \quad \xi = z\cos\theta, \quad \sigma = -z\sin\theta,$$

have been defined. From these expressions the following is obtained:

$$\begin{aligned}\dot{\sigma} &= -\dot{z}\sin\theta - z\dot{\theta}\cos\theta,\\ \dot{\xi} &= \dot{z}\cos\theta - z\dot{\theta}\sin\theta.\end{aligned}$$

It is also important to stress that it is assumed that the ball rotates without slipping, which is established by constraining the ball angular velocity ω and the ball translational velocity $\dot{z}$ to satisfy:

$$r\omega = \dot{z}, \quad \text{or} \quad \dot{z} - r\omega = 0 \tag{14.1}$$

as r is the ball rotation radius.

The computation of the *kinetic energy* for a body whose mass is distributed on a volume can be simplified if it is decomposed into two parts: one part is due to the translational movement of the body's center of mass (assumed to be a particle) and the other part is due to the body's rotative movement around its center of mass. For instance, the ball movement is easily described by its rotative movement around its center of mass (assumed to be located at its geometric center) and the translational movement of its center of mass. Thus, the kinetic energy of the ball is given as:

$$K_b = \frac{1}{2} J_b \omega^2 + \frac{1}{2} m(\dot{\xi}^2 + \dot{\sigma}^2).$$

where $J_b = \frac{2}{5} m R^2$ is the inertia of a solid sphere (the ball) with mass m and radius R rotating on its own axis [3, 4], pp. 271.

The total kinetic energy of the system is obtained just by adding the rotative kinetic energy of the beam, i.e.,

$$\begin{aligned} K &= K_b + \frac{1}{2} J \dot{\theta}^2 = \frac{1}{2} J_b \omega^2 + \frac{1}{2} m(\dot{\xi}^2 + \dot{\sigma}^2) + \frac{1}{2} J \dot{\theta}^2, \\ J &= n^2 \, J_m + J_L. \end{aligned}$$

Note that the inertia J takes into account the beam inertia and the motor inertia coupled with a gear box. The total potential energy is due only to the ball, i.e.,

$$P = \sigma m g = -z m g \sin \theta.$$

Thus, the Lagrangian of the system $L = K - P$ is given as:

$$L = \frac{1}{2} J_b \omega^2 + \frac{1}{2} m(\dot{\xi}^2 + \dot{\sigma}^2) + \frac{1}{2} J \dot{\theta}^2 + z m g \sin \theta. \tag{14.2}$$

According to the above arguments, the system's generalized coordinates can be defined as the ball's translational position $z = \pm\sqrt{\xi^2 + \sigma^2}$, the beam position θ, and the ball's angular position γ, i.e. $\dot{\gamma} = \omega$. Hence, according to the nomenclature in Appendix E:

$$q = [z, \theta, \gamma]^T, \quad \dot{q} = [\dot{z}, \dot{\theta}, \omega]^T.$$

On the other hand, according to (E.2), the constraint on the velocities given in (14.1) can be written as:

$$A(q)\dot{q} = 0, \quad A(q) = [1, 0, -r].$$

Thus, according to (E.4), the Euler–Lagrange equations representing the system model can be written as:

$$\begin{aligned} \frac{d}{dt}\frac{\partial L}{\partial \dot{z}} - \frac{\partial L}{\partial z} &= \lambda, \qquad (14.3) \\ \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} &= \tau_\theta - b\dot{\theta}, \end{aligned}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \omega} - \frac{\partial L}{\partial \gamma} = -r\lambda,$$

where $\tau_Q = [0, -b\dot{\theta}, 0]^T$, $b > 0$, has been assumed, i.e., no friction is considered to exist in the ball dynamics because it rotates without slipping, $\tau = [0, \tau_\theta, 0]^T$ with τ_θ the torque applied by the motor on the beam, and λ is an unknown scalar to be computed later. Replacing (14.2) in (14.3) it is found:

$$\begin{aligned}
&\frac{d}{dt}\left[mC\cos\theta + mB(-\sin\theta)\right] \\
&\quad - \left[mC(-\dot{\theta}\sin\theta) + mB(-\dot{\theta}\cos\theta) + mg\sin\theta\right] = \lambda, \\
&\frac{d}{dt}\left[mC(-z\sin\theta) + mB(-z\cos\theta) + J\dot{\theta}\right] \\
&\quad - \left[mC(-\dot{z}\sin\theta - z\dot{\theta}\cos\theta) + mB(-\dot{z}\cos\theta + z\dot{\theta}\sin\theta) + zmg\cos\theta\right] = \tau_\theta - b\dot{\theta}, \\
&J_b\dot{\omega} = -r\lambda,
\end{aligned}$$

where:

$$C = \dot{z}\cos\theta - z\dot{\theta}\sin\theta, \quad B = -\dot{z}\sin\theta - z\dot{\theta}\cos\theta.$$

Performing the indicated time derivatives and using some trigonometric identities in addition to $\lambda = -\frac{1}{r}J_b\dot{\omega}$, from the last equation, the above equations simplify to:

$$\begin{aligned}
m\ddot{z} - mz\dot{\theta}^2 - mg\sin\theta &= -\frac{1}{r}J_b\dot{\omega}, \\
J\ddot{\theta} + mz^2\ddot{\theta} + mz\dot{z}\dot{\theta} - mgz\cos\theta &= \tau_\theta - b\dot{\theta}.
\end{aligned}$$

According to (14.1), $r\dot{\omega} = \ddot{z}$, and hence:

$$\begin{aligned}
\left(m + \frac{1}{r^2}J_b\right)\ddot{z} - mz\dot{\theta}^2 - mg\sin\theta &= 0, \qquad (14.4) \\
J\ddot{\theta} + mz^2\ddot{\theta} + mz\dot{z}\dot{\theta} + b\dot{\theta} - mgz\cos\theta &= \tau_\theta.
\end{aligned}$$

These equations represent the ball and beam system mathematical model. Notice that this model is nonlinear because of several terms such as $-mz\dot{\theta}^2$, $-mg\sin\theta$, $mz^2\ddot{\theta}$, $mz\dot{z}\dot{\theta}$, $-mgz\cos\theta$. As this book is concerned only with linear control techniques, a linear model for the ball and beam system is required. Such a model is obtained in the following section by considering some simplifications of the above model.

14.1.2 Linear Approximate Model

It is interesting to note that, according to the term $-mgz\cos\theta$ in the second equation in (14.4), the ball weight exerts a torque on the beam axis, which depends on the ball position on the beam. On the other hand, an interesting phenomenon appears because of the term $-mz\dot{\theta}^2$ in the first equation: if the ball position $z > 0$ is large, the beam angle θ must be rendered negative very fast, i.e., $\dot{\theta}^2$ is large, to avoid the ball escaping (see Fig. 14.1), then, according to $-mz\dot{\theta}^2$ a centripetal force is exerted on the ball that forces it to continue escaping instead of stopping. It is not difficult to verify that a similar situation occurs when $z < 0$.

To avoid these phenomena appearing, the following assumptions are considered:

- Torque on the beam, due to weight of the ball, is negligible, which is especially true if the motor actuates on the beam through a gear box with a high reduction ratio. Under these conditions: $mgz\cos\theta \approx 0$ and $mz^2\ddot{\theta} \approx 0$.
- The beam and the ball always move at low velocities $\dot{\theta}$, $\dot{z}$, and the beam angle θ only takes values around zero. Under these conditions $\sin\theta \approx \theta$, if θ is given in radians, and $\dot{\theta}^2 \approx 0$, $\dot{z}\dot{\theta} \approx 0$.

Thus, the model in (14.4) becomes:

$$\left(m + \frac{1}{r^2} J_b\right)\ddot{z} - mg\theta = 0,$$
$$J\ddot{\theta} + b\dot{\theta} = \tau_\theta.$$

Differentiating $z = x - \frac{L}{2}$ twice yields $\ddot{z} = \ddot{x}$. Thus, the above model becomes[1]:

$$\left(1 + \frac{2}{5}\frac{R^2}{r^2}\right)\ddot{x} = g\theta,$$
$$J\ddot{\theta} + b\dot{\theta} = \tau_\theta. \tag{14.5}$$

where $J_b = \frac{2}{5}mR^2$ has been used. As this model is linear, the Laplace transform can be used in the first expression to find:

$$\frac{X(s)}{\theta(s)} = \frac{\rho}{s^2}, \quad \rho = \frac{g}{1 + \frac{2}{5}\frac{R^2}{r^2}}. \tag{14.6}$$

On the other hand, the electric motor can be modeled as in Sect. 10.1 taking into account that the load is represented by the beam, i.e.,

[1] Note that this model can also be expressed in terms of z: just replace $\ddot{z} = \ddot{x}$.

$$
\begin{aligned}
L\,\frac{di}{dt} &= u - R_a\,i - n\,k_e\,\dot{\theta},\\
J\ddot{\theta} &= -b\,\dot{\theta} + n\,k_m\,i, \quad n\,k_m\,i = \tau_\theta,\\
J &= n^2\,J_m + J_L, \qquad b = n^2\,b_m + b_L,
\end{aligned}
$$

where the second expression corresponds to (14.5). Proceeding as in Sects. 10.1, 10.2 and 10.3, the following current loop:

$$u_i = K(i^* - i),$$

and power amplifier:

$$u = A_p\,u_i,$$

can be used to find that the model of the motor and the beam is given as:

$$\theta(s) = \frac{k}{s(s+a)} I^*(s), \tag{14.7}$$

$$a = \frac{b}{J}, \quad k = \frac{nk_m}{J},$$

where it is assumed that no external disturbance is present, i.e., $T_p = 0$. Finally, the linear approximate ball and beam system model is given by the combination of (14.7) and (14.6), i.e.,

$$\frac{X(s)}{\theta(s)} = \frac{\rho}{s^2}, \quad \theta(s) = \frac{k}{s(s+a)} I^*(s). \tag{14.8}$$

14.2 Prototype Construction

The main components are the following.

- Portable computer with USB port and Windows XP.
- USB to series adapter.
- PIC16F877A Microcontroller from Microchip.
- DAC0800 LCN digital/analog converter.
- TL081 operational amplifiers.
- MAX232 driver.
- PM brushed DC motor. Nominal voltage 24[V], nominal current 2.3[A]. Provided with an optical encoder with 400 ppr.
- Gear box with reduction ratio $n = 30$.

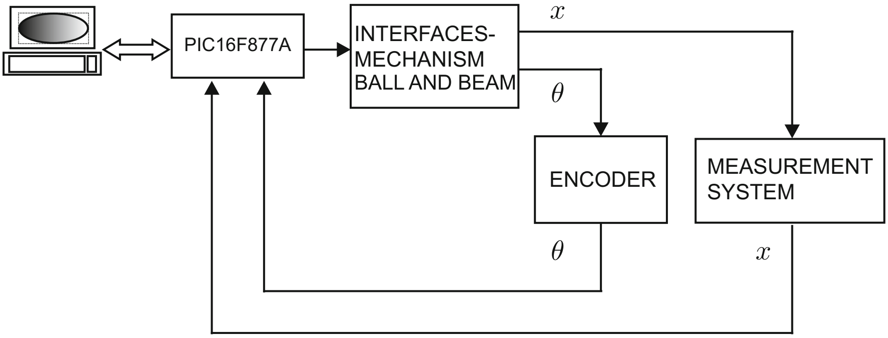


Fig. 14.2 Block diagram of the complete control system. Block interfaces–mechanism ball and beam is shown in Fig. 14.19. Block measurement system is shown in Fig. 14.3

- Beam of the system. Composed of: (i) A beam in wood with a circular section (rail number one), (ii) A beam in aluminum with a circular section (rail number two), (iii) A sheet of wood employed to hold both cylindrical beams.
- A ball position measurement system as described next.

A block diagram of the whole system is presented in Fig. 14.2.

14.2.1 Ball Position x Measurement System

One important and interesting component of the experimental prototype is the ball position measurement system. In fact, this is the most challenging component to construct. The most common way of performing the ball position measurements is employing a resistance distributed along one of the beam channel rails. The other rail is a simple electric conductor whose resistance can be considered to be zero. Hence, as the ball (made in an electric conductor material) rolls, putting both rails in electric contact, the second rail has a voltage that is equal to the voltage of the first rail at the point where the ball touches it. Thus, both rails work together as a potentiometer with the ball acting as the mobile contact.

The electric diagram of the ball position measurement system that has been built is shown in Fig. 14.3. Rail number one consists of a beam in wood with a circular section of about 0.01[m] in diameter and 0.7[m] in length. Magnetic wire is carefully wound onto this beam. All the wire loops must be tight to each other but without overlapping. Thus, a cylinder-shaped coil is obtained with a length of 0.7[m] and a diameter of 0.01[m] whose external surface is even enough to allow the ball to roll smoothly. The reason for using a beam in wood for this rail is to avoid the danger of putting the wire loops in a short circuit if the insulating paint fails.

