

Chapter 16

Control of an Inertia Wheel Pendulum



16.1 Inertia Wheel Pendulum Description

In this chapter, a control strategy is designed for a mechanism known as the inertia wheel pendulum (IWP). A picture of the IWP is shown in Fig. 16.1 when operating at the downward position, whereas the variables and the parameters involved are defined in Fig. 16.2. It is a pendulum with a PM brushed DC motor fixed at its free end, which actuates on a wheel. The pendulum hangs up from a point where no external torque is applied. It is said that this point is not actuated but the wheel is actuated because it receives torque generated by the motor. As the pendulum and the wheel define two-degrees-of-freedom [1] but there is only one actuator (at the wheel), the mechanism is said to be underactuated [2].

The IWP was introduced by Spong et al. [2], and aside from its academic applications it is also widely used for research purposes as a workbench for nonlinear controllers [2–11].

The nomenclature employed is the following:

- q_1 is the pendulum angular position.
- q_2 is the wheel angular position.
- m_1 is the pendulum mass.
- m_2 is the wheel mass.
- τ is torque applied at the wheel.
- l_{c1} is the location of the pendulum's center of mass.
- l_1 is the pendulum's length.
- I_1 is the pendulum' inertia when rotating around its center of mass.
- I_2 is the wheel inertia (plus the motor rotor inertia).
- g is the gravity acceleration.
- R is the motor armature resistance.
- L is the motor armature inductance.

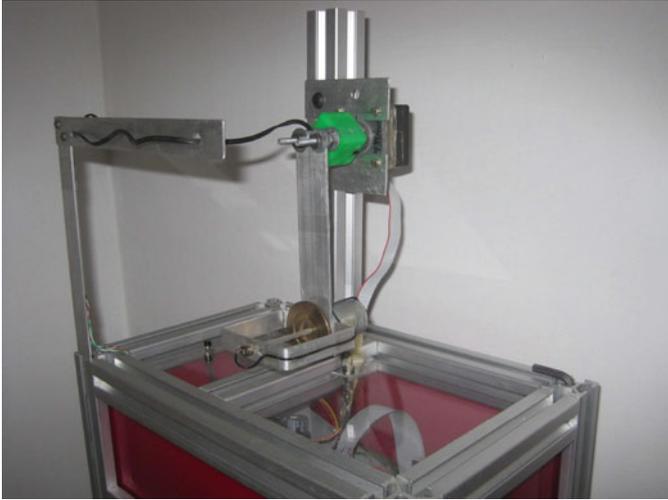
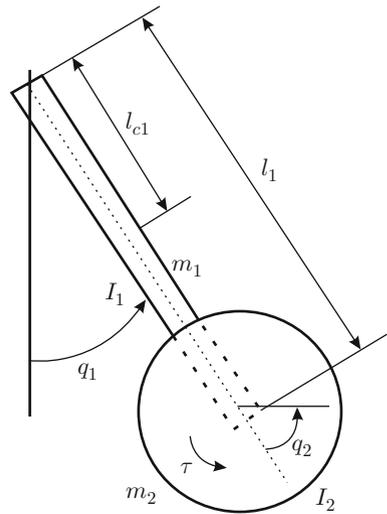


Fig. 16.1 The inertia wheel pendulum at the noninverted position

Fig. 16.2 Inertia wheel pendulum



- k_b and k_m are the motor counter electromotive force constant and the motor torque constant.
- u and i stand respectively, for voltage applied at the motor armature terminals and the electric current through the motor armature.

According to the above description of the IWP, it is clear that it is not easy to take the pendulum directly from $q_1 = 0, \dot{q}_1 = 0$, to $q_1 = +\pi$, or $q_1 = -\pi$, and $\dot{q}_1 = 0$. Moreover, the latter configuration is unstable and it is known as *the inverted unstable configuration*. Thus, it is of interest to design a control strategy such that:

(i) The pendulum is taken from a vicinity of $q_1 = 0, \dot{q}_1 = 0$, to the inverted unstable configuration, (ii) The pendulum stays at the inverted unstable configuration. The first part of this task is known as the *pendulum swing up* and the second part is known as the *pendulum balancing*.

In this chapter it is shown that solution of the swing up problem requires the use of nonlinear control techniques because the IWP has a nonlinear model. Although nonlinear, the model of this mechanism is straightforward enough to design a control strategy using simple ideas. One important objective in including this control problem is to introduce the reader, using a basic example, to nonlinear control systems techniques.

16.2 Mathematical Model

As the inertia wheel pendulum is a mechanism composed of two interacting bodies, its mathematical model is obtained using the *Euler–Lagrange equations*, i.e.,

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} &= 0, \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2} &= \tau. \end{aligned} \quad (16.1)$$

The zero on the right-hand side in the first expression indicates that no external torque is applied at the point where the pendulum hangs up. The *Lagrangian* is given as:

$$\begin{aligned} L &= K - P, \\ K &= K_1 + K_2, \quad P = P_1 + P_2, \end{aligned} \quad (16.2)$$

where K_1 and K_2 represent the kinetic energies of the pendulum and the wheel respectively, whereas P_1 and P_2 represent the potential energies of the pendulum and the wheel respectively. As the pendulum is a body with a mass distributed along a stick, its kinetic energy is computed by adding the kinetic energy of a particle of mass m_1 , which translates on a circumference with radius l_{c1} and the kinetic energy of a stick that rotates around its center of mass, i.e.,:

$$K_1 = \frac{1}{2} m_1 l_{c1}^2 \dot{q}_1^2 + \frac{1}{2} I_1 \dot{q}_1^2.$$

The kinetic energy of the wheel is obtained in a similar manner. However, it must be taken into account that the wheel's angular velocity, as measured by an observer fixed to the laboratory, is $\dot{q}_1 + \dot{q}_2$. Then, the following is found:

$$K_2 = \frac{1}{2}m_2l_1^2\dot{q}_1^2 + \frac{1}{2}I_2(\dot{q}_1 + \dot{q}_2)^2.$$

To compute the potential energy, the configuration where $q_1 = 0$ is used as reference, i.e., where $P_1 = 0$ and $P_2 = 0$. Recall that the pendulum *potential energy* is given as the product of force applied to the pendulum, i.e., the pendulum weight, and the component parallel to the pendulum weight of the distance between the pendulum's center of mass at the actual position q_1 and position of the pendulum's center of mass where the pendulum's potential energy is zero, i.e., when $q_1 = 0$. Then:

$$P_1 = gm_1l_{c1}(1 - \cos(q_1)).$$

Proceeding in a similar manner, it is found that the wheel's potential energy is given as:

$$P_2 = gm_2l_1(1 - \cos(q_1)).$$

Hence, according to (16.2):

$$L = \frac{1}{2}m_1l_{c1}^2\dot{q}_1^2 + \frac{1}{2}I_1\dot{q}_1^2 + \frac{1}{2}m_2l_1^2\dot{q}_1^2 + \frac{1}{2}I_2(\dot{q}_1 + \dot{q}_2)^2 - g\bar{m}(1 - \cos(q_1)),$$

where $\bar{m} = m_1l_{c1} + m_2l_1$. Replacing L in the Euler–Lagrange equations (16.1) and after suitably arranging the resulting terms, the following is found:

$$D \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} \bar{m}g \sin(q_1) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \tau \end{bmatrix}, \quad (16.3)$$

$$D = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} = \begin{bmatrix} m_1l_{c1}^2 + m_2l_1^2 + I_1 + I_2 & I_2 \\ & I_2 \end{bmatrix}.$$

It is simple to verify that the determinant of matrix D is positive, i.e.,:

$$\det(D) = d_{11}d_{22} - d_{12}d_{21} > 0. \quad (16.4)$$

The model in (16.3) is the most complete one for the IWP and it will be used in the next section to design a control strategy to swing up the pendulum to the inverted unstable configuration.

Voltage applied at the armature terminals of the PM brushed DC motor actuating on the wheel is the actual control signal, i.e., the variable that can be directly manipulated, whereas torque at the wheel is a variable that is a consequence of the application of such a voltage. However, in practice it is possible to assume that torque at the wheel is the control signal if it is supposed that the armature inductance is small. The armature circuit model of a PM brushed DC motor is the following:

$$L_a \frac{di}{dt} + Ri + k_b \dot{q}_2 = u. \quad (16.5)$$

If L_a and k_b are assumed to be zero, then:

$$\tau = k_m i = \frac{k_m}{R} u. \quad (16.6)$$

This means that the applied voltage has a direct effect on the generated torque; hence, it is reasonable to assume that torque is the control signal. Assuming k_b to be zero is justified as follows. It is observed from (16.5) that k_b introduces additional viscous friction to the mechanism. As this commonly improves stability in a control system, then assuming $k_b = 0$ renders the control problem more interesting at hand in the sense that the design is performed by assuming that any stabilizing effect is not present naturally in the mechanism.

16.3 Swing Up Nonlinear Control

A controller is designed in this section to take the pendulum to its inverted unstable configuration, i.e., to any of the points $(q_1, \dot{q}_1) = (+\pi, 0)$ or $(q_1, \dot{q}_1) = (-\pi, 0)$. It will be explained later why the wheel position and velocity are not taken into account during the time interval when this task is performed.

