

An Embedded Controller for the Pendubot

Emmanouil Kourtikakis, Emmanouil Kapellakis, John Fasoulas, and Michael Sfakiotakis

Abstract—This paper presents the development of a low-cost embedded controller for the pendubot system. The pendubot is a two-link underactuated mechatronic device, frequently used for research and education in nonlinear control and robotics. The embedded controller employs the 8-bit Arduino Mega 2560 development platform, along with a custom-designed i/o board for interfacing with the pendubot plant. The developed firmware of the unit allows for balancing the pendubot in three different positions, by implementation of a gain-scheduled state feedback control law, designed via pole placement techniques that make use of the linearized dynamic model of the system. The control loop runs at a 1.25 kHz rate, attained by low-level programming of the microcontroller. A serial interface enables capturing and displaying in real-time on an external PC, data regarding the system response, for assessing and tuning the control performance. Overall, the developed embedded system represents a versatile and cost-effective platform for digital control implementation.

Index Terms—Pendubot, Embedded control, Mechatronics, Robotics.

I. INTRODUCTION

The pendubot is an underactuated mechatronic device that is frequently used as an educational and research platform in nonlinear control and robotics [1]–[7]. As shown in Fig. 1, the pendubot consists of a two-link planar robot arm (two coupled pendula) in the vertical plane, with a torque actuator at the shoulder (joint 1), but no actuator at the elbow (joint 2). The control problem of the pendubot typically refers to the task of stabilizing it in the upright position, which is an unstable equilibrium. One distinguishing feature of the pendubot is the infinite number of distinct equilibrium configurations (the first link at any angle, with the second link upright), in contrast to other benchmark examples of underactuated systems such as the pole-on-cart or the rotary (Furuta) pendulum. Despite its simple structure, the pendubot dynamics exhibit a range of properties that render it a useful testbed for studying a wide variety of topics. The latter include system identification, robot kinematics and dynamics, as well as different approaches (nonlinear, linear, fuzzy, optimal, hybrid and switching, gain scheduling) for control design. The pendubot system has also been used to study robotic throwing [8], [9].

A number of companies producing educational control model plants offer pendubot systems [10]–[13]. The controllers for these units are typically executed on a PC equipped with an appropriate data acquisition and control board, under real-time software extensions such as the Real

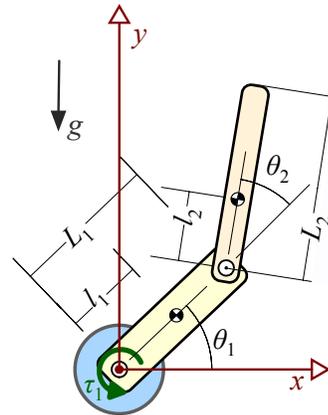


Fig. 1: Schematic diagram of the pendubot.

Time Windows Target (RTWT) for Simulink or the Real-Time module for Labview. This approach, often referred to as Rapid Control Prototyping (RCP), offers significant flexibility for implementing different control methods, since the software typically allows for automatic code generation from a block diagram description of the controller, as well as high sampling rates to limit performance degradation due to discretization. However, the overall costs for deploying such an RCP-based pendubot setup can be quite high.

As an alternative, we present here a low-cost, high-performance embedded control unit, developed for the stand-alone operation of a laboratory pendubot plant. The embedded controller utilizes an affordable 8-bit development platform (*Arduino Mega 2560*), along with a custom i/o board that interfaces the microcontroller with the pendubot plant. Employing a “bare-metal” coding approach allows frequencies as high as 1.25 kHz for the execution of the control loop, while serially streaming data for real-time viewing and/or post-processing of the system response.

Section II of the paper presents the dynamic modeling and control of the pendubot system. The design and construction of the laboratory pendubot plant is described in Section III, while Section IV presents the hardware and software of the developed embedded control unit. Finally, Section V provides experimental data demonstrating the operation of the system.