Rail number two is a cylinder beam in aluminum with a length of 0.65[m] and a diameter of 0.01[m]. Both rails lay one in front of the other, in parallel, with

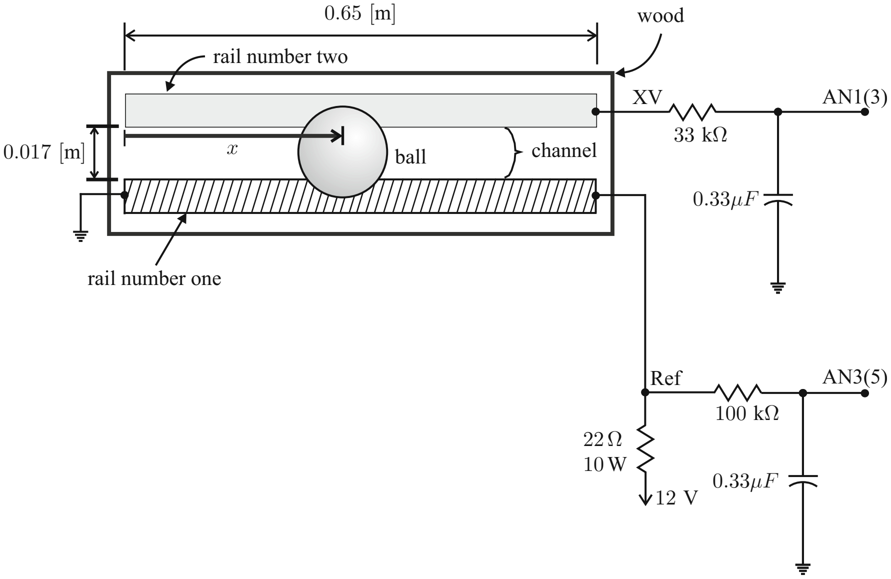


Fig. 14.3 Ball position measurement system

0.017[m] the minimal distance between them, on a sheet in wood. Hence, a channel measuring 0.65[m] in length and 0.017[m] in width is formed. This assembly constitutes the beam of the mechanism.

The ball dimensions are selected such that it is allowed to roll within the channel by putting both rails in electric contact but the channel width is enough to avoid the ball escaping. This feature is important, especially in the first experimental tests when it is usual for the ball to escape from the channel. Finally, the narrow strip where the ball touches rail number one is identified and the insulating paint is stripped with sandpaper along the entire length of rail number one. This allows electric contact, through the rolling ball, between the two rails. The ball that has been selected has the following dimensions (see Fig. 14.4):

$$2R = 0.0238[\text{m}], \quad 2z = 0.01895[\text{m}], \quad r = 0.0072[\text{m}]. \tag{14.9}$$

Finally, a gear box is employed to join the PM brushed DC motor and the beam to magnify the torque applied to the latter.

As stated above, both rails work together as a simple potentiometer delivering a voltage x_v that is proportional to the ball position x. The voltage x_v is measured using one of the analog/digital converters in the PIC16F877A microcontroller by connecting the terminal labeled AN1 to the microcontroller's pin 3. It is important to state that x_v has previously passed through a *low-pass filter* with 14.6148 [Hz] ($R = 33$[KOhm], $C = 0.33[\mu\text{F}]$) as corner frequency. Furthermore, to render the measurement system robust with respect to temperature changes in rail number one,

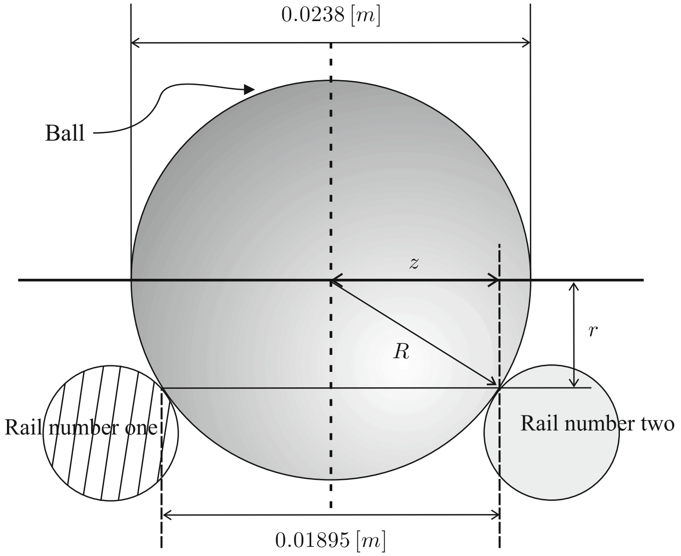


Fig. 14.4 Ball rolling in the channel, i.e., between the rails

voltage at the terminal labeled AN3 (connected to the microcontroller's pin 5) is employed as the reference voltage for the analog/digital converter used to measure x_v. It is also important to state that the signal at AN3 has previously passed through a low-pass filter with 4.8229 [Hz] ($R = 100$[KOhm], $C = 0.33[\mu$F]) as corner frequency.

14.2.2 Beam Angle θ Measurement System

The beam angle θ is measured using an optical encoder integrated into the PM brushed DC motor. The encoder is 400 ppr and is fixed to the motor shaft. Recall that a gear box with ratio $1 : 30$, i.e., $n = 30$, is used between the motor shaft and the beam shaft. Then, the position measured by the encoder (the motor position) must be divided by n to obtain the corresponding beam position measurement θ.

14.3 Parameter Identification

The experimental identification of parameters in the model (14.8) can be performed from the separate study of the motor and the beam dynamics (14.7) and the ball dynamics (14.6). This is explained in the following.

14.3.1 Motor–Beam Subsystem

The procedure suggested in this part is similar to that introduced in Sect. 11.1. Consider the following proportional position control:

$$I^*(s) = k_p A_\theta (\theta_d(s) - \theta(s)), \tag{14.10}$$

where A_θ is included because $\theta_v(s) = A_\theta \theta(s)$ represents the variable delivered by the measurement system and it is the variable actually handled by the computer employed as controller in the experiment. This means that the closed-loop system (14.7), (14.10), is represented as in Fig. 14.5. The corresponding closed-loop transfer function is:

$$\begin{aligned} \frac{\theta_v(s)}{\theta_{vd}s} &= \frac{A_\theta k_p k}{s^2 + as + A_\theta k_p k} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \\ \omega_n &= \sqrt{A_\theta k_p k}, \quad a = 2\zeta\omega_n, \end{aligned} \tag{14.11}$$

where $\theta_{vd}(s) = A_\theta \theta_d(s)$ is the desired beam position in terms of the variable delivered by the measurement system. Some experimental results are presented in Fig. 14.6 when $k_p = 4$ and $\theta_d = 0.2$[rad] is the desired value for θ. This experiment is performed without the ball between the beam rails or channel. From Fig. 14.6, the following is measured $t_r = 0.252$[s] and $M_p(\%) = 79$. These data are employed in:

$$\begin{aligned} \zeta &= \sqrt{\frac{\ln^2\left(\frac{M_p(\%)}{100}\right)}{\ln^2\left(\frac{M_p(\%)}{100}\right) + \pi^2}}, \\ \omega_d &= \frac{1}{t_r}\left[\pi - \arctan\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)\right], \\ \omega_n &= \frac{\omega_d}{\sqrt{1-\zeta^2}}, \end{aligned}$$

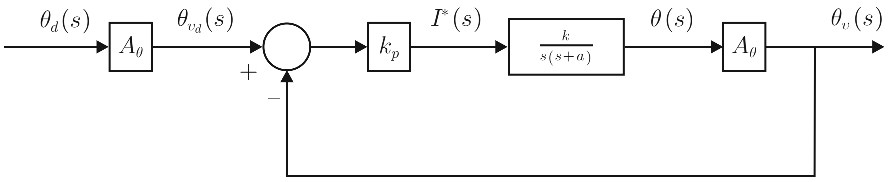


Fig. 14.5 Proportional control of the beam position

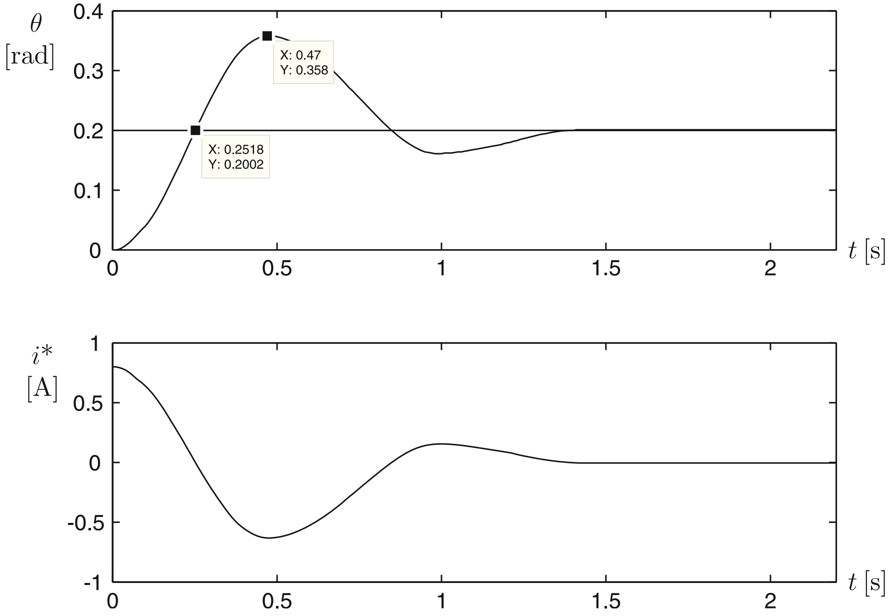


Fig. 14.6 Experimental results using $k_p = 4$ and the block diagram in Fig. 14.5

to find $\zeta = 0.0748$ and $\omega_n = 6.5489$[rad/s]. Finally, using the expressions in (14.11) with $k_p = 4$ and $A_\theta = 1$ (see Sect. 14.7), the following is obtained:

$$k = 10.7219, \quad a = 0.98. \tag{14.12}$$

14.3.2 Ball Dynamics

Applying the inverse Laplace transform to (14.6) and assuming all the initial conditions to be zero, the following is found:

$$\ddot{x} = \rho\, \theta(t). \tag{14.13}$$

Suppose that $\theta(t) = \theta_0$ where θ_0 is a constant. Integrating (14.13) twice:

$$x(t) = \frac{1}{2}\rho\, \theta_0\, t^2. \tag{14.14}$$

A procedure is presented next allowing us to experimentally estimate the parameter ρ.

- Fix the beam at a the constant angle $\theta = \theta_0$, where θ_0 is a small positive value. A good example is $\theta_0 = 0.2$[rad] because $\sin(0.2) = 0.1987$, i.e., $\sin(\theta_0) \approx \theta_0$.
- Place the ball at the beam left end and let it roll until reaching the beam end at the right (the beam must not move during this stage). Measure the ball position described in this experiment. This task is rendered easier if a computer is employed.
- Plot the ball position in this experiment versus time.
- Using θ_0 defined at the first step of this experiment and proposing some arbitrary positive value for ρ, plot the function $\frac{1}{2}\rho\ \theta_0\ t^2$ on the same axes employed at the previous step.
- Propose different values for ρ. The correct value of ρ is that allowing the plot of the function $\frac{1}{2}\rho\ \theta_0\ t^2$ to fit the plot of the experimental data.

The plots referred to in the previous procedure are shown in Fig. 14.7. when using $\theta_0 = 0.2$[rad] and:

$$\rho = 4.8. \tag{14.15}$$

It is important to verify that the experimental data x are plotted using meters as the measurement unit. This is rendered easier if $A_x = 1$, as explained in Sect. 14.7. Finally, it is interesting to state that the value for ρ shown in (14.15) has been corroborated using (14.6), (14.9) and $g = 9.81$[m/s^2], which yields $\rho = 4.687$.

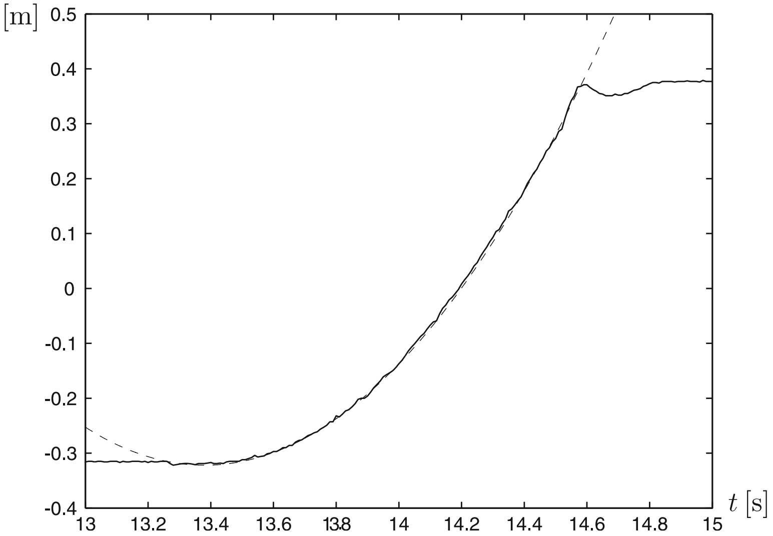


Fig. 14.7 Experimental identification of the parameter ρ. Continuous: $x(t)$ measured in the experiment. Dashed: $x(t) = \frac{1}{2}\rho\theta_0(t - 13.38)^2 - 0.322$, $\theta_0 = 0.2$[rad], $\rho = 4.8$

14.4 Controller Design

Note that the plant model (14.8) has three poles at $s = 0$, i.e., it is open-loop unstable. Hence, an important design objective is to achieve closed-loop stability. A block diagram of the control system to be designed is shown in Fig. 14.8a. This block diagram can be represented as in Fig. 14.8b. It is suggested that the reader might see Sect. 6.7.5 to understand the reason for this block diagram. In this design problem $\lim_{t\to\infty} x(t) = x_d$ must be accomplished, where x_d is a constant standing for the desired ball position on the beam.