Solving the first row in (16.3) for \ddot{q}_2 , replacing this value in the second row of (16.3), and using (16.6) in addition to $d_{12} = d_{22}$, the following is found:

$$J\ddot{q}_1 + mgl \sin(q_1) = \tau_1. \quad (16.7)$$

$$J = \frac{d_{22}d_{11} - d_{21}d_{12}}{d_{12}}, \quad mgl = \bar{m}g, \quad \tau_1 = -\frac{k_m}{R}u.$$

Notice that $J > 0$ because of (16.4). Hence, the model in (16.7) corresponds to a simple pendulum where J , m , l , and g represent the equivalent inertia, mass, length, and gravity acceleration respectively. The total energy of this pendulum is given by adding its *kinetic energy* and potential energy, i.e.,:

$$V(q_1, \dot{q}_1) = \frac{1}{2}J\dot{q}_1^2 + mgl(1 - \cos(q_1)). \quad (16.8)$$

The *level surfaces* of V are depicted in Fig. 16.3. This means that each curve in this figure represents the set of points (q_1, \dot{q}_1) where $V(q_1, \dot{q}_1)$ takes a constant value. The reader should realize that the more external curves correspond to larger values of $V > 0$ and the internal-most curve corresponds to a single point, $(q_1, \dot{q}_1) = (0, 0)$, where $V = 0$ takes its smallest value.

Time derivative of $V(q_1, \dot{q}_1)$, given in (16.8), is found to be:

$$\frac{dV(q_1, \dot{q}_1)}{dt} = \dot{q}_1 J \ddot{q}_1 + mgl \dot{q}_1 \sin(q_1).$$

If $J \ddot{q}_1$ is replaced from (16.7), the following is found:

$$\frac{dV(q_1, \dot{q}_1)}{dt} = \dot{q}_1 \tau_1. \tag{16.9}$$

What has been computed is known as the *time derivative of V along the trajectories of (16.7)* [12]. This means that (16.9) represents the values that $\frac{d}{dt} V$ takes as the pendulum in (16.7) moves under the effect of the external torque τ_1 . For instance, if $\tau_1 = 0$ then $\frac{d}{dt} V = 0$, according to (16.9), which means that the pendulum’s energy V remains constant as the pendulum moves. This is very important for our purposes. From the knowledge of the pendulum’s initial position and velocity $(q_1(0), \dot{q}_1(0))$, it is possible to know the initial value of the pendulum’s energy $V(q_1(0), \dot{q}_1(0))$. As $\frac{d}{dt} V = 0$ because $\tau_1 = 0$ then, as time grows, the pendulum moves only on points representing the level surface where $V(q_1(t), \dot{q}_1(t)) = V(q_1(0), \dot{q}_1(0))$, which are shown in Fig. 16.3. Arrows on the level surfaces represent the sense of the pendulum’s movement. These senses of movement are easy to determine. The state vector of a simple pendulum is defined

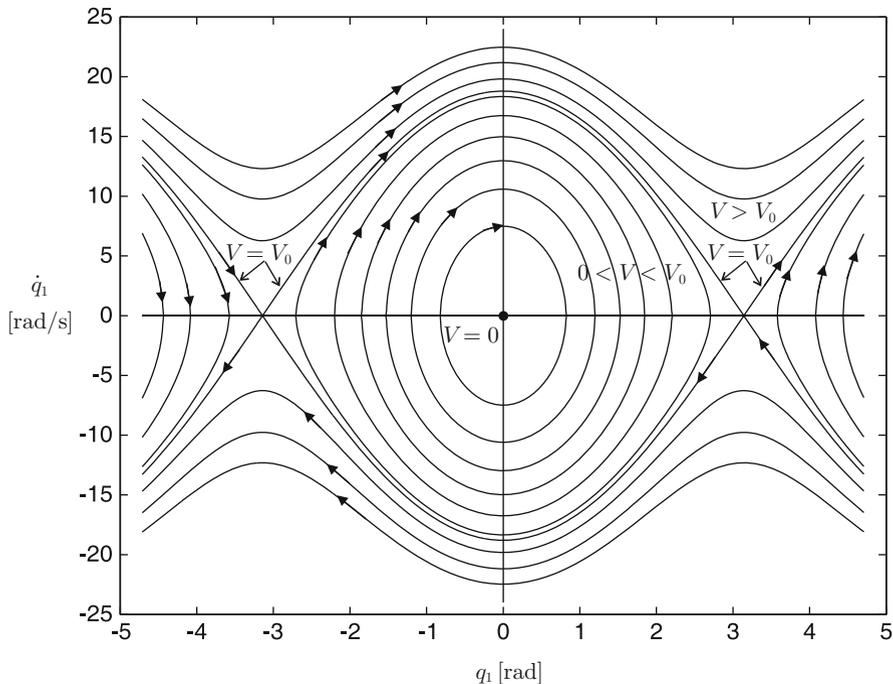


Fig. 16.3 Level surfaces of V in (16.8)

as $y = [q_1, \dot{q}]^T$. The variable $\dot{y} = [\dot{q}_1, \ddot{q}]^T$ is a vector indicating where the state y moves to. Notice that the horizontal component of \dot{y} , i.e., \dot{q}_1 , points to the right as long as the pendulum velocity \dot{q}_1 is positive. This suffices to determine the sense of the arrows on the level surfaces of V .

According to the ideas discussed above, to swing up the pendulum it suffices to increase the pendulum energy to:

$$V(q_1, \dot{q}_1)|_{\substack{q_1=+\pi \\ \dot{q}_1=0}} \text{ or } q_1=-\pi = \frac{1}{2}J(0)^2 + mgl(1 - \cos(\pm\pi)) = 2mgl = V_0,$$

and, in the case that such an energy is reached at a point where $(q_1, \dot{q}_1) \neq (+\pi, 0)$ or $(q_1, \dot{q}_1) \neq (-\pi, 0)$, then apply a torque τ_1 that ensures $\frac{d}{dt}V = 0$ from that time on. Notice that, according to Fig. 16.3, if V is kept constant once $V = V_0$, then the pendulum continues to move until one of the points $(q_1, \dot{q}_1) = (+\pi, 0)$ or $(q_1, \dot{q}_1) = (-\pi, 0)$ is reached. It is shown next that the above is ensured if the following torque is used:

$$\tau_1 = -k_d \text{sat}(\dot{q}_1) \text{sign}(V - V_0), \quad (16.10)$$

$$\text{sat}(\dot{q}_1) = \begin{cases} \varepsilon_0, & \dot{q}_1 > \varepsilon_0 \\ -\varepsilon_0, & \dot{q}_1 < -\varepsilon_0 \\ \dot{q}_1, & |\dot{q}_1| \leq \varepsilon_0 \end{cases}, \quad \text{sign}(V - V_0) = \begin{cases} +1, & V > V_0 \\ -1, & V < V_0 \\ 0, & V = V_0 \end{cases}, \quad (16.11)$$

where ε_0 and k_d are arbitrary positive constants. Replacing (16.10) in (16.9):

$$\frac{dV(q_1, \dot{q}_1)}{dt} = -k_d \dot{q}_1 \text{sat}(\dot{q}_1) \text{sign}(V - V_0).$$

It is easy to realize from (16.11) that the product $\dot{q}_1 \text{sat}(\dot{q}_1)$ is always positive and it is zero only when $\dot{q}_1 = 0$. Hence, as long as $\dot{q}_1 \neq 0$ we have:

- If $V < V_0$ then $\frac{d}{dt}V > 0$ and energy V increases.
- If $V > V_0$ then $\frac{d}{dt}V < 0$ and energy V decreases.
- If $V = V_0$, then $\frac{d}{dt}V = 0$ and energy V remains at V_0 .

Then, it is ensured that the pendulum energy V reaches the required value to arrive at one of the points $(q_1, \dot{q}_1) = (+\pi, 0)$ or $(q_1, \dot{q}_1) = (-\pi, 0)$. Furthermore, the energy remains at that value until one of these points is reached. Now, analyze what happens if $\dot{q}_1 = 0$. Notice that the pendulum is at rest under this condition. However, the only way for $\dot{q}_1 = 0$ to be true for ever is $q_1 = 0$, because it is only there that the pendulum may remain at rest. Hence, it is necessary to slightly hit the pendulum at the beginning to force it to abandon the configuration $(q_1, \dot{q}_1) = (0, 0)$.

The previous analysis ensures that the pendulum reaches the inverted unstable configuration if, from the downward stable configuration, the pendulum is slightly hit to start it moving. It is not difficult to realize that none of the above results change

if $\text{sat}(V - V_0)$ is used in controller (16.10) instead of $\text{sign}(V - V_0)$. The controller in (16.10), (16.11), has its roots in the ideas reported in [13].

On the other hand, it is important to stress that, once the inverted unstable configuration is reached, the controller (16.10) does not ensure that the pendulum stays at such a configuration. This requires another controller to be designed that must be put to work when the pendulum is very close to the inverted unstable configuration. This means that the controller in (16.10) must be disconnected from that time on. In the next section, a controller is designed solving the second part of the control problem: pendulum balancing at the inverted unstable configuration.

Finally, notice that nothing has been said on the wheel position and velocity during the pendulum swing-up stage. As the wheel is symmetrical and homogeneous, then the wheel position does not matter. On the other hand, although the wheel velocity increases in an unknown manner during this time interval, the velocity is finite at the moment when the pendulum reaches the inverted unstable configuration; hence, the balancing controller must just regulate the wheel velocity at the value that this variable has when this controller begins to work.