II. PENDUBOT MODELLING AND CONTROL

A. Modeling

A schematic diagram of the pendubot is provided in Fig. 1, where θ_1 denotes the angle of the first link with respect to the x -axis of the coordinate system, while θ_2 is the relative angle

The authors are with the Control Systems and Robotics Laboratory, School of Engineering, Technological Educational Institute of Crete, Heraklion, Greece. msfak@staff.teicrete.gr.

between the two links. Gravity is assumed to act in the negative y -axis direction. Defining the vector of generalized coordinates $\theta = [\theta_1 \ \theta_2]^\top$, and adopting a Lagrangian analysis approach, the dynamic model of the pendubot, assuming an ideal system with no friction at the joints, can be expressed in the following matrix form [1]:

$$D(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau, \quad (1)$$

where

$$D(\theta) = \begin{bmatrix} a_1 + a_2 + 2a_3 \cos \theta_2 & a_2 + a_3 \cos \theta_2 \\ a_2 + a_3 \cos \theta_2 & a_2 \end{bmatrix} \quad (2)$$

is the (symmetric and positive-definite) inertia matrix,

$$C(\theta, \dot{\theta}) = \begin{bmatrix} -a_3 \dot{\theta}_2 \sin \theta_2 & -a_3 (\dot{\theta}_1 + \dot{\theta}_2) \sin \theta_2 \\ a_3 \dot{\theta}_1 \sin \theta_2 & 0 \end{bmatrix} \quad (3)$$

is the matrix of Coriolis and centripetal terms,

$$G(\theta) = \begin{bmatrix} a_4 g \cos \theta_1 + a_5 g \cos(\theta_1 + \theta_2) \\ a_5 g \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (4)$$

is the vector of gravity terms, while the vector $\tau = [\tau_1 \ 0]^\top$ contains the torque τ_1 applied to the first joint of the mechanism. The parameters $a_1 \dots a_5$ that appear in (2)-(4) are defined in terms of the geometric and mass properties of the two links:

$$\begin{aligned} a_1 &= J_1 + m_1 l_1^2 + m_2 L_1^2 \\ a_2 &= J_2 + m_2 l_2^2 \\ a_3 &= m_2 L_1 l_2 \\ a_4 &= m_2 L_1 + m_1 l_1 \\ a_5 &= m_1 l_2 \end{aligned} \quad (5)$$

where m_i , L_i , l_i and J_i respectively denote the mass, overall length, distance of center-of-mass (see Fig. 1), and moment of inertia with respect to the center of mass, of the i th ($i = 1, 2$) link.

The absence of an external input torque at the second joint highlights the underactuated property of the system. Moreover, it imposes the following second order differential constraint:

$$\begin{aligned} (a_2 + a_3 \cos \theta_2) \ddot{\theta}_1 + a_2 \ddot{\theta}_2 \\ + (a_3 \dot{\theta}_1 \sin \theta_2) \dot{\theta}_1 + a_5 g \cos(\theta_1 + \theta_2) = 0, \end{aligned}$$

which renders the pendubot a second order nonholonomic system (see [5]).

B. State space formulation

Defining the system state as $\mathbf{x} = [\theta_1 \ \theta_2 \ \dot{\theta}_1 \ \dot{\theta}_2]^\top$, the pendubot equation of motion (1) can then be written as a standard nonlinear system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u, \quad (6)$$

where $u = \tau_1$ is the input torque applied to the shoulder motor, while

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{\theta} \\ -D^{-1}(\theta) (C(\theta, \dot{\theta})\dot{\theta} + G(\theta)) \end{bmatrix} \quad (7)$$

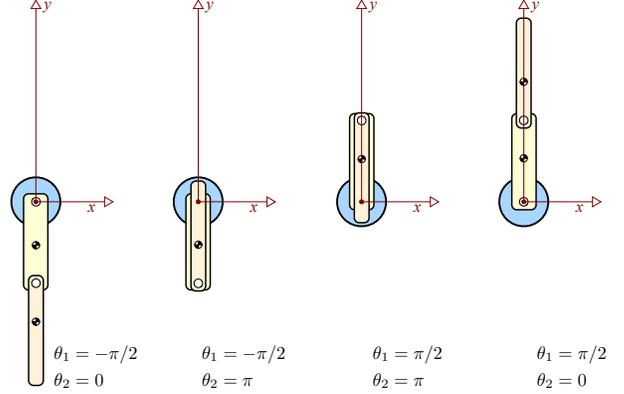


Fig. 2: The system equilibria for zero input torque.