Note that the open-loop transfer function of the system shown in Fig. 14.8b has two poles at $s = 0$, i.e., the system type is 2. Then, as x_d is constant, it is ensured that $\lim_{t\to\infty} x(t) = x_d$ if the closed-loop transfer function $X(s)/X_d(s)$ is stable (see Sects. 3.4 and 4.4). Hence, the only problem that remains is to find some constants α, k_v, γ, b and c such that the transfer function $X(s)/X_d(s)$ is stable. It is explained next how to achieve this.

The open-loop transfer function of the system shown in Fig. 14.8b is given as:

$$G(s)H(s) = \frac{A_x\,\rho\,\gamma b}{cA_\theta}\,\frac{s+b}{b}\,\frac{c}{s+c}\,\frac{\alpha k A_\theta}{s^2+(a+k_v k A_\theta)s+\alpha k A_\theta}\,\frac{1}{s^2}, \tag{14.16}$$

where $c > b > 0$. In Fig. 14.9, the Bode diagrams of $G(s)H(s)$ and each one of the transfer functions that comprise it are shown. Notice that, if the constant $\alpha k A_\theta$ is large enough, i.e., if $\alpha k A_\theta \gg s^2 + (a + k_v k A_\theta)s$, then:

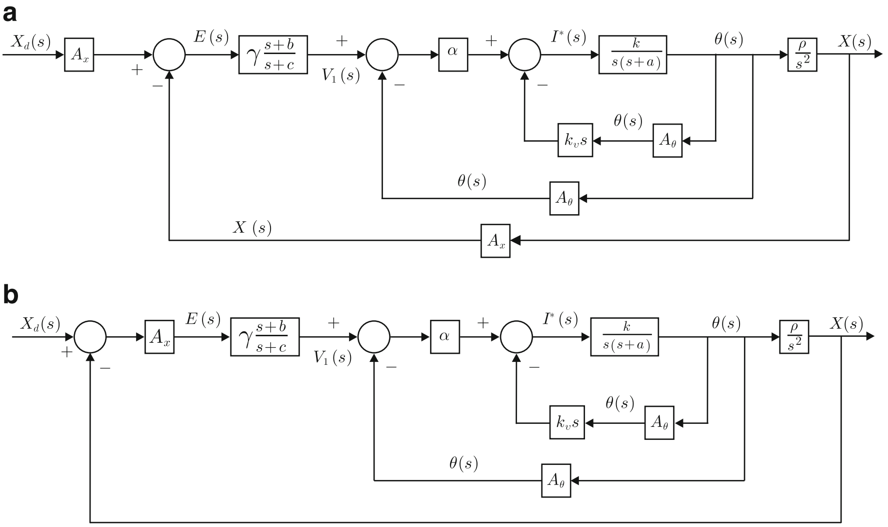
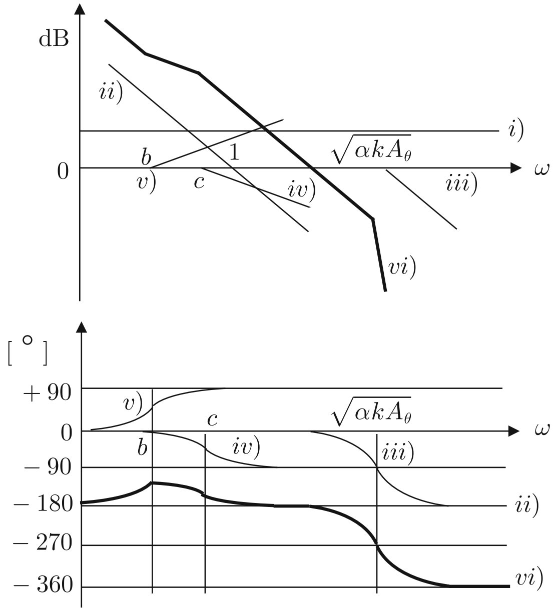


Fig. 14.8 Equivalent block diagrams of the control systems to be designed

Fig. 14.9 Bode diagrams of $G(s)H(s)$ in (14.16)



$$\frac{\alpha k A_\theta}{s^2 + (a + k_v k A_\theta)s + \alpha k A_\theta} \approx 1. \tag{14.17}$$

This can also be interpreted by thinking that the transfer function in (14.17) has 0[dB] as magnitude and 0 degrees as phase. It is interesting to observe that this situation can also be seen in Fig. 14.9: when $\alpha k A_\theta$ is large enough, i.e., when the corner frequency of the transfer function in (14.17) is located far to the right, this second-order factor has no effect on open-loop frequency response; thus, it can be neglected.

Hence, if $\alpha k A_\theta$ is large, the problem is simplified to achieve closed-loop stability for a system whose open-loop transfer function is given as:

$$\frac{A_x\ \rho\ \gamma b}{c A_\theta}\ \frac{s+b}{b}\ \frac{c}{s+c}\ \frac{1}{s^2}. \tag{14.18}$$

This is an important simplification because the phase lag of the transfer function in (14.17) is eliminated, which may take values between 0 and −180 degrees (when such a transfer function is not close to unity). This means that stabilizing (14.18) requires less phase lead than that required to stabilize (14.16). Recall that requiring more phase lead renders the controller design task more restrictive because the compensator $\gamma \frac{s+b}{s+c}$ would be more sensitive to noise in the position sensors.

Note that a large $\alpha k A_\theta$ can be achieved by enlarging α as k and A_θ are fixed once the prototype has been built. However, a large $\alpha k A_\theta$ causes the performance of the control system to deteriorate. In fact, large values for $\alpha k A_\theta$ were employed during the experimental tests and it was observed that this increases the deteriorating effect due to noise in sensors: the mechanism strongly vibrates and the control system does not work. Hence, α is limited by a maximal value imposed by the experimental prototype even when the condition (14.17) is not satisfied.

On the other hand, if $k_v = 0$ the system in (14.17) will be badly damped because $\zeta = a/(2\sqrt{\alpha k A_\theta}\,)$ is small. This produces a resonance peak in the Bode diagrams of (14.16). The problem with a resonance peak is that it increases the effect of high frequencies producing, again, undesirable vibrations in the mechanism. Good results have been obtained using some $k_v > 0$ because this increases the damping factor $\zeta = (a + k_v k A_\theta)/(2\sqrt{\alpha k A_\theta}\,)$ of the system in (14.17); hence, the resonance peak in the Bode diagrams of (14.16) is reduced.

According to the above arguments the following experiment was performed.

- Consider a closed-loop system to control only the beam position, i.e., only consider the block diagram between the signals $V_1(s)$ and $\theta(s)$ in any of the block diagrams in Fig. 14.8. This means that the ball must not be on the beam when performing the experiments.
- Set $V_1(s)$ as a step standing for the desired beam position and $k_v = 0$.
- Propose some values for $\alpha > 0$ such that the beam response is fast.
- When the oscillations are important, keep α constant and increase $k_v > 0$.
- Repeat the previous two items until the beam response is fast and well-damped but avoiding vibrations due to excessive noise amplification. Consider the resulting values of α and k_v as the correct values.

Once the above procedure had been performed, it was found that the use of the following controller gains achieves a satisfactory performance:

$$\alpha = 30, \quad k_v = 0.35 \tag{14.19}$$

Notice that the system in Fig. 14.8b, together with $A_x = A_\theta = 1$ (see Sect. 14.7), and (14.12), (14.15), (14.19) represent the control problem analyzed in Sect. 6.7.5.2, where the solution is given by the following compensator:

$$G_c(s) = \gamma \frac{s+b}{s+c}, \quad b = 0.6404, \quad c = 4.5746, \quad \gamma = 1.6406.$$

Finally, according to Sect. 6.5.2, to take into account the effects of time delay contributed by the digital implementation of the controller, the transfer function:

$$\frac{k}{s(s+a)},$$

in Fig. 14.8b may be replaced by:

$$\frac{k\,\mathrm{e}^{-Ts}}{s(s+a)},$$

as the time delay T appears at the controller output, i.e., at the plant input. Using suitable block algebra it is possible to show that the corresponding open-loop transfer function is the same as $G(s)H(s)$ in (14.16) with e^{-Ts} as a new factor. This means that the phase lag $-\omega T$[rad] must be added to the phase in Fig. 6.87. There, the crossover frequency is $\omega = 1.71$[rad/s]. Also $T = 0.01$[s] can be set as this is the sampling period used in the experiments presented next, i.e., the worst case time delay. Hence, $-\omega T \times 180°/\pi = -0.97°$. This means that the phase margin remains almost the same as in Fig. 6.87, i.e., about 47.5°, and the time delay induced by the digital implementation of the controller does not cause the performance to deteriorate.

14.5 Experimental Results

In Fig. 14.10 some experimental results are presented that are obtained when using the following desired ball position:

$$x_d = \begin{cases} 0.0, & 0 \le t \le 10 \\ 0.1, & 10 < t \le 20 \\ 0.2, & 20 \le t \le 30 \\ 0.1, & 30 \le t \le 40 \\ 0.0, & 40 \le t \le 50 \\ 0.1, & 50 \le t \le 60 \\ 0.2, & 60 \le t \le 70 \\ 0.1, & 70 \le t \le 80 \\ 0.0, & 80 \le t \end{cases}.$$

where x_d is given in meters and t in seconds. It can be observed that, although the ball position tracks its desired value, a steady-state error that is different from zero is present. This steady-state error is due to nonlinear friction introduced by the gear box. Also notice that the overshoot is not the same when responding to different step commands. This effect is mainly due to beam inertia changes because of the ball staying in a different position on the beam. Note, however, that this effect has not been considered in the mechanism model.

To measure the rise time and overshoot to one of the commanded step references, a zoom-in in Fig. 14.10 is presented in Fig. 14.11. There, it is observed that $t_r = 0.79$[s] and $M_p(\%) = 33\%$. The steady-state value of x is 0.077[m] in this figure. It is interesting to recall that a 30% overshoot and a 0.927[s] rise time were obtained through simulations in Fig. 6.88. These values are very close to those obtained through the experiments in Fig. 14.11.

The nonzero steady-state error observed in the above experiments may be attributed to nonlinear friction but, also, recall that a torque on the beam due to the ball weight has been neglected in Sect. 14.1.2 to obtain a linear approximation

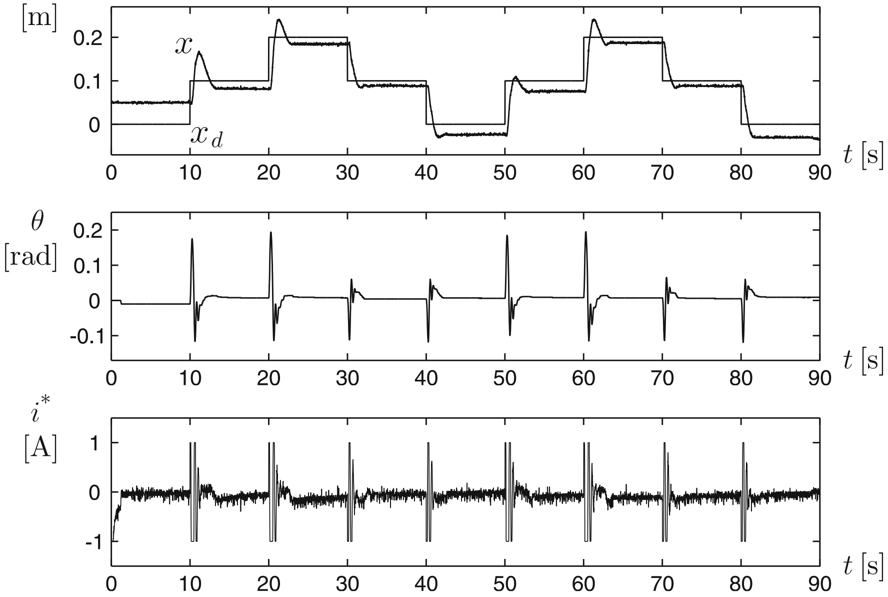


Fig. 14.10 Experimental results obtained with the prototype that has been built

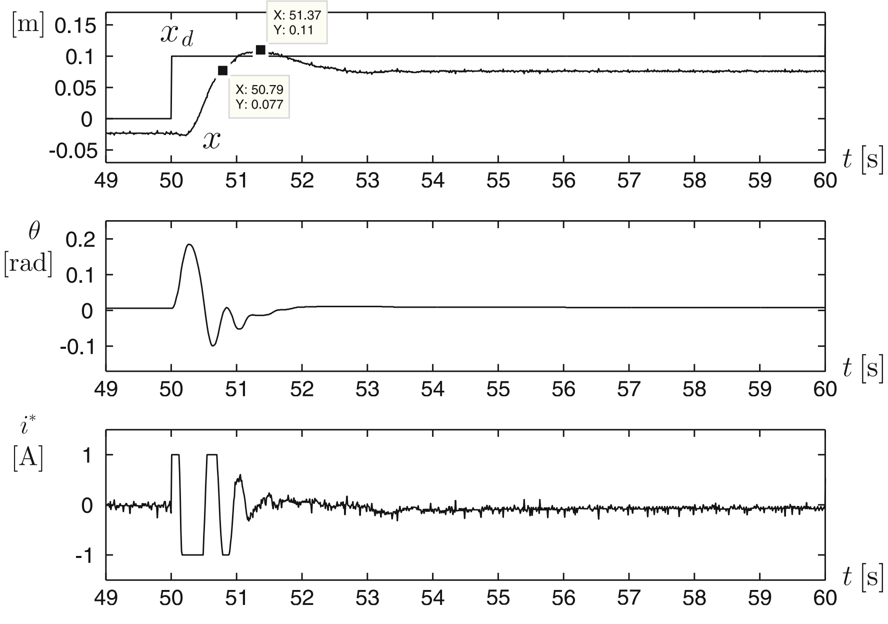


Fig. 14.11 Experimental results obtained with the prototype that has been built (continued)