Example 16.1 Consider the alternative controller:

$$\tau_1 = -k_d (V - V_0) \text{sat}(\dot{q}_1), \quad u = -\frac{R}{k_m} \tau_1, \quad (16.12)$$

$$\text{sat}(\dot{q}_1) = \begin{cases} d, & \dot{q}_1 > d \\ -d, & \dot{q}_1 < -d \\ \dot{q}_1, & |\dot{q}_1| \leq d \end{cases}, \quad d = \frac{u_{\max} k_m}{k_d R V_0}, \quad (16.13)$$

where d , u_{\max} and k_d are positive arbitrary constants. Replacing (16.12) in (16.9):

$$\frac{dV(q_1, \dot{q}_1)}{dt} = -k_d (V - V_0) \dot{q}_1 \text{sat}(\dot{q}_1).$$

It is easy to realize, from (16.13), that the product $\dot{q}_1 \text{sat}(\dot{q}_1)$ is always positive and it is zero only when $\dot{q}_1 = 0$. Hence, as long as $\dot{q}_1 \neq 0$, we have the following:

- If $V < V_0$ then $\frac{d}{dt} V > 0$ and energy V increases.
- If $V > V_0$ then $\frac{d}{dt} V < 0$ and energy V decreases.
- If $V = V_0$ then $\frac{d}{dt} V = 0$ and energy V remains at V_0 .

Thus, it is ensured again that the pendulum energy V reaches the required value to arrive at one of the points $(q_1, \dot{q}_1) = (+\pi, 0)$ or $(q_1, \dot{q}_1) = (-\pi, 0)$.

16.4 Balancing Controller

The strategy for balancing the pendulum is a linear state feedback controller designed on the basis of a linear approximation of (16.3). This linear approximation

is obtained in the following. Note that the model in (16.3) does not depend on the wheel position q_2 . Hence, the state vector can be defined as:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} q_1 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}, \quad (16.14)$$

From (16.3):

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = D^{-1} \left\{ - \begin{bmatrix} \bar{m}g \sin(q_1) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tau \end{bmatrix} \right\}, \quad (16.15)$$

$$D^{-1} = \begin{bmatrix} \bar{d}_{11} & \bar{d}_{12} \\ \bar{d}_{21} & \bar{d}_{22} \end{bmatrix} = \frac{1}{d_{11}d_{22} - d_{12}d_{21}} \begin{bmatrix} d_{22} & -d_{12} \\ -d_{21} & d_{11} \end{bmatrix}. \quad (16.16)$$

Note that $\det(D^{-1}) \neq 0$ because it is the inverse matrix of a nonsingular matrix. Using (16.14) and (16.15), the model in (16.3) can be written as:

$$\dot{x} = f(x, \tau) = \begin{bmatrix} x_2 \\ -\bar{d}_{11}\bar{m}g \sin(x_1) + \bar{d}_{12}\tau \\ -\bar{d}_{21}\bar{m}g \sin(x_1) + \bar{d}_{22}\tau \end{bmatrix}. \quad (16.17)$$

According to Sect. 7.3.2, the operation points (x^*, τ^*) of (16.17) are found from the condition $f(x^*, \tau^*) = 0$, i.e.,:

$$\begin{bmatrix} x_2^* \\ -\bar{d}_{11}\bar{m}g \sin(x_1^*) + \bar{d}_{12}\tau^* \\ -\bar{d}_{21}\bar{m}g \sin(x_1^*) + \bar{d}_{22}\tau^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (16.18)$$

It is clear that:

$$x_2^* = 0. \quad (16.19)$$

On the other hand, from the second and third rows in (16.18), the following is found:

$$\begin{aligned} \tau^* &= \frac{\bar{d}_{11}\bar{m}g \sin(x_1^*)}{\bar{d}_{12}}, \\ \tau^* &= \frac{\bar{d}_{21}\bar{m}g \sin(x_1^*)}{\bar{d}_{22}}. \end{aligned} \quad (16.20)$$

Solving these expressions simultaneously:

$$\frac{\bar{m}g}{\bar{d}_{12}\bar{d}_{22}} [\bar{d}_{11}\bar{d}_{22} - \bar{d}_{21}\bar{d}_{12}] \sin(x_1^*) = 0. \quad (16.21)$$

As $\det(D^{-1}) = \bar{d}_{11}\bar{d}_{22} - \bar{d}_{21}\bar{d}_{12} \neq 0$, then (16.21) is only satisfied if:

$$x_1^* = \pm n\pi, \quad n = 0, 1, 2, 3, \dots \quad (16.22)$$

Replacing this condition in any of the expressions in (16.20), the following is found:

$$\tau^* = 0. \quad (16.23)$$

Finally, as no restriction exists on x_3^* , it is concluded that:

$$x_3^* = c, \quad (16.24)$$

where c is any real constant. According to Sect. 7.3.2, the linear approximation of (16.17) at the operation point (16.19), (16.22), (16.23), (16.24), with $n = \pm 1$, is given as:

$$\dot{z} = Az + Bw, \quad (16.25)$$

$$A = \left. \frac{\partial f(x, \tau)}{\partial x} \right|_{x_{\tau}^*} = \left[\begin{array}{ccc} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{array} \right]_{x_{\tau}^*} = \left[\begin{array}{ccc} 0 & 1 & 0 \\ \bar{d}_{11}\bar{m}g & 0 & 0 \\ \bar{d}_{21}\bar{m}g & 0 & 0 \end{array} \right],$$

$$B = \left. \frac{\partial f(x, \tau)}{\partial \tau} \right|_{x_{\tau}^*} = \left[\begin{array}{c} \frac{\partial f_1}{\partial \tau} \\ \frac{\partial f_2}{\partial \tau} \\ \frac{\partial f_3}{\partial \tau} \end{array} \right]_{x_{\tau}^*} = \left[\begin{array}{c} 0 \\ \bar{d}_{12} \\ \bar{d}_{22} \end{array} \right],$$

where:

$$z = x - x^*, \quad (16.26)$$

$$w = \tau - \tau^*, \quad (16.27)$$

remain small, i.e., $z \approx 0$, $w \approx 0$. On the other hand, recall that $\tau^* = 0$ and, hence, $w = \tau$. Then, (16.6) can be used to find the following modified version of (16.25):

$$\dot{z} = Az + Bu, \quad (16.28)$$

$$\mathcal{B} = \begin{bmatrix} 0 \\ \bar{d}_{12} \\ \bar{d}_{22} \end{bmatrix} \frac{k_m}{R}.$$

This approximate linear model is to be used to design the balancing controller.

First, the pair (A, \mathcal{B}) is checked to be controllable. This requires computation of the determinant of the controllability matrix $C = [\mathcal{B}|A\mathcal{B}|A^2\mathcal{B}]$, which is found to be:

$$\det(C) = \frac{\bar{m}gk_m^3}{R^3} \bar{d}_{12}^2 (\bar{d}_{11}\bar{d}_{22} - \bar{d}_{12}\bar{d}_{21}) \neq 0. \quad (16.29)$$

Hence, it is concluded that the pair (A, \mathcal{B}) is controllable. This means that it is always possible to find a gain vector K such that all eigenvalues of the closed-loop matrix $(A - \mathcal{B}K)$ are assigned at arbitrary locations. This is achieved using the controller:

$$u = -Kz. \quad (16.30)$$

The gain vector K is computed once the numerical values of the parameters involved in matrices A and \mathcal{B} are known. The controller (16.30) renders it possible to balance the IWP at the inverted unstable configuration defined by the operation point (16.19), (16.22), (16.23), (16.24), with $n = \pm 1$. It is important to stress that this requires the mechanism state vector x to be close enough to such an unstable configuration at the time when the controller (16.30) starts to work.

Finally, it is important to say that the constant c introduced in (16.24) takes the value of the wheel velocity at the time when the controller (16.30) starts to work. As this velocity may take any value, it is instrumental that c can be defined as any real constant.

16.5 Prototype Construction and Parameter Identification

A drawing of the IWP that has been built is shown in Fig. 16.4. There, the main components are shown. Some of these components are included to successfully assemble the mechanical pieces. An incremental encoder and a tachometer are also included, to measure the pendulum position q_1 and the wheel velocity \dot{q}_2 respectively. The dimensions of the pieces employed are presented in Table 16.1.