and

$$\mathbf{g}(\mathbf{x}) = -D^{-1}(\theta) \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}. \quad (8)$$

C. Equilibrium configurations

The system equilibria can be found by setting

$$\dot{\mathbf{x}} = [\dot{\theta}_1 \ \dot{\theta}_2 \ \ddot{\theta}_1 \ \ddot{\theta}_2]^\top = [0 \ 0 \ 0 \ 0]^\top$$

in (6), which results in the following expression:

$$\begin{bmatrix} a_4 g \cos \theta_1 + a_5 g \cos(\theta_1 + \theta_2) \\ a_5 g \cos(\theta_1 + \theta_2) \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix} \quad (9)$$

i.e., $a_4 g \cos \theta_1 = u$ and $\theta_1 + \theta_2 = \pm \pi/2$.

The four equilibrium configurations for $u = 0$ (i.e., for zero input torque) are shown in Fig. 2. They are all unstable, with the exception of the one corresponding to $(\theta_1 = -\pi/2, \theta_2 = 0)$, i.e., when both links hang down vertically.

D. Linear balancing control

The Jacobian linearization of the equations of motion about an equilibrium point $(\mathbf{x}_{eq}, u_{eq})$, where \mathbf{x}_{eq} and u_{eq} respectively denote the equilibrium values of the state and the control input, yields the following model:

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x} - \mathbf{x}_{eq}) + \mathbf{B}(u - u_{eq}) \quad (10)$$

where, denoting as $\mathbf{h}(\mathbf{x}, u) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u$:

$$\mathbf{A} = \left. \frac{\partial \mathbf{h}(\mathbf{x}, u)}{\partial \mathbf{x}} \right|_{\mathbf{x}_{eq}, u_{eq}} \quad \mathbf{B} = \left. \frac{\partial \mathbf{h}(\mathbf{x}, u)}{\partial u} \right|_{\mathbf{x}_{eq}, u_{eq}} = \mathbf{g}(\mathbf{x}_{eq}) \quad (11)$$

Analytic expressions for the elements of the \mathbf{A} matrix and the \mathbf{B} vector can be found in [2]. It can be shown that the (\mathbf{A}, \mathbf{B}) pair is controllable for all equilibria of the pendubot, except for the ones corresponding to $\theta_2^d = \pm \pi/2$ (i.e., when the first link is aligned with the x -axis).

The linearized model can then be employed to design a full state feedback controller of the form

$$u - u_{eq} = -\mathbf{K}(\mathbf{x} - \mathbf{x}_{eq}), \quad (12)$$

for balancing the pendubot in the vicinity of a controllable equilibrium configuration (x_{eq}, u_{eq}) . The gain vector \mathbf{K} can be specified using standard linear control methods, such as pole placement or lqr design.

For discrete-time implementation of the state feedback controller, the equivalent gains, here denoted as $\tilde{\mathbf{K}}$, can be computed using the average gain method [14], according to the following transformation:

$$\begin{aligned} \tilde{\mathbf{K}} &= \mathbf{K} \left[\frac{1}{T_s} \int_0^{T_s} e^{(\mathbf{A}-\mathbf{BK})t} dt \right] \\ &\approx \mathbf{K} + \frac{T_s}{2} \mathbf{K}(\mathbf{A}-\mathbf{BK}), \end{aligned} \quad (13)$$

where T_s denotes the sampling interval.

III. THE PENDUBOT PHYSICAL SYSTEM

The experimental setup, developed for practical implementation of the pendubot control, is shown in Fig. 3. It consists of the pendubot physical plant, a servo amplifier, the embedded controller, and a power supply unit.