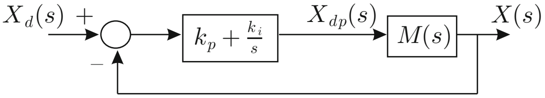


Fig. 14.12 Block diagram of a control scheme that includes a controller with integral action

model. Note that this torque disturbance becomes constant when the ball position is constant. Hence, a controller with an integral action is well suited to solving this problem and this is presented next. Designate by $M(s)$ the closed-loop function defined as:

$$M(s) = \frac{X(s)}{X_d(s)},$$

in the block diagram in Fig. 14.8b when all the parameters and gains take values according to the above design. In Fig. 14.12 a block diagram is presented that illustrates how an integral action is included in the control system. There, k_p and k_i are the gains of a PI controller that is included in the open-loop transfer function. Note that $X_{dp}(s)$ represents an intermediate variable that replaces $X_d(s)$ in all the previous block diagrams, whereas the desired ball position $X_d(s)$ is now shifted to the left to allow the PI controller to be driven by the ball position error $X_d(s)-X(s)$.

The controller gains $k_p = 0.5$ and $k_i = 1.5$ are selected to obtain a $\omega_1 = 1.98 \approx 1.71$[rad/s] crossover frequency and a $K_f = 45°$ phase margin, as depicted in Fig. 14.13. Recall that 1.71 is the crossover frequency selected in the previous design. In Fig. 14.14 a simulation response is presented when $X_d(s)$, in Fig. 14.12, is a unit step change.

Some experimental results are presented in Fig. 14.15 when the control scheme in Fig. 14.12 is used. Note that a zero steady- state error is reached now each time that a step change is commanded. Moreover, to further demonstrate the effectiveness of this approach, a constant disturbance is applied for approximately $t \geq 45$[s]. This disturbance is produced by adding a constant tilt angle to the mechanism base. This forces θ to be different from zero at a steady state to stabilize the ball at some constant position x. However, this increases the nonzero steady-state error in x if the controller does not possess an integral action. This is what happens in Fig. 14.16 where the ball position x stays far away from x_d once the disturbance appears, i.e., for $t \geq 45$[s]. In the case of the control scheme shown in Fig. 14.12 the integral action of the external loop controller achieves a zero steady-state error even when such a disturbance is applied, i.e., for $t \geq 45$[s], as can be seen in Fig. 14.15. This experimental test demonstrates the superiority of a control scheme with an integral action.

On the other hand, in Sect. 5.2.9, Chap. 5, the control scheme depicted in Fig. 5.53 was designed where:

$$\rho = 4.8, \;\; a = 0.98, \;\; k = 10.729, \;\; k_v = 0.35, \;\; \alpha = 30,$$

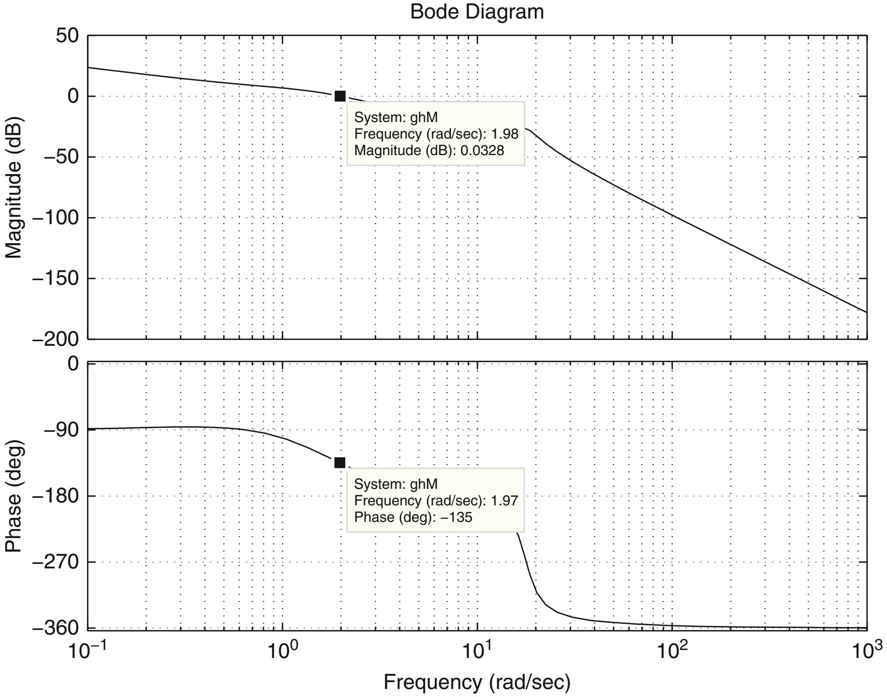


Fig. 14.13 Bode diagrams for the transfer function $\left(k_p + \frac{k_i}{s}\right) M(s)$, $k_p = 0.5$ and $k_i = 1.5$

$$A_\theta = A_x = 1, \;\; b = 1.4, \;\; c = 5.1138, \;\; \gamma = 2.1868.$$

This design was performed to achieve a rise time $t_r = 1$[s] and an overshoot $M_p = 25\%$. This control scheme is now tested experimentally using the ball and beam prototype described in this section. A step change in the desired position x_d is applied at $t = 0$, which changes from 0.05[m] to 0.15[m]. The corresponding experimental results are shown in Fig. 14.17, where $t_r = 1.169$[s] and $M_p = 14\%$ has been obtained. These results are considered to be satisfactory given the accuracy of the prototype that has been built. Note that, contrary to the previous experiments shown in the present section, the steady-state error is zero without the necessity of an integral controller. Recall that friction is responsible for such a steady-state error and friction is also uncertain and changes during normal operation. Hence, a zero steady-state error is sometimes accomplished, despite the presence of friction. Moreover, the effect of the ball weight can be reduced if the experiment is performed when the ball is close to the center of the beam (where the beam axis is located) as in Fig. 14.17. Finally, it is shown through simulations in Sect. 5.2.9, Chap. 5, that the controller gains $b = 2$, $c = 11.7121$, $\gamma = 4.6336$, result in a better time response. However, it was observed in experiments that use of these controller gains results in

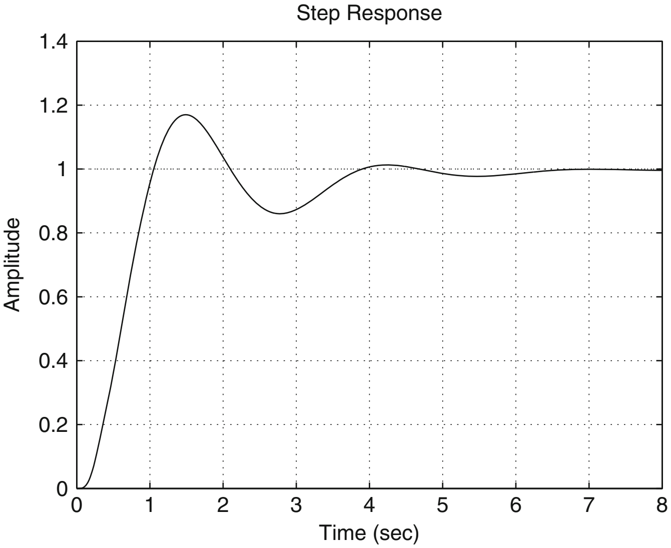


Fig. 14.14 Simulation response of the closed-loop system in Fig. 14.12, $k_p = 0.5$ and $k_i = 1.5$

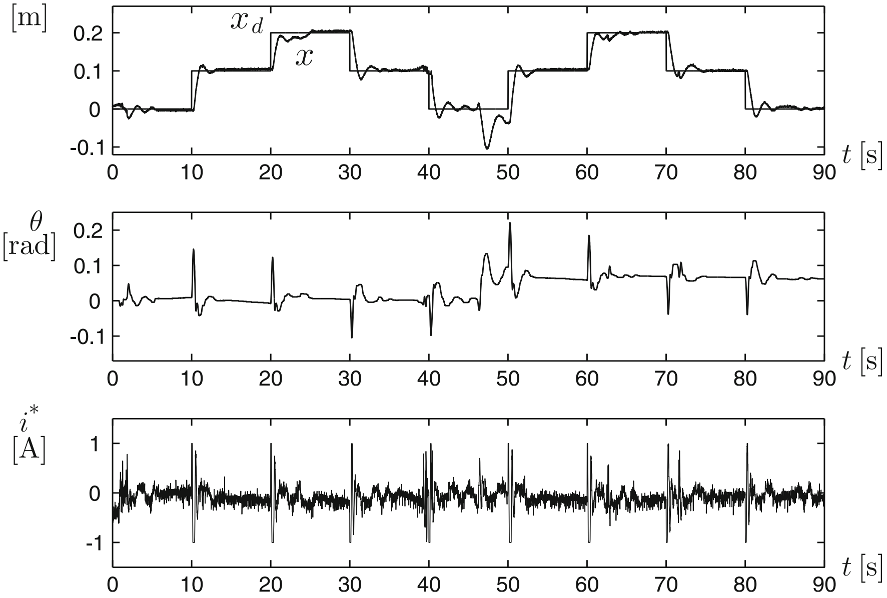


Fig. 14.15 Experimental response when the control scheme in Fig. 14.12 is used, $k_p = 0.5$ and $k_i = 1.5$. A constant disturbance exists for $t \geq 45$[s]

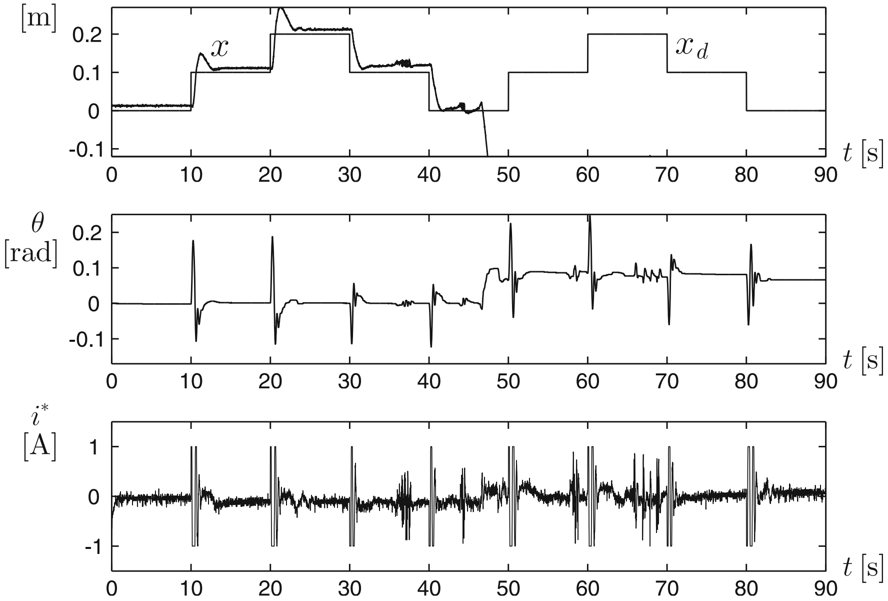


Fig. 14.16 Experimental response when the control scheme in Fig. 14.8 is used. A constant disturbance exists for $t \geq 45$[s]

noise amplification and the closed-loop control system has a bad performance. This is because these controller gains are large.

Finally, a picture of the ball and beam prototype that has been built is presented in Fig. 14.18.