The pendulum equivalent mass m_1 is defined as:

$$m_1 = m_{sol} + m_{car} + m_{sop} = 0.05975 \text{ [Kg]}. \quad (16.31)$$

Fig. 16.4 Inertia wheel pendulum

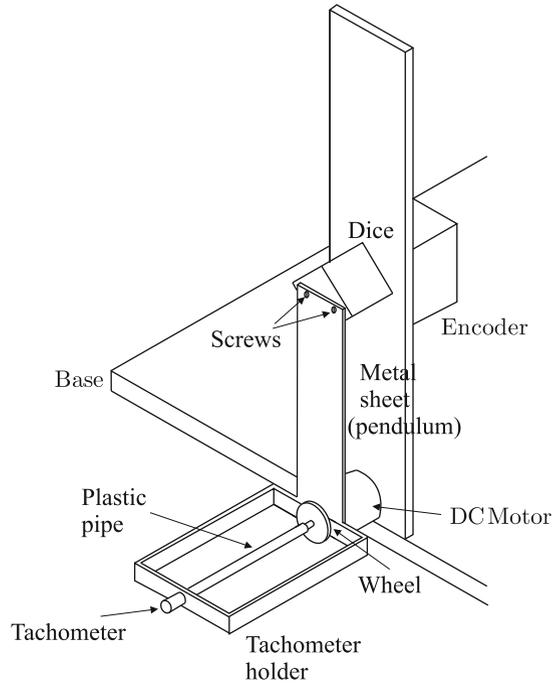


Table 16.1 Simple parameters

Symbol	Description	Value	Units
l_1	Metal sheet length	0.117	m
a_{sol}	Metal sheet width	0.0254	m
m_{sol}	Metal sheet mass	0.016	Kg
m_{car}	Actuator stator mass	0.03	Kg
m_{sop}	Tachometer holder mass	0.01375	Kg
m_{rot}	Actuator rotor mass	0.02	Kg
r_{rot}	Actuator rotor radius	0.00915	m
m_{rue}	Wheel mass	0.038	Kg
r_{rue}	Wheel radius	0.0189	m
R	Armature resistance	4.172	Ohm
L	Armature inductance	0.0009	H
k_b	Counter electromotive force constant	0.00775	Vs/rad
k_m	Torque constant	0.00775	Nm/A

To compute the pendulum’s center of mass including all its components, the balance principle is employed, i.e., the sum of all the torques around the support point is zero. Hence, when applying an equivalent weight at a distance l_{c1} on the opposite end of the metal sheet, the assembly in Fig. 16.5 must be balanced. Using (16.31) and the data in Table 16.1, the following is found:

$$l_{c1} = \frac{m_{sol} \frac{l_1}{2} + m_{car} l_1 + m_{sop} l_1}{m_1} = 0.10133 \text{ [m]}. \tag{16.32}$$

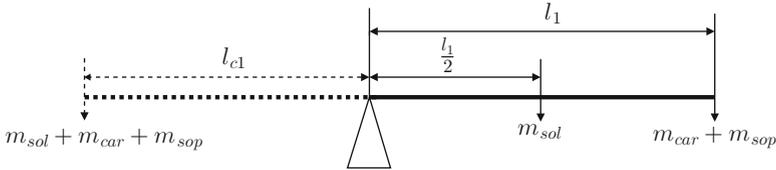


Fig. 16.5 The balance principle to compute l_{c1}

The moment of inertia of a rectangular thick sheet with mass m , width b , and length d , with rotation axis x orthogonal to the plane, with edges b and d passing through its geometric center, is given as [14, 15], pp. 271:

$$I_x = \frac{1}{12} m (b^2 + d^2). \tag{16.33}$$

Using (16.33) and the data in Table 16.1, the moment of inertia I^* is obtained as:

$$I^* = \frac{1}{12} m_{sol} (a_{sol}^2 + l_1^2) = 0.0000191122 \text{ [Kg m}^2\text{]}. \tag{16.34}$$

However, the moment of inertia I_1 is defined around an axis passing through the pendulum center of mass located at a distance $l_{c1} - l_1/2$ from the center of the rectangle with edges b and d . According to the *parallel axis theorem* or *Steiner theorem* [14, 15], pp. 272:

$$I_1 = I^* + m_{sol} \left(l_{c1} - \frac{l_1}{2} \right)^2 = 0.000048463 \text{ [Kg m}^2\text{]}.$$

On the other hand, the moment of inertia of a disc with a mass m and radius r with rotation axis x orthogonal to the disc and passing through its geometric center is given as [14, 15], pp. 271:

$$I_x = \frac{1}{2} m r^2. \tag{16.35}$$

Using the data in Table 16.1 and the expression in (16.35) it is found that the wheel's moment of inertia I_{rue} is given as:

$$I_{rue} = \frac{1}{2} m_{rue} r_{rue}^2 = 0.000006787 \text{ [Kg m}^2\text{]}. \tag{16.36}$$

The moment of inertia of a cylinder with a mass m , radius a , and height l , with a rotation axis x passing through its longitudinal axis is given as [14, 15], pp. 271:

$$I_x = \frac{1}{2}ma^2. \quad (16.37)$$

Using the expression in (16.37) and the data in Table 16.1, the moment of inertia of the actuator rotor I_{rot} is found to be:

$$I_{rot} = \frac{1}{2}m_{rot}r_{rot}^2 = 0.000000837225 \text{ [Kg m}^2\text{]}. \quad (16.38)$$

The equivalent wheel mass m_2 is defined as:

$$m_2 = m_{rot} + m_{rue} = 0.058 \text{ [Kg]}. \quad (16.39)$$

Using (16.36) and (16.38), it is found that the equivalent wheel moment of inertia I_2 is given as:

$$I_2 = I_{rue} + I_{rot} = 0.0000076242 \text{ [Kg m}^2\text{]}. \quad (16.40)$$

Finally, using the data in Table 16.1, the parameters in (16.34), (16.31), (16.32), (16.39), (16.40), and $g = 9.81 \text{ [m/s}^2\text{]}$, the following numerical values for the parameters of model (16.3) are found:

$$\begin{aligned} d_{11} &= 0.0014636, & d_{12} &= 0.0000076, & (16.41) \\ d_{21} &= 0.0000076, & d_{22} &= 0.0000076, \\ \bar{m}g &= 0.12597. \end{aligned}$$

It is worth saying that the tachometer mass is considered as a part of the tachometer holder mass and the effect of the tachometer rotor is neglected when computing I_2 . This is because it has a mass less than 0.0001 [Kg] and a radius less than 0.001 [m] . On the other hand, the screws holding the dice, the dice itself, and the incremental encoder shaft are neglected when computing the inertia I_1 . This is because the addition of these masses is less than 0.008 [kg] and, more importantly, they are placed at the point where the pendulum hangs up, which reduces strongly their inertial effects. Finally, the mass of the electric wires and the plastic pipe used to couple the tachometer to the wheel were also neglected.

Example 16.2 Some simulation results are presented in the following when controlling the IWM (16.3), with numerical parameters in (16.41), using the controller in (16.10) to solve the swing-up task and the controller in (16.30) to balance the pendulum at the inverted configuration. The switching condition between the controllers is:

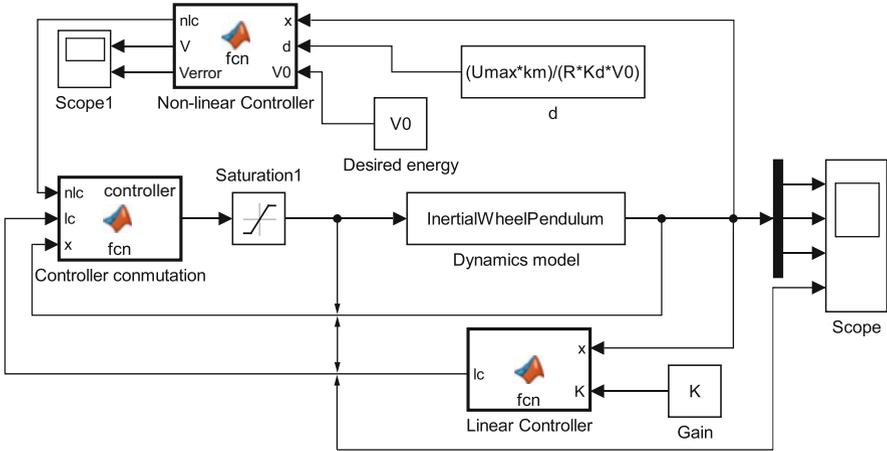


Fig. 16.6 MATLAB/Simulink diagram for the inertia wheel pendulum

If: $z_1^2 + z_2^2 \leq 0.01$, controller in (16.30) is on,
 else: controller in (16.10) is on,

with z_1 and z_2 defined in (16.26). All of the initial conditions were chosen to be zero except for $\dot{q}_1(0) = 0.00001[\text{rad/s}]$. The controller in (16.10) was set with $k_d = 20$. The vector of gains K for controller (16.30) was selected such that the eigenvalues of the closed-loop matrix $A - BK$ are assigned at $-9.27 \pm 20.6j$ and -0.719 . These simulations were performed using the MATLAB/Simulink diagram in Fig. 16.6 where there are four main blocks that contain the code listed in Appendix G.2. Before running the simulation in Fig. 16.6, the code in Appendix G.2.5 must be executed to assign numerical values to all the parameters.

It is observed in Fig. 16.7 that the pendulum swings up and is finally balanced at its inverted unstable configuration. Moreover, the wheel velocity remains bounded along the complete simulation. Also note that the applied voltage remains within a suitable range, i.e., between $-13[\text{V}]$ and $+13[\text{V}]$ and, finally, the pendulum energy converges to its desired value V_0 . Thus, these simulation results corroborate the theoretical results stated above.