The pendubot mechanism is comprised of CNC-milled aluminium parts, which include the two links and a set of coupling hubs for the two joints. Both links are 5mm thick and 40mm wide, while the length of the first one is $L_1=180\text{mm}$, and that of the second one is $L_2=220\text{mm}$. To reduce friction, a ball bearing is installed on the elbow joint. Details regarding the mechanical design of the pendubot system can be found in [15].

A 48V brushed dc motor (*Pittman* model 14206-48) with a torque constant $k_t=0.141\text{Nm/A}$ and a stall torque rating of 2.05Nm, is used to actuate the shoulder joint. The motor is driven by a current amplifier (*TRM* model 10/100), capable of delivering up to 10A. The amplifier accepts a $\pm 10\text{V}$ analog command signal to set the motor current (a gain factor of $k_a=1.25\text{A/V}$ is used), and hence the torque applied to the actuated joint. Finally, the angular position of the shoulder joint is measured by a capacitive 2048cpr incremental encoder (*CUI* model AMT-103), and that of the elbow joint by an optical 1024cpr incremental encoder (*Kuebler* model 2400). With quadrature decoding, the effective resolution of the angle position measurement is 0.044° and 0.088° for the shoulder and elbow joint, respectively.

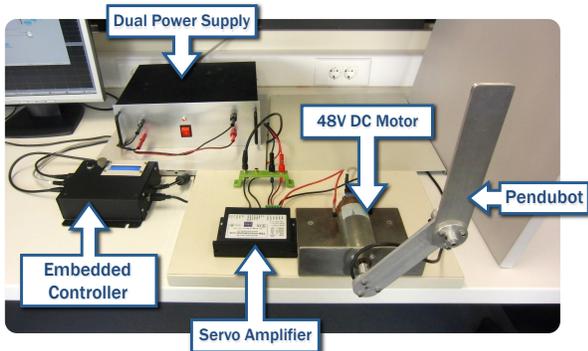


Fig. 3: Overview of the experimental setup.

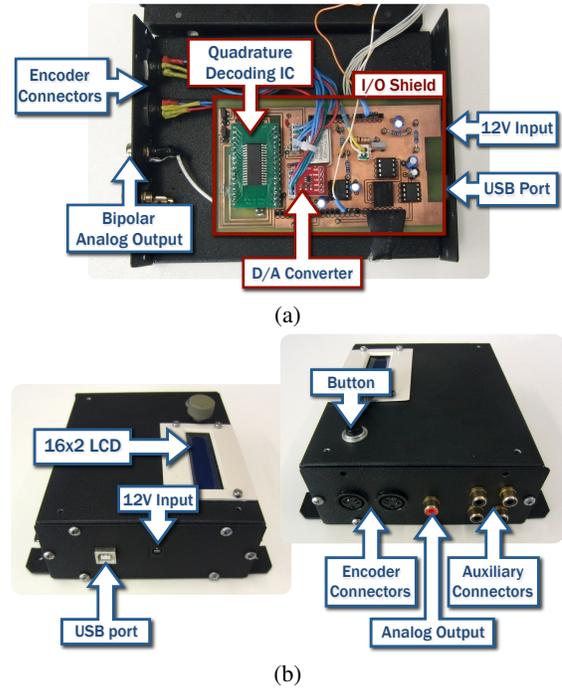


Fig. 4: The developed embedded control unit: (a) Main components, including the i/o shield. (b) External views.

Finally, a dual power supply provides 12V to the embedded control unit and 45V for driving the motor.

IV. THE EMBEDDED CONTROLLER

In this Section, we present the hardware and software design of the embedded control unit, which has been developed for implementation of the pendubot balancing scheme, presented in Section II-B.

A. Hardware design

The embedded control unit employs the Arduino Mega 2560 development platform, which is built around the *Atmel* ATmega2560 8-bit RISC microcontroller, clocked at 16 MHz. This processor features 256 kbytes of program memory, 6 hardware timers, a total of 54 digital input/output pins (14 of which can be used for PWM output), a 10-bit analog-to-digital converter, 4 hardware serial ports, as well as support for SPI and I2C communication. Due to its low cost, open-source design, and the compatibility of the supplied IDE with different operating systems, the Arduino platform has become very popular, and a host of hardware shields and software libraries are available in the public domain for interfacing with a wide variety of peripherals.