14.6 Control System Electric Diagram

The control system electric diagram is shown in Fig. 14.19. This diagram is complemented with Fig. 14.3. A PIC16F877A Microchip microcontroller-based electronic board is employed as interface between the control program, implemented in a portable computer, and the ball and beam system. The communication is performed through the driver MAX232. This interface board receives pulses from the encoder and voltage delivered by the ball position x measurement system, and sends them to the portable computer. There, the control program handles these data numerically (which is explained later) to obtain the beam position θ in radians, and the ball position x in meters.

On the other hand, as the result of the control algorithm evaluation, the portable computer obtains the electric current that must flow though the DC motor and sends it back to interface board. There, the microcontroller sends such an electric current value to the power amplifier, which applies a suitable voltage at the motor

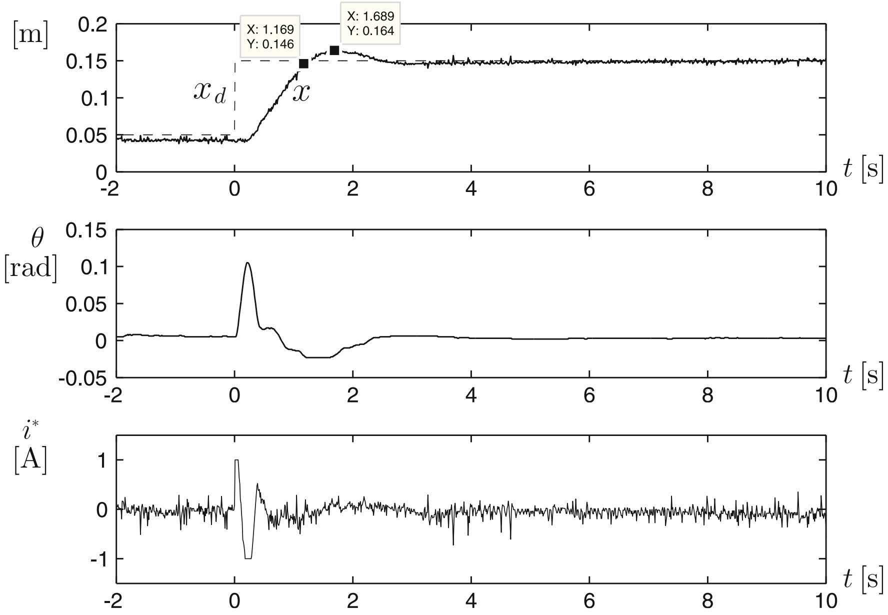


Fig. 14.17 Experimental response when the control scheme in Fig. 5.53 is used with $k_v = 0.35$, $\alpha = 30$, $b = 1.4$, $c = 5.1138$, $\gamma = 2.1868$

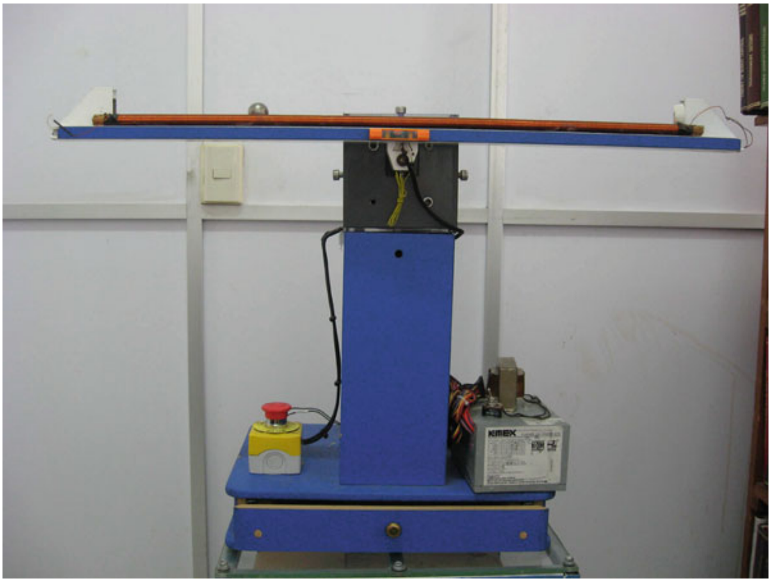


Fig. 14.18 Ball and beam system prototype that has been built

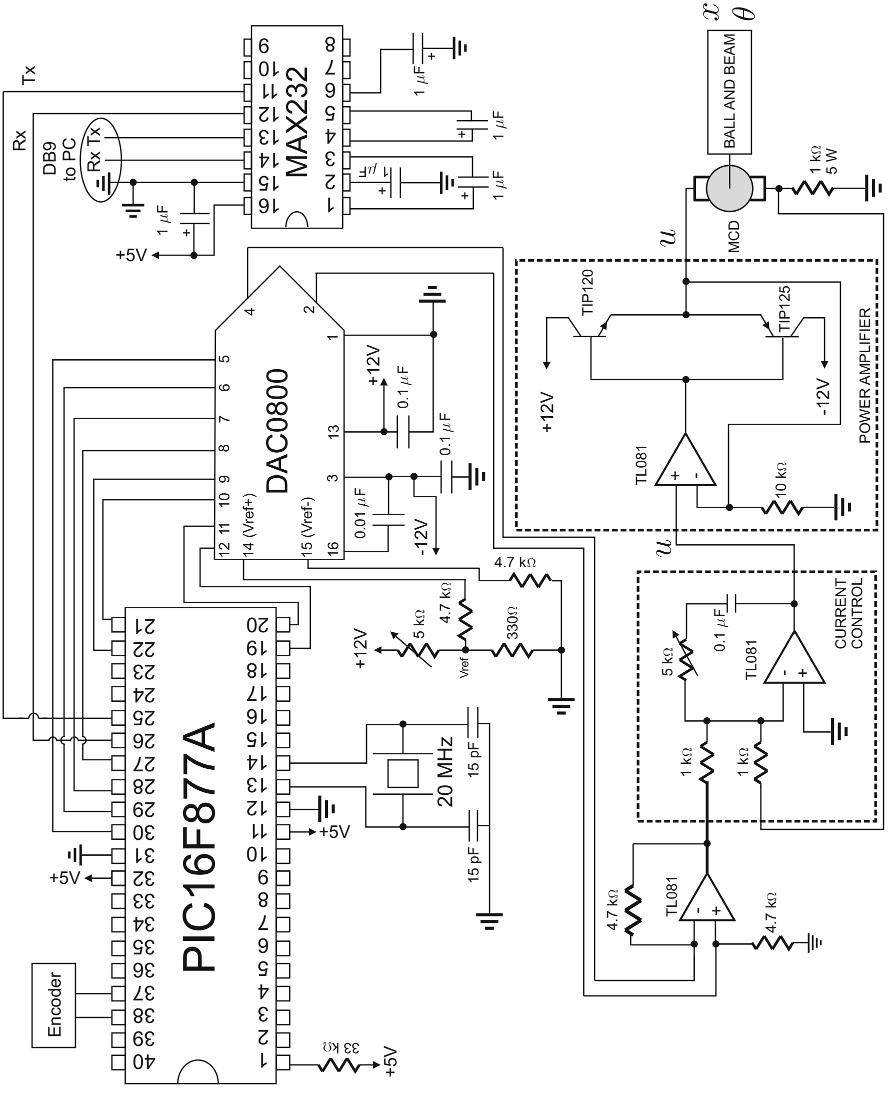


Fig. 14.19 Control system electric diagram

terminals, ensuring that such a desired current actually flows through the motor armature. To achieve this task, a DAC0800 digital/analog converter is employed which, working together with a TL081 operational amplifier, delivers $-i^*$ as an analog voltage signal. Then, an operational amplifier performs as a proportional–integral (PI) electric current controller with $\approx$ 5 as proportional gain and 10000 as integral gain. The voltage applied at the motor armature terminals, u, must be equal to the signal at the output of this PI controller, u_i. However, because of the dead zone

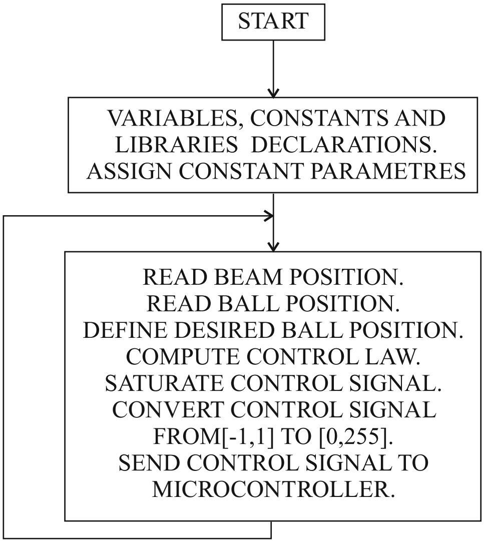

START
VARIABLES, CONSTANTS AND LIBRARIES DECLARATIONS. ASSIGN CONSTANT PARAMETRES
READ BEAM POSITION.
READ BALL POSITION.
DEFINE DESIRED BALL POSITION.
COMPUTE CONTROL LAW.
SATURATE CONTROL SIGNAL.
CONVERT CONTROL SIGNAL FROM[-1,1] TO [0,255].
SEND CONTROL SIGNAL TO MICROCONTROLLER.

Fig. 14.20 Flow diagram for PC programming when used to control a ball and beam system

nonlinearity produced by a couple of bipolar junction transistors (BJTs), feedback must be employed around such BJTs (see Sect. 9.1.2). The electric current through the motor armature circuit is measured using a 1[Ohm], 5[W], power resistance in series with the motor. The motor applies a torque to the ball and beam mechanism and the measurement systems described in Sects. 14.2.1 and 14.2.2 deliver to the microcontroller the positions of the beam and ball.

14.7 Builder 6 C++ Code Used to Implement the Control Algorithms

In this section we explain the main steps in implementing the control algorithms using the Builder 6 C++ code listed below. The corresponding flow diagram is shown in Fig. 14.20. Before defining some libraries and constants, the executable part of the program begins with the instruction “void ProcessByte(BYTE byte)”. There, the program remains waiting for the recognition code “0xAA” to be sent by the microcontroller PIC16F877A as an indicative that data from the mechanism is to be sent in the subsequent instructions. The beam position is first received and assigned to the variable “pos”, then the ball position is received and assigned to the variable “conv”. After that, the desired ball position is generated and assigned to the variable “xd”. The beam position in radians is assigned in the variable “theta” where “esctheta” is a factor converting pulses from encoder to radians using the relation:

$$2\pi[\text{rad}] \rightarrow 4 \times 400[\text{pulses/rev}]$$

theta[rad] → theta[number of pulses]/n.

$$\Rightarrow \qquad \text{theta[rad]} = \frac{2\pi\,\text{[rad]}}{4n \times 400\text{[pulses/rev]}}\text{theta[number of pulses]}.$$

Note that the gear box ratio n is taken into account as the encoder is fixed to the motor shaft. This allows us to consider that $A_\theta = 1$. The ball position in meters is assigned to the variable "x" where "escx= 0.7/1022" is a factor converting from conv $\in$ [0, 1022] (the code from a 10-bit analog/digital converter) to $x \in [-0.7/2, 0.7/2]$ in meters. Note that the subtraction "conv-511" and the fact that 0.7[m] is the beam length are important to performing this task. This allows us to consider that $A_x = 1$. This also means that in all the experiments presented until now, the ball position x is measured from the middle point of the beam, i.e., instead of using the symbol x it would be more correct to use the variable z (which was not done, however) defined as $z = x - \frac{L}{2}$ at the beginning of Sect. 14.1.1. This has not affected the correctness of the designs presented so far, because, as noted in the footnote above (14.5), the dynamical model of the ball and beam is identical when written in terms of either x or z.