16.6 Controller Implementation

Using the numerical values in Sect. 16.5, $k_d = 1$ and according to (16.7), (16.10), it is found that the swing-up controller is given as:

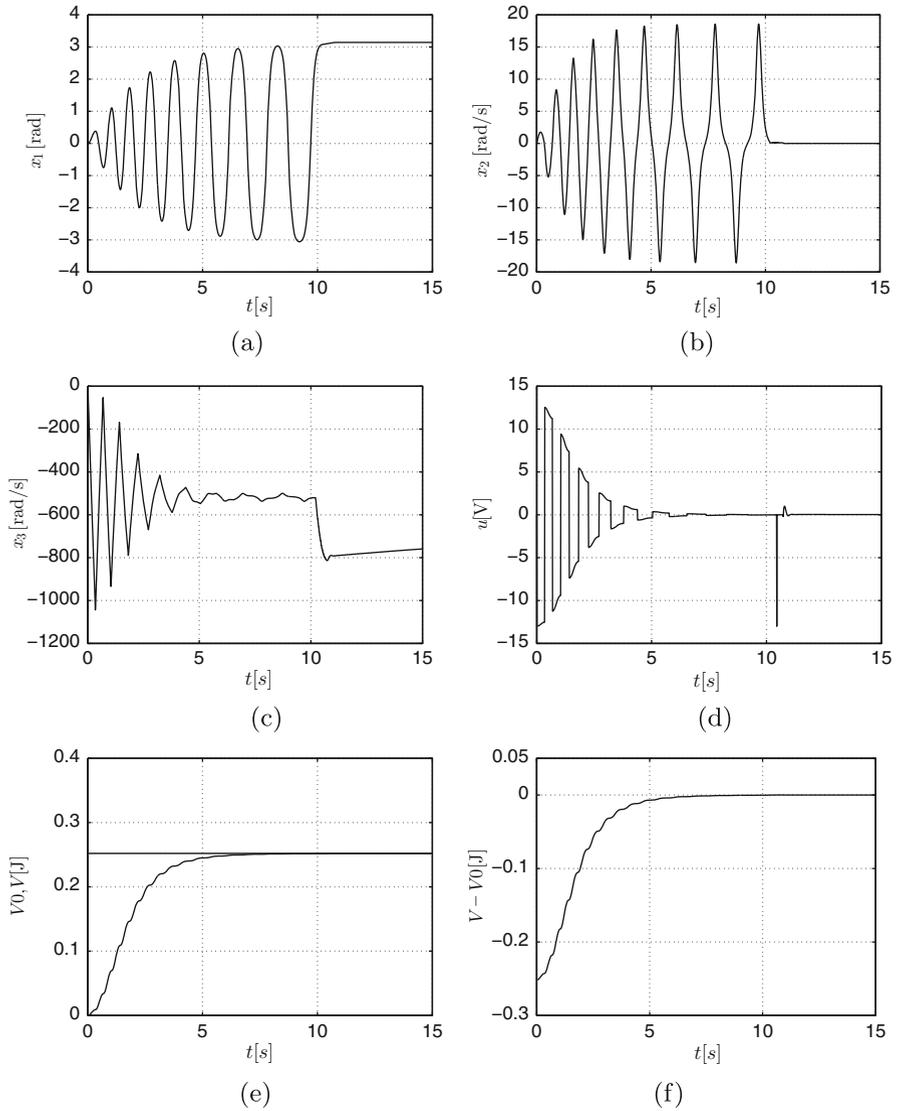


Fig. 16.7 Simulation results when swinging up and stabilizing the IWP with controllers in (16.10) and (16.30) . **(a)** Pendulum position. **(b)** Pendulum velocity. **(c)** Wheel velocity. **(d)** Applied voltage. **(e)** Pendulum energy. **(f)** Pendulum energy error

$$\begin{aligned}
 V &= C_1 \dot{q}_1^2 + \bar{m}g(1 - \cos(q_1)), & C_1 &= \frac{1}{2}J, & V_0 &= 2\bar{m}g, \\
 u &= C_2 \text{sat}(\dot{q}_1) \text{sign}(V - V_0), & C_2 &= \frac{Rk_d}{k_m}, & \varepsilon_0 &= \frac{u_{\max}k_m}{k_d R}, \quad (16.42)
 \end{aligned}$$

where $u_{\max} = 13[\text{V}]$, whereas the constants C_1 and C_2 are computed only once when initializing variables, to reduce the number of multiplications required for implementation.

On the other hand, according to Sects. 7.14.1 and 7.14.2, in Chap. 7, it is found that the linear controller (16.30), together with the gain vector $K = [k_1, k_2, k_3] = [-340, -11, -0.0085]$ assign the eigenvalues of matrix $(A - BK)$ at: $\bar{\lambda}_1 = -5.8535 + 17.7192j$, $\bar{\lambda}_2 = -5.8535 - 17.7192j$, $\bar{\lambda}_3 = -0.5268$. Hence, the controller (16.30) is expressed as:

$$u = -k_1(q_1 - q_{1d}) - k_2\dot{q}_1 - k_3(\dot{q}_2 - \dot{q}_{2d}), \quad (16.43)$$

where any of $q_{1d} = +\pi$ or $q_{1d} = -\pi$ is used, depending on which one is first reached by q_1 . Recall that $\dot{q}_{2d} = c$ is the wheel velocity at the time when the controller (16.43) starts to work.

Define the condition:

$$(q_1 - q_{1d})^2 + \dot{q}_1^2 < \delta, \quad (16.44)$$

for some constant that is small enough $\delta > 0$ and $q_{1d} = +\pi$ or $q_{1d} = -\pi$. This condition indicates that the pendulum is close to one of the points $(q_1, \dot{q}_1) = (+\pi, 0)$ or $(q_1, \dot{q}_1) = (-\pi, 0)$. In the experiments presented later $\delta = 0.3$ has been used.

The control strategy to swing up and to balance the IWP is to employ (16.42) until (16.44) is satisfied and, from that time on, turn off (16.42) and turn on (16.43). This task is performed using the Microchip PIC18F4431 microcontroller. It is explained next how this is done. This part is not intended to present a detailed explanation on how this microcontroller works, but merely a basic description of how this microcontroller is employed to implement the control strategy (16.42), (16.43), (16.44).

The complete code used to program the microcontroller is presented at the end of this chapter. The microcontroller PIC18F4431 [16] works at 40 MHz as maximal clock frequency. In this chapter, a 11-MHz crystal is used. Five programmable input–output ports, three 16-bit timers, one 8-bit timer (employed to fix the sample period $T_s = 0.01[\text{s}]$), and nine 10-bit analog/digital channels (the tachometer voltage enters through channel 0), are available. There are also eight 14-bit PWM channels among which CCP1 is programmed to work with 8 bit to deliver voltage to be applied to the actuator (PM brushed DC motor).

The position q_1 is measured using an incremental encoder model CP-360-S from Computer Optical Products Inc. [17]. This device, once connected to the microcontroller, has a resolution of 1440 ppr. Pulses of this incremental encoder are read through the microcontroller pins 5 and 6 (see Fig. 16.8). The incremental encoder

counts are stored in two 8-bit bytes called POSCNTH, the most significant, and POSCNTL, the less significant. The actual count is computed as:

$$pos = 256 * POSCNTH + POSCNTL,$$

The instruction “q1=(signed long)pos” assigns to q1 the numerical value of the encoder count, which includes the corresponding sign, i.e., positive if movement is as that shown in Fig. 16.2 or negative otherwise. Finally, the position q_1 is given in radians when performing:

$$q_1 = \frac{2\pi}{1440} q1, \quad \frac{2\pi}{1440} = 0.004363.$$

It is worth stating that the complete control system must be turned on when the pendulum is at rest at $q_1 = 0$, according to the convention in Fig. 16.2. The pendulum velocity \dot{q}_1 is computed by numerical differentiation of the pendulum position q_1 , i.e.,:

$$\dot{q}_1 = \frac{q_1(k) - q_1(k-1)}{T},$$

where k stands for the discrete time, \dot{q}_1 represents the estimate of the pendulum velocity in radians/second whereas $T_s = 0.01[s]$ is the sampling period. Hence, $q_1(k)$ represents the value of q_1 measured at the present sample time and $q_1(k-1)$ represents the value of q_1 measured in the previous sample time.

The wheel velocity \dot{q}_2 is measured using a tachometer model 34PC. It is a miniature PM brushed DC motor used in cell phones as a buzzer. Voltage delivered by the tachometer is low-pass filtered using the RC circuit shown in Fig. 16.8. It was found that this tachometer delivers 0.58[V] when the wheel velocity is maximal, i.e., 1634 [rad/s]. It was assumed that this voltage is proportional at any other intermediate velocity. To improve the resolution of this velocity measurement, the tension divider connected to pin 4 (see Fig. 16.8) was employed to fix at 3.8372[V] the minimal value of the analog range of the analog/digital converter channel 0 (the maximal value is fixed at 5[V]). On the other hand, the tachometer is series connected to another tension divider, such that, when the tachometer is at rest, i.e., when $\dot{q}_2 = 0$, voltage at channel 0 is $(5-3.8372)/2+3.8372=4.4186[V]$. Notice that variations of about 0.58[V], i.e., 1634[rad/s], are allowed upward and downward from the middle point of the converter analog range. Hence, when $\dot{q}_2 = 1634[\text{rad/s}]$ the analog/digital converter delivers the count $1023 = 2^{10} - 1$ (the AD converter is 10-bit) and when $\dot{q}_2 = 0[\text{rad/s}]$ the analog/digital converter delivers the count $1023/2 = 511$. Counts smaller than 511 represent negative velocities \dot{q}_2 . According to this information, the wheel velocity is computed as follows. The instruction “vel=read-adc()-511” assigns to the variable “vel” the code corresponding to the wheel velocity, whereas the instructions “q2_p=(signed long)vel” and “q2_p=3.2039*q2_p” assign to variable “q2_p” a value that is numerically equal to

the wheel velocity \dot{q}_2 in radians/second, the sign included. Notice that the constant $3.2039=1634/510[(\text{rad/s})/\text{count}]$ represents the measurement system gain.