For this particular application, a custom i/o board was built as an add-on shield for the Arduino Mega 2560, to enable and facilitate interfacing with the pendubot system (Fig. 4a). In particular, the i/o board integrates a dedicated quadrature decoding chip (*Allegro* Hct1-2032) for efficient reading of the data from the two joint angle encoders, as well as an electronic circuit, utilizing a 12-bit digital-to-analog converter (*Microchip* Mcp-4725), to generate the bipolar

command signal sent to the servo amplifier to specify the torque applied to the shoulder joint of the mechanism. The printed circuit board (PCB) of the i/o shield was designed with the aid of *Cadsoft Eagle* software (Fig. 5).

The embedded controller, which is housed inside a metal enclosure (see Fig. 4b), is also equipped with an LCD screen, through which information is provided regarding system status. In addition, a pushbutton has been installed on the unit, for basic user interaction. The system is powered by 12V, required for generating the bipolar analog output.

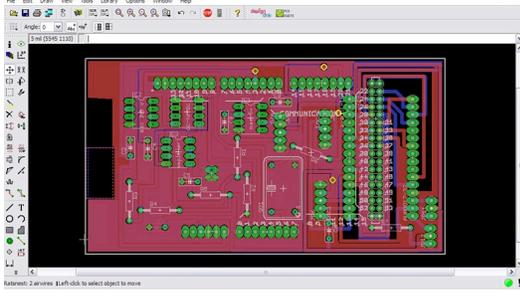


Fig. 5: Design of the i/o board PCB in *Cadsoft Eagle* software.

B. Design and digital implementation of balancing control

Following a series of identification tests with the pendubot system [15], the numerical values of the parameters in (5) were estimated as $a_1=0.0049$, $a_2=0.0019$, $a_3=0.0024$, $a_4=0.0308$, and $a_5=0.0131$ (note that the units for $a_1\dots a_3$ are Nm s^2 , while those for a_4 and a_5 are Ns^2).

The balancing control scheme was designed by considering three different equilibrium configurations, namely those corresponding to a desired shoulder joint angle of $\theta_1^d = \pi/3$, $\pi/2$, and $2\pi/3$. Substituting for the above numerical values of $a_1\dots a_5$, the three linearized models were obtained through (10), (11). For each of these models, the Matlab function `acker` was subsequently used to calculate the corresponding gain vector \mathbf{K} to place the eigenvalues of the continuous-time closed-loop system at $[-7.5+j2 \ -7.5-j2 \ -8.5 \ -8]$. These eigenvalue locations were selected after some experimentation with the real system. Subsequently, the discrete-time gain vectors $\tilde{\mathbf{K}}$ were calculated via (13), for a sampling time $T_s=0.8\cdot 10^{-3}\text{s}$. The thus derived gains for the three equilibrium configurations, along with the corresponding u_{eq} value are listed below:

$$\begin{aligned} \tilde{\mathbf{K}}_{\pi/3} &= -[1.2348 \ 1.3391 \ 0.2231 \ 0.1458], & u_{eq} &= -0.151 \\ \tilde{\mathbf{K}}_{\pi/2} &= -[0.7106 \ 0.9007 \ 0.1448 \ 0.0899], & u_{eq} &= 0 \\ \tilde{\mathbf{K}}_{2\pi/3} &= -[1.2348 \ 1.3391 \ 0.2231 \ 0.1458], & u_{eq} &= 0.151 \end{aligned}$$

Note that, due to symmetry, the gains for the $\theta_1^d = \pi/3$ equilibrium are the same as the ones for $\theta_1^d = 2\pi/3$, while the corresponding u_{eq} values are equal in magnitude but with opposite sign.