Two controllers are coded: the first one corresponds to the control scheme in Fig. 14.12 and the second one corresponds to the control scheme in Fig. 14.8. The electric current to be commanded by these controllers is represented by the variable "iast". This variable is saturated so as not to take values out of the admissible range and its sign is changed to deliver $-i^*$ at the output of the digital/analog converter stage described in Sect. 14.6. Then "iast" is converted into the range [0, 255] to send it to the microcontroller as an 8-bit word. This is performed via the instruction "MainForm− >send_byte(cuentas)", some variables are saved in the file MONIT.txt, and time is increased by 0.01[s]. Finally, the program returns and remains waiting for the recognition code "0xAA" again.

```
//Ball and beam control computer program
//-----------------------------------
#include <vcl.h>
#pragma hdrstop
#include "Main.h"
#include <math.h>
#include <stdio.h>
#include <share.h>
#include <conio.h>
#include <dos.h>
//-----------------------------------
#pragma package(smart_init)
#pragma link "CSPIN"
#pragma resource "*.dfm"
//-------------------------------------
#define Ts 0.01 //sampling period in sec
#define pi_ 3.1416

#define gamma 1.6406 //0.5781
#define b 0.6404 //0.3745
```

```
#define c 4.5746 //2.6752
#define alfa 30.0
#define kv 0.35
#define kp2 0.5
#define ki2 1.5

#define maxI 2.0
#define IM 1.0
#define ppr 400.0
#define n 36.0
#define lriel 0.7
#define cADC 1022.0

FILE *ptrMonit;
TMainForm *MainForm;
unsigned char flagcom=0,flagfile=0,cuentas;
unsigned short int conv,pos; //de 16 bits
float t=0,inte=0,area=0,escs=255.0/(2.0*maxI);
float esctheta=pi_/(2*ppr*n),x,xd=0.0,theta,e,iast;
float v=0,thetad,theta_1=0,escx=lriel/cADC,iTs=1/Ts;
float ic=0.6,ee,k1,k2;
float integ=0.0,integ1=0.0,iast1,ese,ep,xd2;
//--------------------------------------------
void ProcessByte(BYTE byte)
{
if(flagcom!=0)
flagcom++;
if((byte==0xAA)&&(flagcom==0))
{
pos=0;
conv=0;
flagcom=1;
}
if(flagcom==2)
{
pos=byte;
pos=pos<<8;
}
if(flagcom==3)
pos=pos+byte;
if(flagcom==4)
{
conv=byte;
conv=conv<<8;
}
if(flagcom==5)
{
if(t>10){
xd=0.1;
}
if(t>20){
xd=0.2;
}
if(t>30){
```

```
xd=0.1;
}
if(t>40){
xd=0.0;
}
if(t>50){
xd=0.1;
}
if(t>60){
xd=0.2;
}
if(t>70){
xd=0.1;
}
if(t>80){
xd=0.0;
}
conv=conv+byte;
theta=(signed short int)pos;
theta=esctheta*theta;
x=(signed short int)(conv-511);
x=escx*x;
e=xd-x;
/*_______________Controllers___________________ */
/* ---------- Integral control ------*/
/* xd2=kp2*e+ki2*integ;
integ=integ+Ts*e;
ep=xd2-x;

thetad=gamma*(ep+v);//thetad=v1
v=(-c*v+(b-c)*ep)*Ts+v;
iast1=alfa*(thetad-theta)-kv*(theta-theta_1)*iTs;
iast=iast1;
theta_1=theta; */
/* ---------- Integral control ends ------*/
/* ---------- without integral control ------*/

thetad=gamma*(e+v);//thetad=v1
v=(-c*v+(b-c)*e)*Ts+v;
iast1=alfa*(thetad-theta)-kv*(theta-theta_1)*iTs;
iast=iast1;
theta_1=theta;
/* ---------- without integral control ends-----*/
/* _____________Controllers ends __________ */
/* ________Output saturation____________ */
if(iast>IM)
iast=IM;
if(iast<-IM)
iast=-IM;
/* ___________________________________________ */
MainForm->Edit3->Text = FloatToStr (iast);
iast=-iast;
cuentas=escs*(iast+maxI);
/* ___Physical constraint on the beam position___ */
```

```
if(theta>1.0)//0.5
cuentas=127;
if(theta<-1.0)// -0.5
cuentas=127;
/* ____________________________________________ */
MainForm->Edit1->Text = FloatToStr (xd);
MainForm->Edit2->Text = FloatToStr (x);
MainForm->Edit4->Text = FloatToStr (theta);
MainForm->Edit5->Text = FloatToStr (t);
MainForm->Edit6->Text = IntToStr (conv);
//Sending output byte
MainForm->Acknowledge();
MainForm->send_byte(cuentas);
flagcom=0;
/* open/close a file */
if(flagfile==0)
if((ptrMonit=fopen("MONIT.TXT", "w"))==NULL){}
flagfile=1;
/*Write to file*/
fprintf(ptrMonit,"%3.3f\t%3.3f\t%3.3f\t%3.3f\t%3.3f\n",t,xd,
x,-iast,theta);
t=t+Ts;
}
}
//---------------------------------------------
__fastcall TMainForm::TMainForm(TComponent* Owner)
: TForm(Owner), SerialPort(1, ProcessByte), fAcknowledge(true)
{
if (SerialPort.IsReady() != TRUE)
MessageBox(NULL, "Problems with port", "Error", MB_OK);
}
//---------------------------------------------
void TMainForm::send_byte(unsigned char byte_sal)
{
if (fAcknowledge == false)
return;
SerialPort.WriteByte(byte_sal);
fAcknowledge = false;
}
//----------------------------------------------
void TMainForm::Acknowledge()
{
fAcknowledge = true;
}
void __fastcall TMainForm::Button1Click(TObject *Sender)
{
/* close the file */
fclose(ptrMonit);
Close();
}
//---------------------------------------------
```

14.8 PIC C Code Used to Program the Microcontroller PIC16F877A

In this section, the main steps are described to program the microcontroller PIC16F877A used to exchange data between the computer and the mechanism. This program is written in PIC C language and is listed below. The corresponding flow diagram is presented in Fig. 14.21. After defining some libraries and constants, the interruption "int_rb" is used to count pulses from the encoder. The number of pulses are represented by the variable "cuenta". The executable part of the program begins at the instruction "void main(void)". Then, "TMR0=0" is set to start the first sample period, "cuenta=0" is set to fix $\theta = 0$ as the initial position of the beam (the beam must be horizontal at this position) and "PORTD=127" is set to fix to zero the initial commanded electric current, i.e., to maintain the motor without movement as the starting point. The instruction "while(TRUE)" defines an infinite cycle where the control task is to be implemented. The ball position is measured via an analog/digital converter at "conv=read_adc()". The instructions "putc(0xAA); putc(cuentaH); putc(cuentaL); putc(convH); putc(convL);" send the recognition code "0xAA" first, and then the beam position and the ball position, in that order. The instruction "while(TMR0<197)" defines a cycle where the microcontroller remains waiting for two events: (i) For the computer to send back the computed desired current, which is taken and sent to the digital/analog converter via the instructions "if(kbhit()) PORTD=getc();", (ii) That the sample period elapses (0.01[s]), which is true when

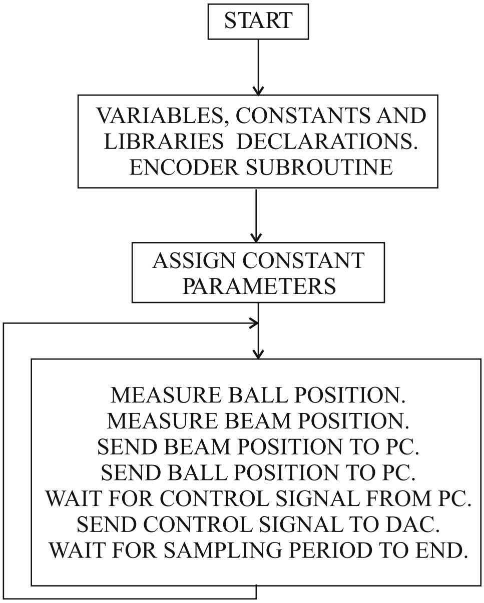

START
VARIABLES, CONSTANTS AND LIBRARIES DECLARATIONS. ENCODER SUBROUTINE
ASSIGN CONSTANT PARAMETERS
MEASURE BALL POSITION. MEASURE BEAM POSITION. SEND BEAM POSITION TO PC. SEND BALL POSITION TO PC. WAIT FOR CONTROL SIGNAL FROM PC. SEND CONTROL SIGNAL TO DAC. WAIT FOR SAMPLING PERIOD TO END.

Fig. 14.21 Flow diagram for microcontroller PIC16F877A programming when used as interface between a ball and beam system and a PC

the variable TMR0 is equal to or larger than 197. Once this happens, TMR=0 is set again and the program returns to the beginning of the infinite cycle “while(TRUE)”.

```
// PIC16F877A Microcontroller program
#include<16f877A.h>
#device adc=10 //adc, 10 bits
#include<stdlib.h>
#include<math.h>
#fuses HS,NOWDT,PUT,NOBROWNOUT,NOLVP,NOWRT,NOPROTECT,NOCPD
#use delay(clock=20000000)// (frequency Xtal)
//Config. serial port
#use rs232(baud=115200,XMIT=PIN_C6,RCV=PIN_C7,BITS=8,PARITY=N)
//ports and registers addresses
#byte OPTION= 0x81
#byte TMR0 = 0x01
#byte PORTA = 0x05
#byte PORTB = 0x06
#byte PORTC = 0x07
#byte PORTD = 0x08
#byte PORTE = 0x09

#bit PC0 = 0x07.0
#bit PC1 = 0x07.1
//------------Variables declaration ------------//
int16 inter,cuenta,conv;
int8 cuentaH,cuentaL,convH,convL,puerto,AB,AB_1,aux;
//int1 ban;
//------------Encoder interrupt subroutine --------//
#int_rb
void rb_isr()
{
puerto=PORTB;
AB=((puerto)&(0x30))>>4;
aux=AB^AB_1;
if(aux!=0)
if(aux!=3)
if(((AB_1<<1)^AB)&(0x02))
cuenta--;
else
cuenta++;
AB_1=AB;
}

//-----------main program-----------//
void main(void)
{
set_tris_a(0b11111111);
set_tris_b(0b11111111);
set_tris_c(0b10000000); //pin config.
set_tris_d(0b00000000);
set_tris_e(0b11111111);
OPTION=0x07; //pre_scaler timer0, 1:256
PORTC=0;
TMR0=0;
cuenta=0;
AB=0;
AB_1=0;
PORTD=127;
setup_adc(ADC_CLOCK_INTERNAL); //Config. ADC
setup_adc_ports(ANALOG_RA3_REF);//Config. reference ADC
set_adc_channel(1);
delay_ms(3000);
enable_interrupts(global);
enable_interrupts(int_rb);

while(TRUE)
{
PC0=1; //sending starts
conv=read_adc();
inter=(conv)&(0xFF00);
convH=inter>>8;
convL=(conv)&(0x00FF);

inter=(cuenta)&(0xFF00);
cuentaH=inter>>8;
cuentaL=(cuenta)&(0x00FF);

putc(0xAA); //serial port acknowledgment
putc(cuentaH); //sending to serial port
putc(cuentaL);
putc(convH);
putc(convL);


PC0=0; //sending ends
PC1=1; //waiting for sampling time
while(TMR0<197) //each count= (4/FXtal)*256 s
{
if(kbhit())//asking for a datum at buffer
PORTD=getc();
}
PC1=0; //sampling time is arrived
TMR0=0;
} //closing infinite while
} //closing main
```

14.9 Control Based on a PIC16F877A Microcontroller

14.9.1 Prototype Construction

In this section, a PIC16F877A microcontroller [5] is used to evaluate the control algorithm for the ball and beam system. Although an computer is also employed, this is only to save the important data and to plot them. It is also remarked that the

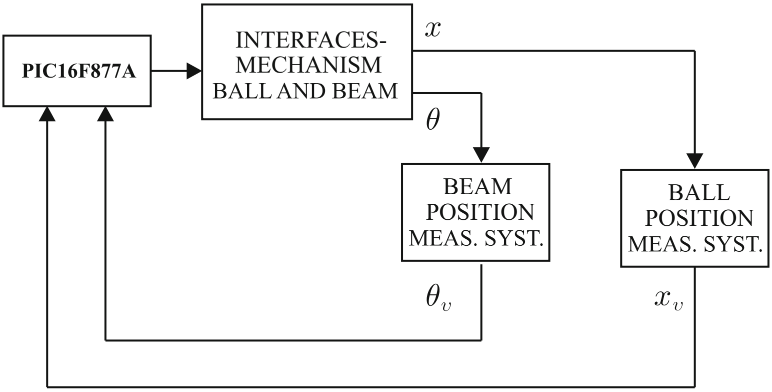

PIC16F877A
INTERFACES-MECHANISM BALL AND BEAM
x
θ
BEAM POSITION MEAS. SYST.
BALL POSITION MEAS. SYST.
θ_v
x_v

Fig. 14.22 Block interfaces and ball and beam mechanism are shown in Fig. 14.24. The beam position measurement system is composed of a potentiometer fixed to the beam shaft. The ball position measurement system is shown in Fig. 14.23

prototype mechanical part has also been slightly modified, which will be described in the following. A block diagram of the whole system that has been built is presented in Fig. 14.22.

The new ball position x measurement system[2] is shown in Fig. 14.23. It has the same components as the sensor in Fig. 14.3, but the method used to measure the ball position is different. In this case, rail number one is employed as an inductance at whose terminals a 16[KHz] sinusoidal voltage is applied. Hence, this AC voltage uniformly distributes along this coil and an AC voltage is obtained in rail number two whose peak value is proportional to the ball position x on the beam. Then, this signal is rectified and low-pass filtered by a RC circuit to minimize voltage ripple. The voltage x_v represents the ball position measurement, which is delivered to the microcontroller through one of its analog/digital converters. This sensor gain is computed as:

$$A_x = \frac{3.7625[\mathrm{V}]}{0.7[\mathrm{m}]} = 5.3750[\mathrm{V/m}], \tag{14.20}$$

where the numerator is x_v when the ball is at the right end of the beam and the denominator is the beam length.