With this information, any of the expressions in (16.42) or (16.43) are computed. The resulting variable u is sent to the microcontroller PWM as described in the following.

An equivalent way of implementing the saturation function in (16.42) with a smaller number of computations is the following. It is known that u only takes values in the range $[-13, +13][\text{V}]$; hence, its value is constrained to this range of values by using instructions such as:

$$\begin{aligned} \text{if } u > 13 \text{ then } u &= 13, \\ \text{if } u < -13 \text{ then } u &= -13. \end{aligned}$$

As the PWM resolution is $2^8 - 1 = 255$ counts, then the following value must be sent to the microcontroller PWM to correctly deliver u to the power amplifier:

$$\text{cuenta} = \frac{255}{13} \text{abs}(u),$$

where $\text{abs}(u)$ represents the absolute value of u . The sign of u is stored in two control bits. The PWM signal is applied to an H bridge (L293B from SGS-Thomson Microelectronics), which suitably amplifies the power of the PWM signal. This H bridge is fed with a 15.3[V] power supply. However, according to the data sheet of this device [18], the voltage drops constrain to $\pm 13[\text{V}]$ the voltage signal to be delivered. This device has two bits, which are employed to indicate the sign of the power voltage to deliver. Thus, the sign of u is sent to the H bridge through the bits 0 and 1 of the microcontroller port D. This means that the average power voltage sent to the motor is in the range $[-13, +13][\text{V}]$. Hence, it is ensured that the voltage applied at the motor terminals u has an average value that is numerically equal to u , computed by any of the expressions in (16.42) or (16.43). The electric diagram used to implement the control strategy in (16.42), (16.43), (16.44), is shown in Fig. 16.8.

16.7 Experimental Results

The experimental results obtained with the proposed control scheme are shown in Figs. 16.9, 16.10, 16.11, 16.12, and 16.13. It is important to recall that it is necessary to apply a slight hit on the pendulum at the beginning to abandon the configuration $(q_1, \dot{q}_1) = (0, 0)$. Although this can be replaced by sending a small voltage pulse to the actuator at the beginning, this was not programmed.

In Fig. 16.9, it is observed how the pendulum oscillates with ever-growing amplitude until the position $q_1 = -\pi$ is reached with zero velocity at $t \approx 14[\text{s}]$ and the pendulum stays there from that time on. In Fig. 16.10, it is observed that the