Implementation of the state feedback control law (12) requires that the angular velocities of the two joints are known. Since these cannot be directly measured, a filtered

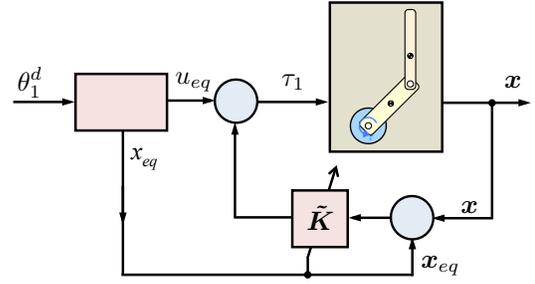


Fig. 6: Block diagram of the scheme for balancing control with gain scheduling.

numerical differentiation of the encoder data was implemented to estimate the values of $\dot{\theta}_1$ and $\dot{\theta}_2$ at each control loop iteration. In this approach, the estimated derivative $\hat{\dot{\theta}}(t)$ is obtained from the measured angular position $\hat{\theta}(t)$ according to:

$$\hat{\dot{\theta}}(t) = \frac{y(t) + \hat{\dot{\theta}}(t-T_s) + \hat{\dot{\theta}}(t-2T_s)}{3} \quad (14)$$

where $T_s=0.8\cdot 10^{-3}\text{s}$ is the sampling interval, and

$$y(t) = \frac{\hat{\theta}(t) - \hat{\theta}(t-T_s)}{T_s}. \quad (15)$$

The above filtering scheme was found to reduce, to an adequate extent, the noise arising from the numerical differentiation, without incurring excessive delay that could degrade the control performance. Nevertheless, this noise cannot be completely eliminated, and is the main limiting factor in increasing the control gains $\tilde{\mathbf{K}}$ to improve the transient response characteristics of the system.

As a final implementation step, the control torque u , calculated from the state feedback law (12), is used to specify the command signal v to the servo amplifier, according to:

$$v = \frac{u}{k_t k_a} \quad (16)$$

where k_t is the torque constant of the motor, and k_a is the gain factor of the servo amplifier.

C. Software design

The code running on the embedded controller allows for the stand-alone balancing operation of the pendubot, through implementation of the digital control strategy presented in the previous Section. To optimize performance, a “bare metal” coding approach was adopted, utilizing low-level register and interrupt programming of the ATmega2560 microcontroller. This allowed the control loop to run at 1.25kHz, paced by a dedicated hardware timer of the processor, while using single-precision float variables for the calculations.

For monitoring purposes, the developed firmware of the embedded controller outputs a serial stream of the pendubot’s joint positions (as measured by the encoders), as well as the reference angle θ_1^d and the control signal of the torque applied by the shoulder joint motor. The above data packets are sent in raw binary format, at a 312.5Hz-rate, over a

0.5Mbps serial connection, through the USB port of the Arduino Mega 2560. A Simulink model, running on a host PC under the Real Time Windows Target extension, was also developed to capture this serial data stream for real-time viewing and/or post-processing.

At startup, the pendubot should be at its lower (stable) equilibrium, to allow correct initialization of the joint encoders' output. Subsequently, the user manually brings the system in the vicinity of the upright position, by rotating it in the counter-clockwise direction, thereby prompting the activation of the balancing controller. The latter is invoked when the following condition holds for the angular position of the two joints of the mechanism:

$$(|\theta_1 - \theta_1^d| \leq 0.35\text{rad}) \wedge (|\theta_2 - \theta_2^d| \leq 0.45\text{rad}) \quad (17)$$

Using the pushbutton of the control unit, the user may then cycle through different reference positions for the shoulder joint, with the program altering accordingly the feedback gains $\tilde{\mathbf{K}}$, as well as \mathbf{x}_{eq} and u_{eq} (cf. Fig. 6).

The software implements a number of measures to ensure safe operation of the system. In particular, the configuration of the system is monitored whilst in balancing mode, and the motor is turned off if, e.g., due to a large disturbance, the first joint's angular position deviates significantly from the setpoint value, as per the following condition:

$$(|\theta_1 - \theta_1^d| \leq 1\text{rad}) \quad (18)$$

In addition, the motor is also disabled if there is a significant voltage drop in the 12V power input, that could cause the digital-to-analog circuitry on the i/o shield to issue the wrong command signal to the servo amplifier.