On the other hand, the beam position θ measurement system is a potentiometer fixed to the beam shaft. The gain of this measurement system is computed as the division of a voltage increment at the potentiometer terminals, 0.275[V], and the corresponding angle increment θ, i.e., 0.3[rad]:

[2] In all the experiments that are presented in the remainder of this chapter, the ball position x is measured from the left end of the beam.

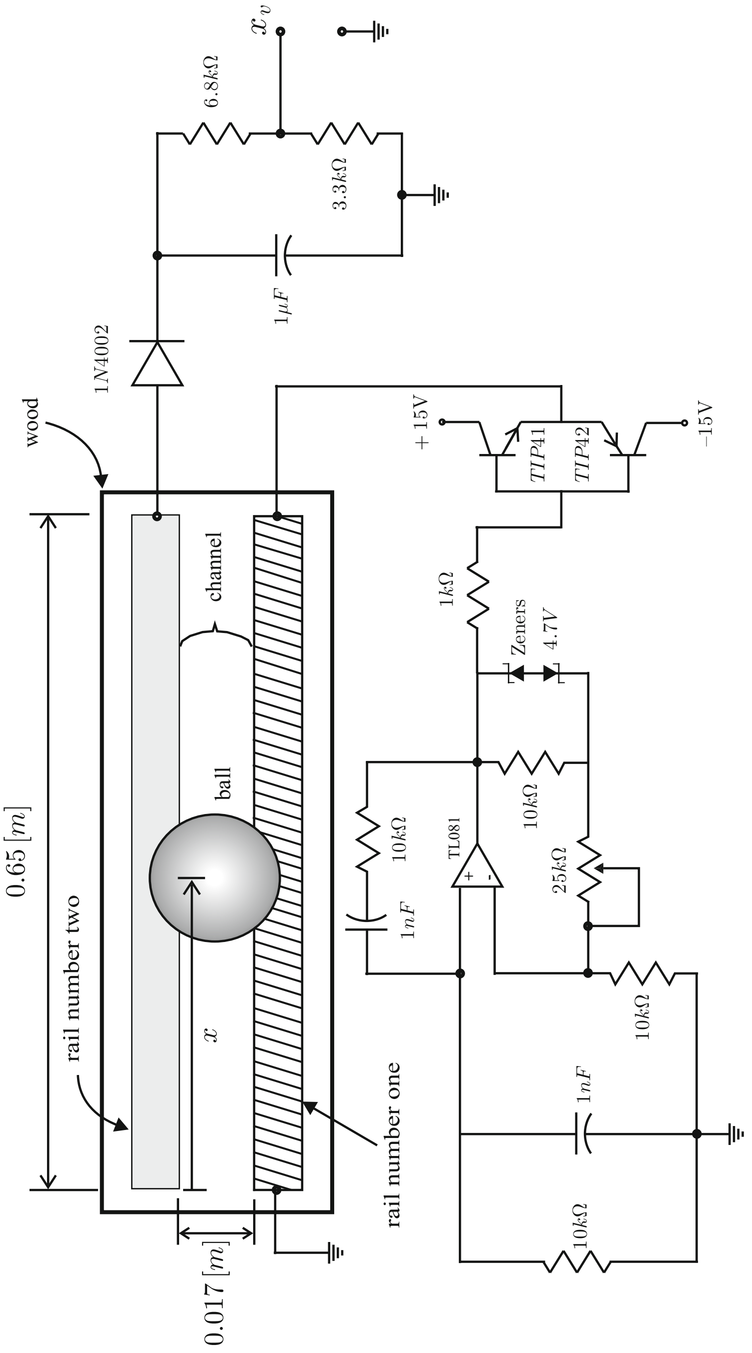


Fig. 14.23 Ball position x measurement system

$$A_\theta = \frac{0.275[\mathrm{V}]}{0.3[\mathrm{rad}]} = 0.9167[\mathrm{V/rad}]. \tag{14.21}$$

The following parameters:

$$k = 16.6035, \quad a = 3.3132, \quad \rho = 5, \tag{14.22}$$

were obtained using identical procedures to those described in Sects. 14.3.1 and 14.3.2. Note that these values are different from those shown in (14.12) and (14.15) mainly because a different gear box has been employed.

Finally, in Fig. 14.24 an electric diagram of the control system based on the PIC16F877A microcontroller is shown, whereas the code used to program the microcontroller is shown in Sect. 14.9.4. For a detailed explanation of Fig. 14.24 and the program in Sect. 14.9.4 the reader is advised to see Sects. 10.6 and 11.5 corresponding to the velocity and the position control of a PM brushed DC motor. It is stressed that the two channels of the analog/digital converter that possesses the microcontroller are employed: one is used to measure the ball position and the other to measure the beam position. Recall that the portable computer is only used to plot the important variables, but it is not used to evaluate the control algorithm.

14.9.2 Controller Design

It is proposed to employ a control scheme that is identical to that presented in Sect. 14.4, i.e., the closed-loop block diagram is identical to that shown in Fig. 14.8. The gain values:

$$\alpha = 12, \quad k_v = 0.2, \tag{14.23}$$

were selected experimentally such that using a constant value for $v_1(t)$ (see Fig. 14.8), i.e., without feeding back the ball position error and without using the lead compensator at the left of $V_1(s)$, a fast and well-damped θ response was obtained. Notice that $\alpha = 12$ shown in (14.23) is different from $\alpha = 30$ in (14.19). This because of the changes made in the mechanism. Using (14.20), (14.21), (14.22), (14.23) and the open-loop transfer function in (14.16) the root locus method is employed to find that the following gains ensure closed-loop stability:

$$\gamma = 1.2, \quad c = 20, \quad b = 2.5. \tag{14.24}$$

The detailed procedure followed to obtain these gains is presented in Sect. 5.2.8.

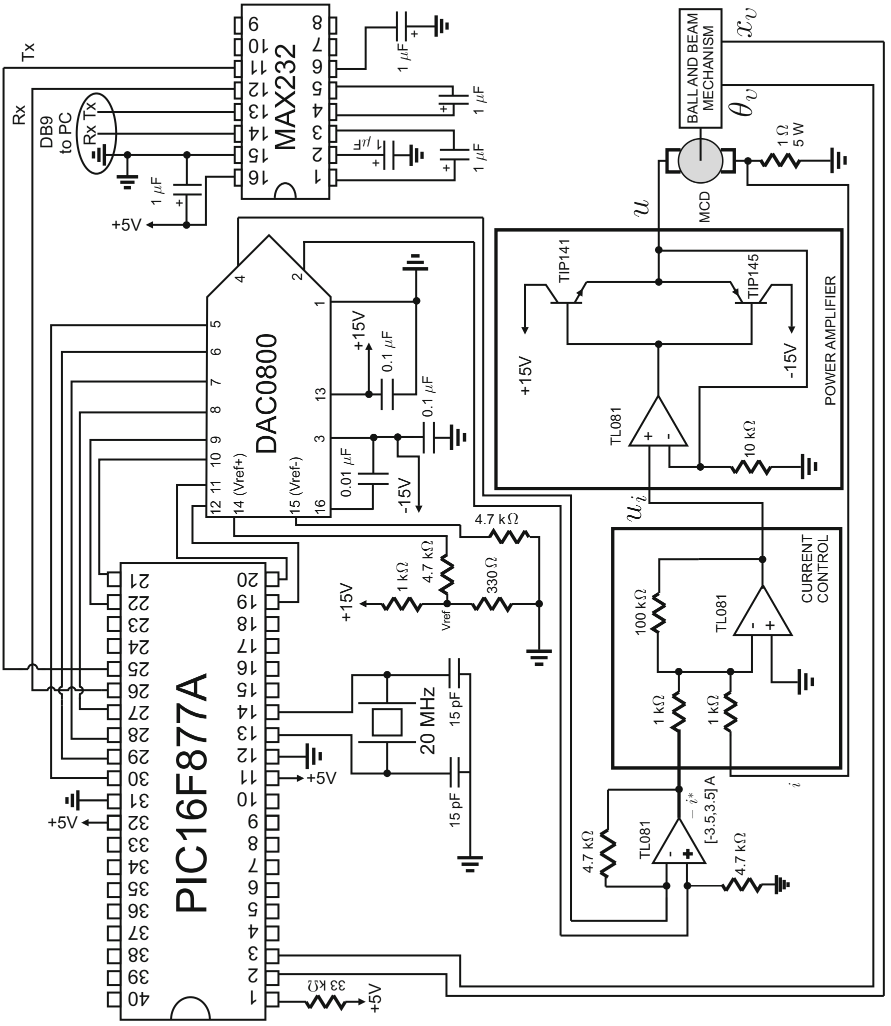


Fig. 14.24 Electric diagram of the ball and beam control system based on the PIC16F877A microcontroller

14.9.3 Experimental Results

The software implementation of the controller designed in the previous section is performed as explained in Sect. F.4, whereas the code used to program the microcontroller PIC16F877A [6] is shown in Sect. 14.9.4. The electric diagram in Fig. 14.24 is also used. Some experimental results obtained when using such a controller are shown in Fig. 14.25, i.e., using the gains in (14.23) and (14.24). It is observed that the ball reaches a position x_v close to 1[V], where $x_{vd} = 1$[V]

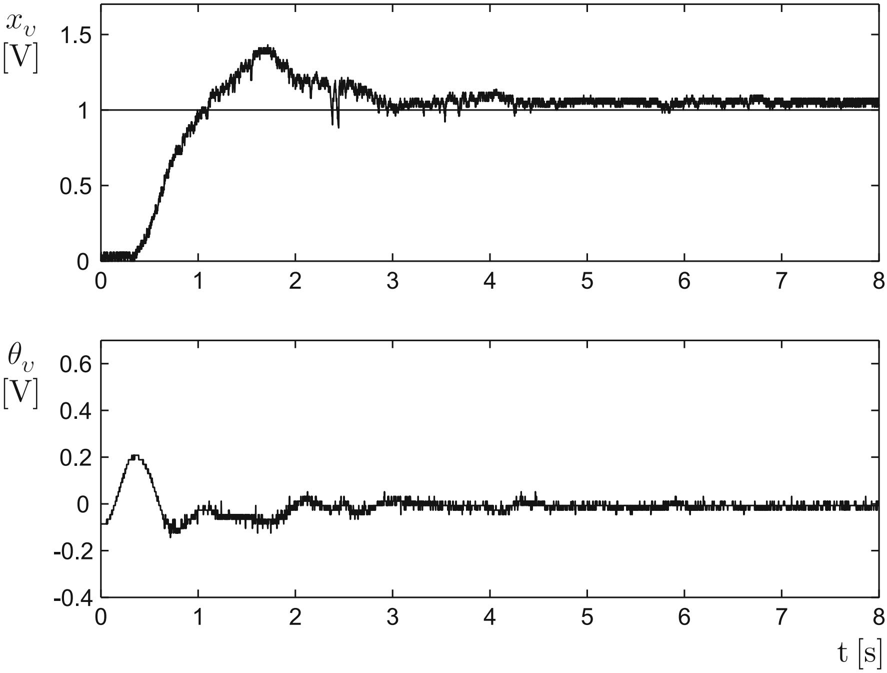


Fig. 14.25 Experimental results using microcontroller PIC16F877A

is the desired ball position. The small difference that is observed is due to the dead zone produced by friction at the motor shaft. In Fig. 14.26, some simulation results are presented using the same controller, the block diagram in Fig. 14.8 and the numerical values in (14.20), (14.21), (14.22), (14.23), (14.24).[3] Notice that the responses in Figs. 14.25 and 14.26 are very similar despite the noise content in the experimental prototype. Finally, in Fig. 14.27 some other experimental results are presented where it is observed that the ball position x_v is stabilized very close to its desired value $x_{vd} = 1$[V] despite an external agent hitting the ball several times. These results show that the control objective has been accomplished: the ball is stabilized very close to its desired position. Finally, a picture of the prototype used in the experiments is shown in Fig. 14.28.

14.9.4 PIC16F877A Microcontroller Programming

The flow diagram of the following code is presented in Fig. 14.29.

[3] See Sect. 5.2.8 for instructions on how these simulations can be performed.