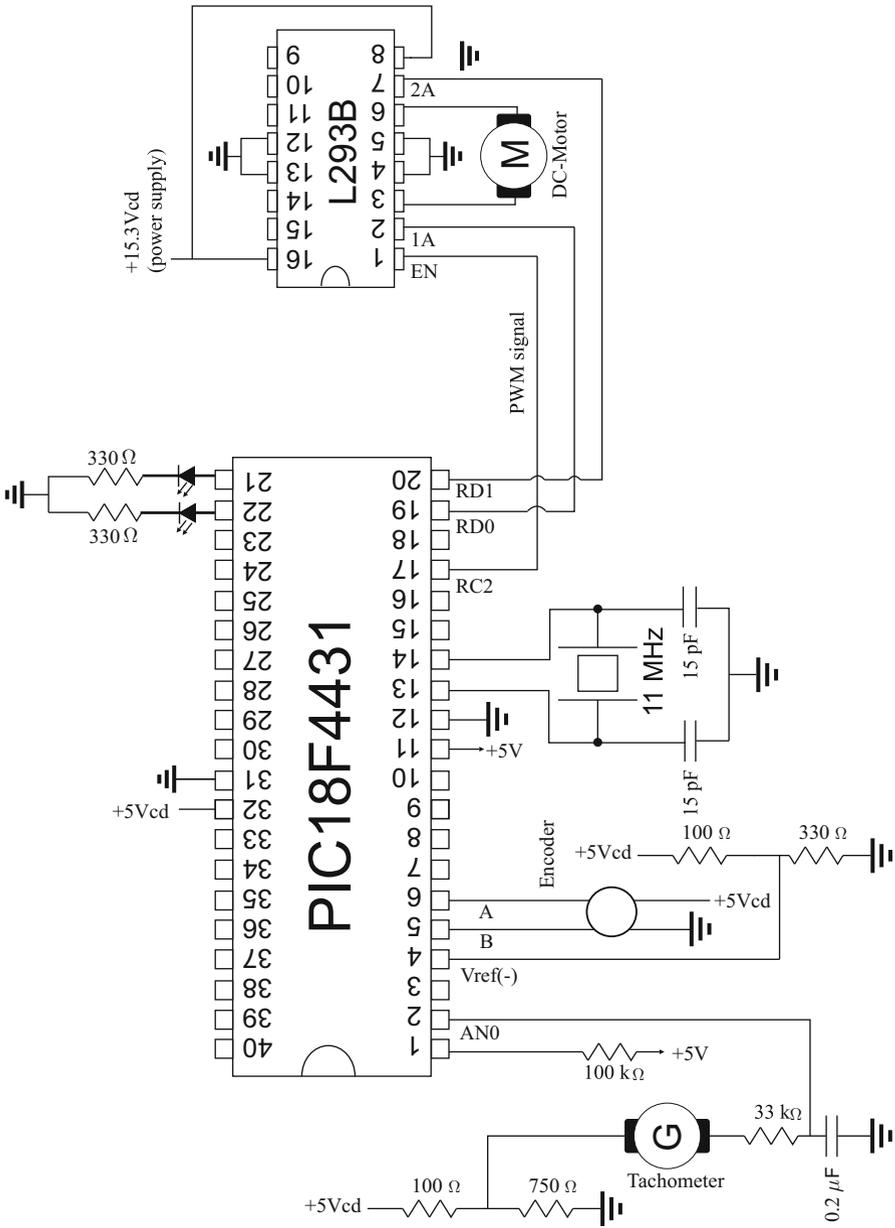


Fig. 16.8 Control system electric diagram

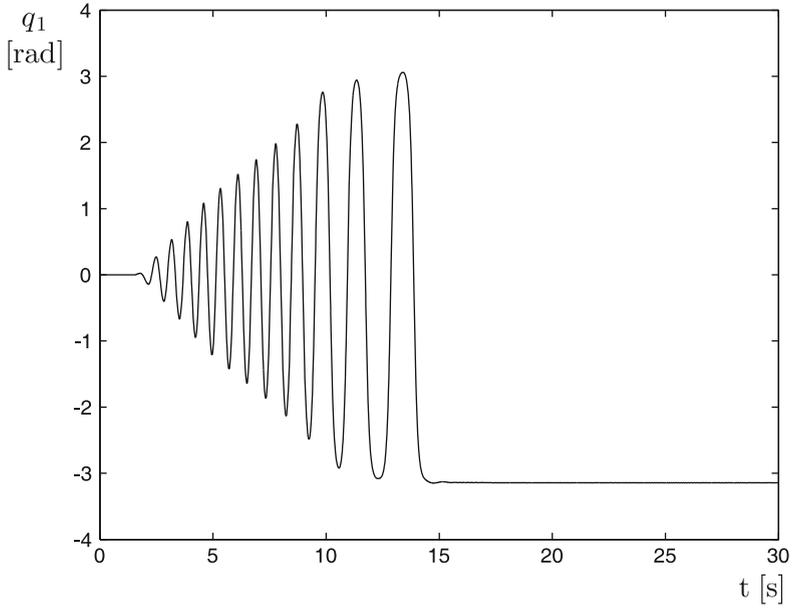


Fig. 16.9 Pendulum position as the inverted unstable configuration is reached

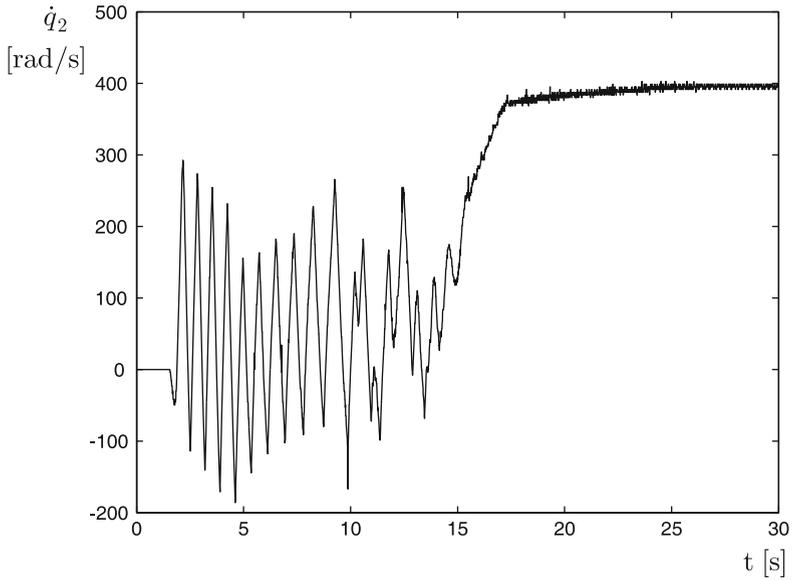


Fig. 16.10 Wheel velocity

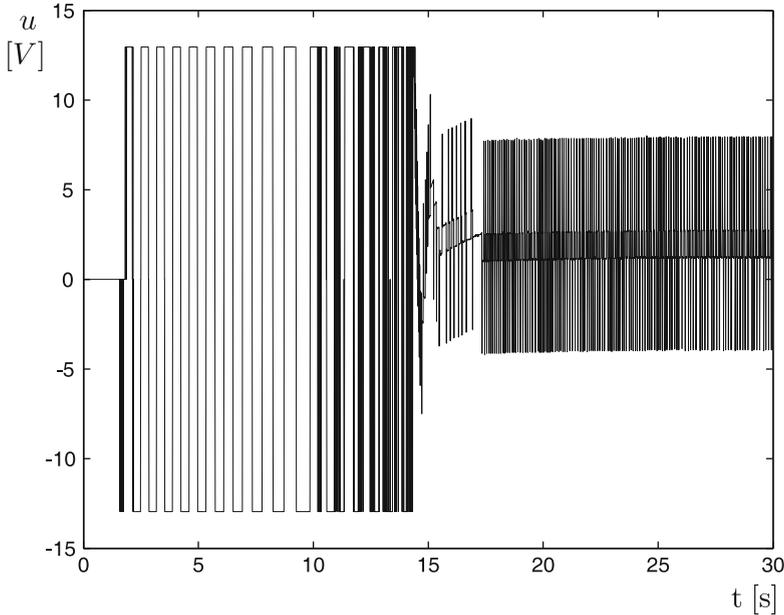


Fig. 16.11 Voltage applied to the motor

wheel velocity remains constant at $380[\text{rad/s}]$ from $t \approx 17[\text{s}]$ on. It is interesting to point out that the wheel velocity is commanded to be regulated at $c \approx 150[\text{rad/s}]$, which is the wheel velocity at $t \approx 14[\text{s}]$, i.e., when the controller (16.43) starts to work. The difference between c and the final wheel velocity $380[\text{rad/s}]$ is due to friction present at the wheel shaft, which has not been taken into account during the design stage.

In Fig. 16.11, the control signal u is shown. Note that this variable saturates most of the time during the swing-up stage ($t < 14[\text{s}]$). This is the reason why saturation and sign functions are used in the controller (16.42). In fact, $\varepsilon_0 = 0.024$ is chosen so that with this value and the factor $\frac{4.172}{0.00775}$, u saturates at $\pm 13[\text{V}]$.

In Fig. 16.12, it is shown how the pendulum evolves on the plane defined in Fig. 16.3, Sect. 16.3. It is clear that, comparing both figures, the pendulum moves such that its energy V increases (see also Fig. 16.13) until $V = V_0 = 0.2519$. Note that V reaches such a value as $t \approx 11[\text{s}]$, but the pendulum is balanced until $t \approx 14[\text{s}]$. This demonstrates that energy is regulated at $V = V_0$ until the point $(q_1, \dot{q}_1) = (-\pi, 0)$ is reached. It is clear that the pendulum also passes close to the point $(q_1, \dot{q}_1) = (+\pi, 0)$ at $t \approx 13[\text{s}]$ but it is not balanced, perhaps the condition in (16.44) was not satisfied at that time, i.e., the pendulum did not pass close enough to such a point.

Finally, in Fig. 16.14, a picture is shown where the IWP is observed balancing at the inverted unstable configuration.

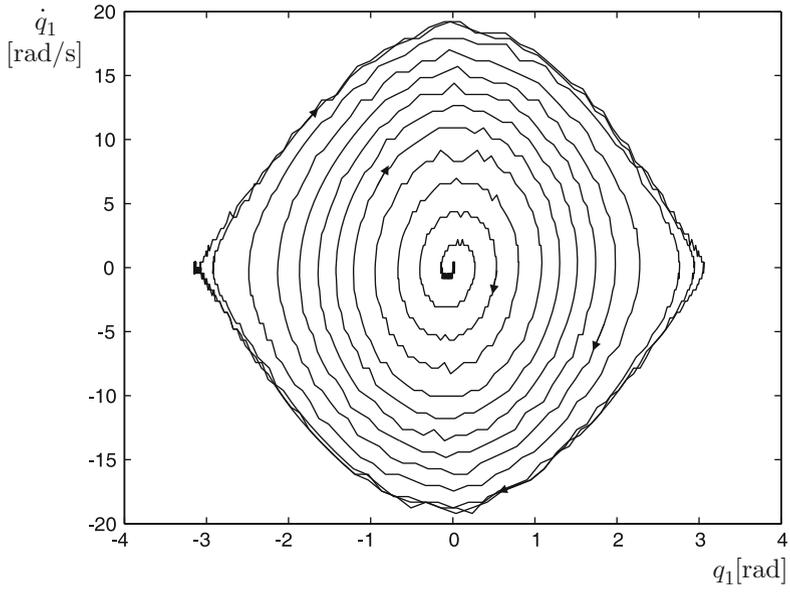


Fig. 16.12 Pendulum evolution on the plane $\dot{q}_1 - q_1$ (phase plane)

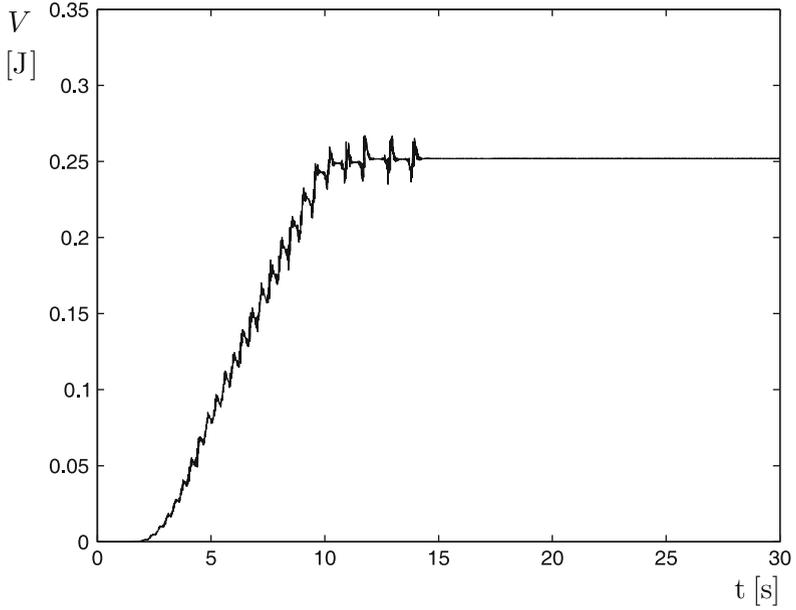


Fig. 16.13 Energy evolution as the inverted unstable configuration is reached

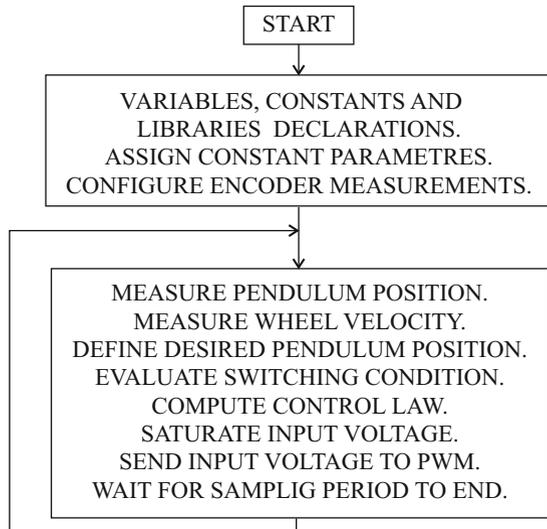


Fig. 16.14 The inertia wheel pendulum balancing at the inverted unstable configuration

16.8 PIC18F4431 Microcontroller Programming

The reader is referred to [19] for detailed explanations on each one of the instructions appearing in the following program. The corresponding flow diagram is presented in Fig. 16.15.