V. EXPERIMENTAL RESULTS

Indicative results, demonstrating the operation of the embedded controller, are provided in Fig. 8. Starting from the downwards position, the mechanism is manually brought near the upright configuration (i.e., for a reference value $\theta_1^d = \pi/2$ of the shoulder joint, with $\theta_2^d = 0$), to activate the balancing controller (indicated in the data by the initial spike in the control signal). The controller can be seen to maintain the pendubot at this reference configuration, being able to effectively reject a series of disturbances, which are manually introduced, at various time instances, by tapping the second link. The data also indicate a small oscillation in the steady-state response. This is attributed to the presence of friction, which is not explicitly included in the model and the control design. In addition, noise arising from the numerical differentiation of the encoder measurements, is evident in the control signal u .

The results shown in Fig. 7 demonstrate balancing of the system in the three different equilibrium configurations (designated by the different θ_1^d setpoints), the sequencing of which is determined by the user pressing the button of the embedded control unit. The small deviation present for $\theta_1^d = 2\pi/3$ and $\theta_1^d = \pi/3$ could be attributed to small errors in the specification of the $a_1 \dots a_5$ parameters, and/or to a biasing effect of the cable of the second joint's encoder. A

snapshot, showing balancing of the pendubot at the $\theta_1^d = \pi/3$ equilibrium configuration is provided in Fig. 9.

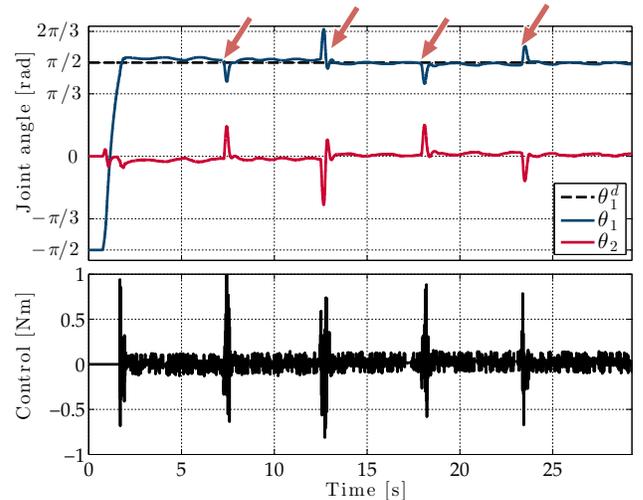


Fig. 7: Experimental results for balancing of the pendubot in the upright position. The red arrows indicate manually introduced (by tapping the upper link) disturbances to the system.

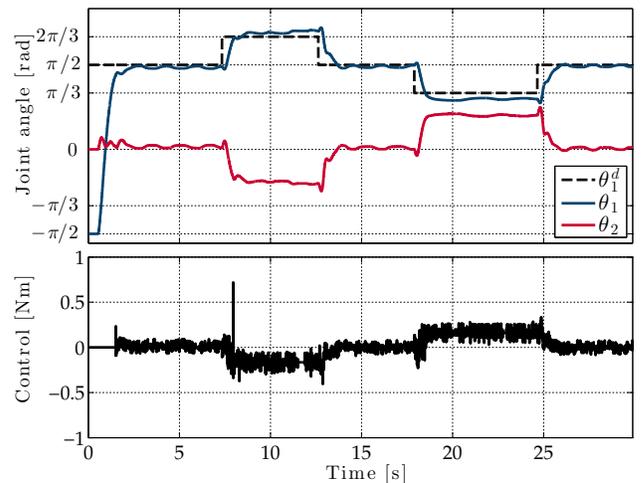


Fig. 8: Experimental results for the response of the system to a sequence of the three different equilibrium configurations.