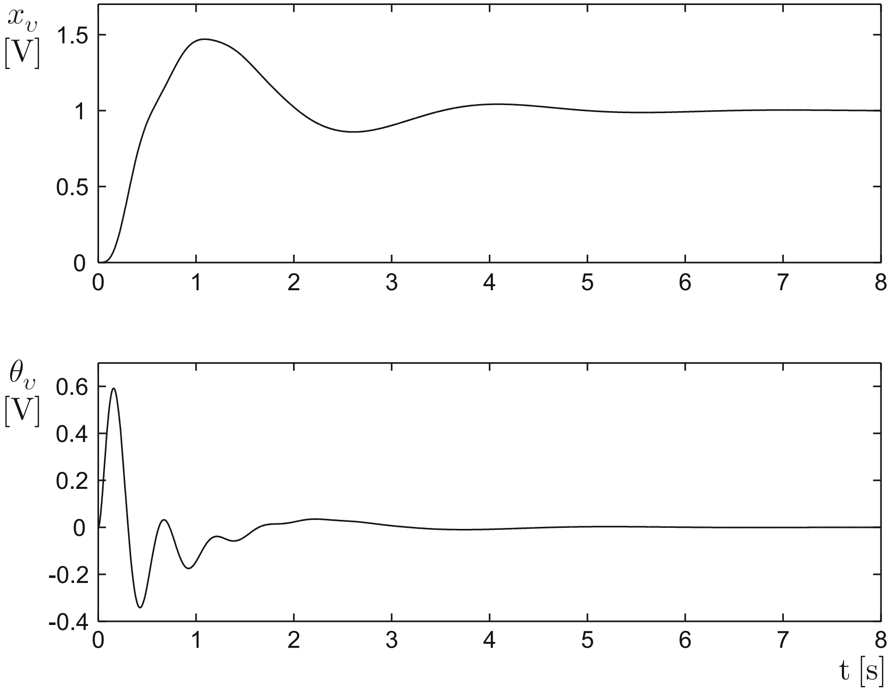


Fig. 14.26 Simulation results

```
#include<16f877A.h>
//#device adc=10 //adc, 10 bits
#include<stdlib.h>
#include<math.h>

#fuses HS,NOWDT,PUT,NOBROWNOUT,NOLVP,NOWRT,NOPROTECT,NOCPD

#use delay(clock=20000000) //Time base for delays
//(frequency Xtal)
//Config.
//P. Serial
#use rs232(baud=115200,XMIT=PIN_C6,RCV=PIN_C7,BITS=8,PARITY=N)
//port and registers addresses
#byte OPTION= 0x81
#byte TMR0 = 0x01
#byte PORTA = 0x05

#byte PORTB = 0x06
#byte PORTC = 0x07
#byte PORTD = 0x08

#byte PORTE = 0x09
#byte ADCON0= 0x1F
```

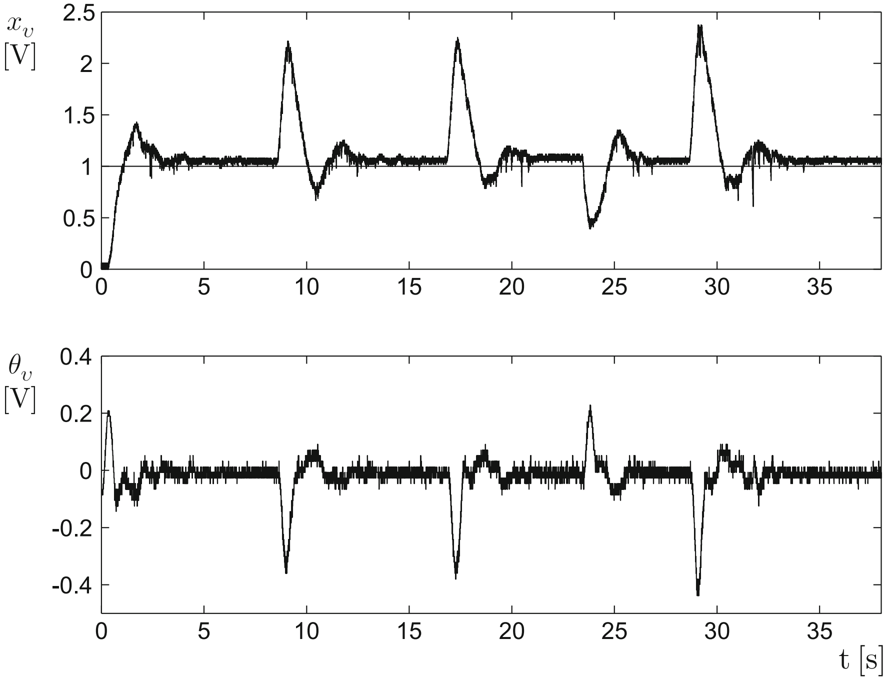


Fig. 14.27 Experimental results using the microcontroller PIC16F877A. Some ball position deviations are introduced by an external agent

```
#byte ADCON1= 0x9F

#bit PC0 = 0x07.0
#bit PC1 = 0x07.1
//------------variables declaration------------//
int16 inter,cuenta;

int8 cuentaH,cuentaL,puerto,AB,AB_1,aux;

int8 cont,iastd,cont2,cont3,cont4,t_0,t_1;

float x,xd,theta,error,iast,tiempo,c,b;

float gamma,alfa,kv,v,v1,thetam1; unsigned int u,i;
//int1 ban;
//------------encoder interrupt subroutine-------//
#int_rb void rb_isr() {
puerto=PORTB;
AB=((puerto)&(0x30))>>4;
aux=AB^AB_1;
if(aux!=0)
if(aux!=3)
if(((AB_1<<1)^AB)&(0x02))
```

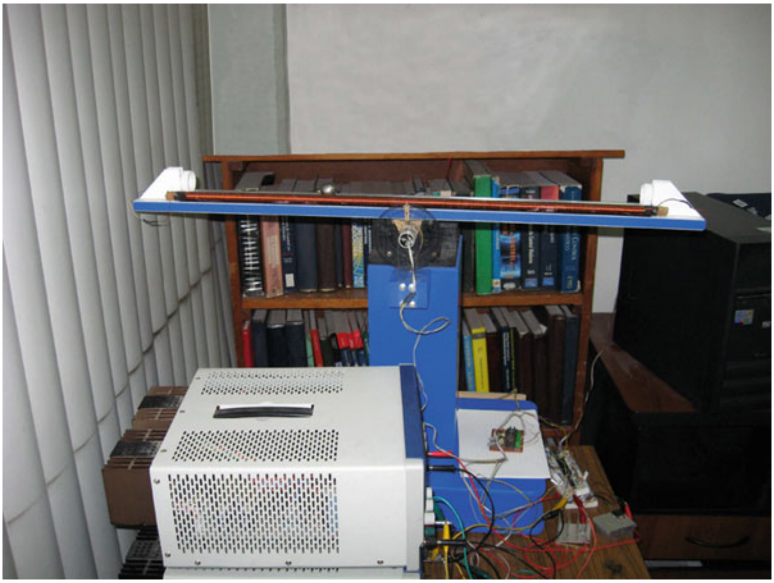
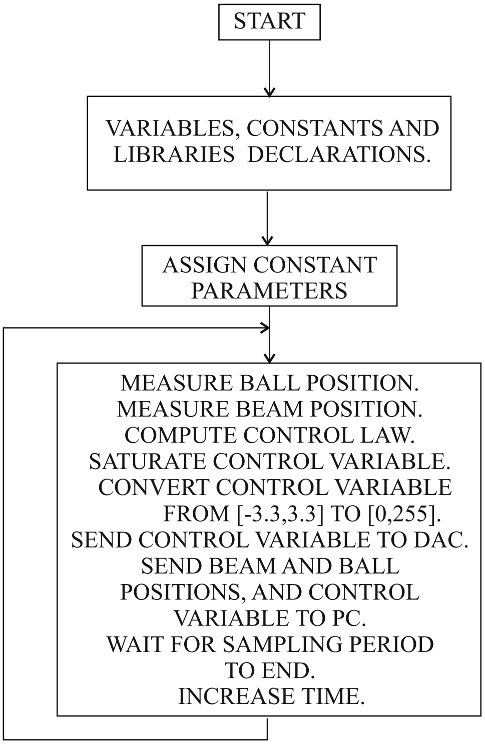


Fig. 14.28 Ball and beam system prototype used in the experiments

```
cuenta--;
else
cuenta++;
AB_1=AB;
}
//------------Main program------------//
void main(void) {
setup_adc(ADC_CLOCK_INTERNAL ); //ADC (internal clock)
set_tris_a(0b11111111);
set_tris_b(0b11111111);
set_tris_c(0b10000000); //serial communication
set_tris_d(0b00000000);
set_tris_e(0b11111111);
OPTION=0x07; //pre_scaler timer0, 1:256
PORTC=0;
TMR0=0;
cuenta=0;
AB=0;
AB_1=0;
enable_interrupts(global);
enable_interrupts(int_rb);
cont=0;
cont2=0;
cont3=0;
cont4=0;
PORTD=127;
xd=1.0;
tiempo=0.0;
```

Fig. 14.29 Flow diagram for PIC16F877A microcontroller programming


START
VARIABLES, CONSTANTS AND LIBRARIES DECLARATIONS.
ASSIGN CONSTANT PARAMETERS
MEASURE BALL POSITION.
MEASURE BEAM POSITION.
COMPUTE CONTROL LAW.
SATURATE CONTROL VARIABLE.
CONVERT CONTROL VARIABLE FROM [-3.3,3.3] TO [0,255].
SEND CONTROL VARIABLE TO DAC.
SEND BEAM AND BALL POSITIONS, AND CONTROL VARIABLE TO PC.
WAIT FOR SAMPLING PERIOD TO END.
INCREASE TIME.

```
gamma=1.2;
c=20.0;
b=2.5;
alfa=12.0;
kv=0.2;
thetam1=0.0;
v=0.0;
ADCON1=0x04;
ADCON0=0x81;
while(cont2<3)
{
while(cont3<255)
{
TMR0=0;
while(TMR0<255)
{
}
cont3++;
}
cont2++;
}
TMR0=0;
while(TRUE)
```

```
{
ADCON0=0x81; //Changing to CH0
for(i=0;i<37;i++); /stabilizing internal capacitor
t_0=read_adc(); //reading ADC
x=0.0196*t_0;

ADCON0=0x89; //changing to CH1
for(i=0;i<37;i++); //stabilizing internal capacitor
t_1=read_adc(); //reading ADC
theta=0.0196*t_1-2.81;//2.81;

error=xd-x;
/* Controller */
v1=gamma*(error+v);
v=(-c*v+(b-c)*error)*0.002+v;
iast=alfa*(v1-theta)-kv*(theta-thetam1)/0.002;
/* ___________________________________________ */
if(iast<-3.3)
iast=-3.3;
if(iast>3.3)
iast=3.3;
iast=-iast;
iastd=(unsigned int)(36.4286*iast+127);
PORTD=iastd;
thetam1=theta;

cuentaH=t_1;
cuentaL=t_0;
putc(0xAA); //serial port acknowledgment
putc(cuentaH); //sending to serial port
putc(cuentaL);
/*
iast=-iast;
iastd=(unsigned int)(36.4286*iast+127);
cuentaH=0x00;
cuentaL=(iastd)&(0xFF);
putc(0xAA); //serial port acknowledgment
putc(cuentaH); //sending to serial port
putc(cuentaL);
*/
PC1=1; //waiting for sampling time
while(TMR0<40) //1 count=(4/FXtal)*256 s, 20=1 ms
{
}
PC1=0; //sampling time is arrived
TMR0=0;
tiempo=tiempo+0.002;
} //closing infinite while
} //closing main
```

14.10 Summary

In this chapter, a ball and beam system has been built in addition to all the necessary interfaces for closed-loop control. The proposed control scheme has two loops. The internal loop is useful to control the beam angle, whereas the external loop is used to control the ball position. An interesting feature of the prototype that has been built is the construction of the sensor used to measure the ball position. The main idea is to build a channel composed of two rails which, together, work in a similar manner to a potentiometer. Hence, the voltage delivered by this measurement system is proportional to the position of the ball on the beam.

The controller design has been performed using both fundamental approaches in classical control: time response (root locus) and frequency response (Nyquist stability criterion and phase margin). The controllers that have been designed have been tested through several experiments to observe the achieved performance. Also, some interesting experiments have been performed to identify the system parameters.

14.11 Review Questions

1. Why is the ball and beam system open loop unstable?
2. How do the measurement systems designed for the ball position work?
3. Give a descriptive explanation of why it is necessary to control the beam angle aside from the ball position.
4. What would happen with ρ if $r = 0$ (see (14.6)) and what does this mean?
5. Recall the experimental procedure described in Sect. 14.3.2 to identify the only parameter in the ball dynamics. What else can you do to identify this parameter?
6. The automatic navigation control system of a ship is similar to a ball and beam system, where the role of the beam is replaced by the rudder. Using this idea try to control the direction of a ship.
7. Why does a lead compensator have fewer noise problems than a PD controller?
8. Why do you think it is preferable to use a proportional controller with velocity feedback instead of a PD controller for the beam angle?

References

1. Quanser Inc. 119 Spy Court Markham, Ontario L3R 5H6, Canada. Available at: http://www.quanser.com
2. P. V. Kokotovic, The joy of feedback: nonlinear and adaptive (1991 Bode Prize Lecture), *IEEE Control Systems Magazine*, pp. 7–17, June 1992.
3. R. C. Hibbeler, *Engineering Mechanics Dynamics*, 10th. Edition, Prentice Hall, Upper Saddle River, NJ, 2004.

4. M. Alonso and E. J. Finn, *Physics*, Addison-Wesley, Reading, MA, 1992.
5. PIC16F877A Enhanced Flash Microcontroller, Data sheet, Microchip Technology Inc., 2003.
6. Custom Computer Services Incorporated, CCS C Compiler Reference Manual, 2003.