Fig. 16.15 Flow diagram for PIC18F4431 microcontroller programming when used to control an inertia wheel pendulum



```
#include<18f4431.h>
#device adc=10 //adc, 10 bits
#include<stdlib.h>
#include<math.h>

#fuses HS,NOWDT,NOPROTECT,PUT,NOLVP,NOBROWNOUT,NOWRTC,
MCLR,SSP_RC
#define pi_ 3.14159
//nonlinear controller gain
#define kd 1.0
//linear controller gains
#define k1 340
#define k2 11
#define k3 0.0085
//parameters
#define mbg 0.12597
#define R 4.172

#define L 0.0009
#define Kb 0.00775
#define Km 0.00775
#define d11 0.0014636
#define d12 0.0000076

#define d21 0.0000076
#define d22 0.0000076

#define delta 0.3
#define ts 0.01
//with timer=108, each count= (4/FXtal)*256 sec

/*Register addresses*/
#use delay(clock=11000000)
//time basis (11 MHz)
#byte porta = 0xf80
//ports addresses
#byte portb = 0xf81

#byte portc = 0xf82
#byte portd = 0xf83
#byte portd = 0xf84

#byte TMR0H = 0xfd7
#byte TMR0L = 0xfd6
#byte T0CON = 0xfd5

#byte INTCON= 0xff2
#byte TMR5H=0xf88
//high encoder count

#byte TMR5L=0xf87 //low encoder count

#byte QEICON=0xfB6 //config. quadrature module
```

```

#byte T5CON=0xfB7 //Config. TIMER 5

#byte POSCNTL=0xF66 //CAP2BUFL (reg. low count)
#byte POSCNTH=0xF67 //CAP2BUFH (reg. high count)
#byte CAP1CON=0xF63 //reg. config. reset time basis

#byte PIE3=0xFA3

/*bits addresses */

#bit VCFG1=0xfc1.7 //Bit config. ADC

#bit PD0=0x0f83.0 //control CCW H bridge
#bit PD1=0x0f83.1
//control CW H bridge
#bit PD2=0x0f83.2
//led at board
#bit PD3=0x0f83.3 //led a board
/*variables declaration*/

float u, q1, q1_p, q1_l, q2_p, q1_d, q2_pd, gr, V, S, nf, i_ts,
      Je2, V0, RKdeKm;
long pos, vel, inter;
int pwm;
signed int n, mod;

/*main program*/
void main(void)
{
set_tris_a(0b11111111);
//Config. I/O ports
set_tris_b(0b00000000);
set_tris_c(0b10000000);
set_tris_d(0b00000000);
portb=0X00;
portc=0X00;
portd=0X00;
//CONFIG. PWM
setup_ccp1(CCP_PWM);
setup_ccp2(CCP_PWM);
setup_timer_2(T2_DIV_BY_16, 255, 1);
// Config. encoder reading
QEICON=QEICON | (0b00010101);
QEICON=QEICON & (0b01110101);
T5CON=T5CON | (0b00011001);
T5CON=T5CON & (0b10011101);
CAP1CON=CAP1CON | (0b01001111);

setup_timer_1(T1_INTERNAL | T1_DIV_BY_8);
TMR5L=0;
TMR5H=0;
POSCNTL=0;
POSCNTH=0;
//Config. ADC

```

```

setup_port_a(sAN0);
setup_adc(ADC_CLOCK_INTERNAL);
set_adc_channel(0);
delay_us(50);
VCFG1=1; //-Vref=AN2
INTCON=0;//turn off interrupt
//Config. timer 0
TMROL=0;
T0CON=0xC7;
//initializing variables
i_ts=1/ts;
Je2=(d11*d22-d12*d21)/(2*d12);
V0=2*mbg;
RKdeKm=R*Kd/Km;
q1=0.0;
q1_1=0.0;
q2_pd=0.0;
while (TRUE)
{
PD3=1;
q1_1=q1; // q1(k-1)
pos=POSCNTH;
pos=pos<<8;
pos=pos+POSCNTL;
q1=(signed long)pos;
q1=0.004363*q1; //arm position in rad
q1_p=(q1-q1_1)*i_ts; //arm velocity
vel=read_adc()-511; //wheel velocity
q2_p=(signed long)vel;
q2_p=3.2039*q2_p; //wheel velocity in rad/seg
//desired q1
n=q1/pi_;
nf=(float)n;
mod=n%2;
if(nf==0)
if(q1>=0) // desired value
q1_d=pi_;
else
q1_d=-pi_;
if((nf!=0)&&!PD2)
{
if(q1>=0) // desired value
{
if(mod==0)
q1_d=pi_*(nf+1.0);
else
q1_d=pi_*nf;
}
else
{
if(mod==0)
q1_d=pi_*(nf-1.0);
else
q1_d=pi_*nf;
}
}
}
}

```

```

}
}
//defining the operation region for each controller
if((q1-q1_d)*(q1-q1_d) + q1_p*q1_p < delta)
{
PD2=1; //linear controller
u= k1*(q1-q1_d) + k2*q1_p + k3*(q2_p-q2_pd);
}
else
{
PD2=0; //nonlinear controller
V=Je2*q1_p*q1_p+mbg*(1-cos(q1));
V=V-V0;
S=0; //sign function
if(V>0)
S=1.0;
if(V<0)
S=-1.0;
u=RKdeKm*q1_p*S; //nonlinear controller ends
q2_pd=q2_p; //wheel desired velocity
}
//saturation 13V(15.3 V - 2.3 V drop at H bridge= 13 V)
if(u>13)
u=13;
if(u<-13)
u=-13;
//scaling output, pwm 8 bits, 13 V
u=19.6*u;
//sense of rotation
if(u>=0)
{
PD0=1;
PD1=0;
}
else
{
PD0=0;
PD1=1;
}
//sending pwm
pwm=(unsigned int)abs(u);
set_pwm1_duty(pwm); //pwm
PD3=0; //control computation ends
while(TMR0L<108); //each count = (4/FXtal)*256 sec
TMR0L=0;
TOCON=0xC7;
} //closing infinite while
} //closing main

```

16.9 Summary

In this chapter, another underactuated nonlinear system has been presented: the inertia wheel pendulum. The mathematical model has been obtained and two controllers have been used to solve two different control problems: swing-up and balancing. The balancing control problem has been solved using a linear state feedback controller, whereas the swing-up control problem has been solved using a nonlinear controller. This has been done to introduce the reader to the study of nonlinear control systems. All parameters of the mechanism have been identified and the complete control system has been tested in experiments. A detailed description of the mechanism construction has also been presented.

16.10 Review Questions

1. Why is the inertia wheel pendulum an underactuated mechanism?
2. Why must the IWP oscillate several times before reaching the inverted configuration?
3. What is the corner frequency in [rad/s] and [Hz] of the tachometer resistor-capacitor filter shown in Fig. 16.8?
4. Why does the amplitude of voltage in Fig. 16.11 remain between 13[V] and -13[V] when the swing-up task is performed?

References

1. R. Kelly, V. Santibañez, and A. Loria, *Control of robot manipulators in joint space*, Springer, London, 2005.
2. M. W. Spong, P. Corke, and R. Lozano, Nonlinear control of the reaction wheel pendulum, *Automatica*, vol. 37, no. 11, pp. 1845–1851, 2001.
3. R. Olfati-Saber, Global stabilization of a flat underactuated system: the inertia wheel pendulum, *Proc. of the 40th IEEE Conf. on Decision and Control*, Orlando, Florida, pp. 3764–3765, 2001.
4. R. Ortega, M. W. Spong, F. Gómez-Estern, and G. Blankenstein, Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment, *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1218–1233, 2002.
5. S. Afkhami, M. J. Yazdanpanah, and P. J. Maralani, Stabilization of inertia wheel pendulum using output feedback back-stepping, *Proc. of IEEE Conf. on Control Applications*, vol. 2, pp. 977–982, 2003.
6. D. M. Alonso, F. I. Robbio, E. E. Paolini, and J. L. Moiola, Modelling an inertia wheel pendulum benchmark, *Mathematical and Computer Modelling of Dynamical Systems*, vol. 11, no. 3, pp. 255–272, 2005.
7. R. Kelly and R. Campa, IDA-PB control of the inertia wheel pendulum: the Lagrangian approach (in Spanish), *Revista Iberoamericana de Automática e Informática Industrial*, vol. 2, no. 1, pp. 36–42, 2005.
8. V. Santibañez, R. Kelly, and J. Sandoval, Control of the inertia wheel pendulum by bounded torques, *Proc. of the 44th IEEE Conf. on Decision and Control, and European Control Conf. CDC-ECC '05*, Seville, Spain, pp. 8266–8270, 2005.

9. M. Abrahantes, J. Mulder, and K. Butter, Modeling, identification and control of an under actuated inertial wheel pendulum, *Proc. of the 39th Southeastern Symposium on System Theory*, Macon, Georgia, pp.1–5, 2007.
10. N. Qaiser, N. Iqbal, A. Hussain, and N. Qaiser, Exponential stabilization of a class of underactuated mechanical systems using dynamic surface control, *International Journal of Control, Automation, and Systems*, vol. 5, no. 5, pp. 547–558, 2007.
11. R. V. Carrillo-Serrano and V. M. Hernández-Guzmán, Control of the inertia wheel pendulum taking into account the actuator dynamics, *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 12, pp. 5553–5563, 2010.
12. H. Khalil, *Nonlinear Systems*, 3rd Edition, Prentice-Hall, Upper Saddle River, 2002.
13. A. L. Fradkov, P. Y. Guzenko, D. J. Hill, and A. Y. Pogromsky, Speed gradient control and passivity of nonlinear oscillators, *Proc. 3rd. IFAC Symposium on Nonlinear control systems (NOLCOS'95)*, vol. 2, pp. 655–659, 1995.
14. R. C. Hibbeler, *Engineering Mechanics Dynamics*, 10th Edition, Prentice Hall, Upper Saddle River, NJ, 2004.
15. M. Alonso and E. J. Finn, *Physics*, Addison-Wesley, Reading, MA, 1992.
16. PIC18F4431 Enhanced Flash Microcontroller, Data sheet, Microchip Technology Inc., 2007.
17. CP360S Housed Encoder, Computer Optical Products Inc., www.opticalencoder.com.
18. L293B Push-Pull Four channel drivers, Data sheet, SGS-Thomson Microelectronics, 1993.
19. Custom Computer Services Incorporated, CCS C Compiler Reference Manual, 2003.