VI. CONCLUSIONS

In this work, we have presented the integrated development of a digital control system for a laboratory plant of a pendubot system. Following the derivation of the system's dynamic model, a gain-scheduled full state feedback scheme was designed for balancing the system at a series of unstable equilibria. This was implemented as a stand-alone program that runs on an embedded hardware platform, developed around an 8-bit microcontroller. The experimental results demonstrate the efficacy of the digital control design, with the system exhibiting accurate setpoint regulation and effective disturbance rejection.

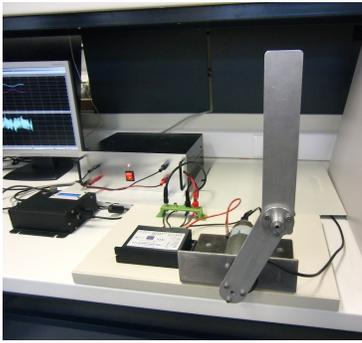


Fig. 9: Snapshot of balancing the pendubot at the $\theta_1^d = \pi/3$ equilibrium configuration.

One possible extension of the control program is the implementation of a “swing-up” strategy to automatically bring the system in the region of attraction of the linear controller. A variety of approaches can be used to this end, such as partial feedback linearization [1], and energy-based [4] or passivity-based [3] control. Since all of these schemes are more computationally-intensive than the linear state feedback balancing control law, implementation of the swing-up algorithm is likely to increase the control loop execution time.

We note here that it would be straightforward to employ the embedded unit for controlling other underactuated plants, such as the pole-on-cart or the Furuta pendulum. Both of these systems can be readily interfaced to the controller, as they have two degrees of freedom (typically using incremental encoders for joint position measurements) and a single actuation input (typically the torque generated by an electric motor). Systems furnished with analog sensors (e.g., potentiometers) can also be interfaced with the control unit, thanks to the 16-channel, 10-bit A/D converter of the ATmega2560 processor.

Moreover, by re-programming the firmware, the unit could be used as a low-cost USB data acquisition and control module, as part of a Rapid Control Prototyping scheme. In this approach, the main control algorithm would be executed on an external PC under an appropriate real-time framework, with the embedded unit only handling the quadrature decoding of the joint encoder signals and the D/A output, as a less expensive alternative to PCI- or PCIe-bus interface boards.

In view of the above options, the developed embedded control unit also represents an affordable and versatile learning platform for digital control systems, microcontroller programming, and related courses in engineering curricula.

REFERENCES

- [1] M. W. Spong and D. J. Block, “The pendubot: A mechatronic system for control research and education,” in *Proc. IEEE Conf. on Decision and Control*, pp. 555–557, 1995.
- [2] D. Block, *Mechanical Design and Control of the Pendubot*. M.s. thesis, University of Illinois at Urbana-Champaign, 1996.
- [3] I. Fantoni, R. Lozano, and M. W. Spong, “Passivity based control of the pendubot,” in *Proc. Amer. Contr. Conf. (ACC’99)*, vol. 1, pp. 268–272, 1999.

- [4] I. Fantoni, R. Lozano, and M. Spong, “Energy based control of the pendubot,” *IEEE Trans. Autom. Contr.*, vol. 45, no. 4, pp. 725–729, 2000.
- [5] M. Zhang and T.-J. Tarn, “Hybrid control of the pendubot,” *IEEE/ASME Trans. Mechatron.*, vol. 7, no. 1, pp. 79–86, 2002.
- [6] T. Albahkali, R. Mukherjee, and T. Das, “Swing-up control of the pendubot: An impulse-momentum approach,” *IEEE Trans. Robot.*, vol. 25, pp. 975–982, 2009.
- [7] A. Formal’skii, Y. Martynenko, and Y. Aoustin, “Pendubot: combining of energy and intuitive approaches to swing up, stabilization in erected pose,” *Multibody System Dynamics*, vol. 25, no. 1, pp. 65–80, 2011.
- [8] U. Mettin, A. S. Shiriaev, L. B. Freidovich, and M. Sampei, “Optimal ball pitching with an underactuated model of a human arm,” in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA’07)*, pp. 5009–5014, IEEE, 2010.
- [9] T. Shoji, S. Katsumata, S. Nakaura, and M. Sampei, “Throwing motion control of the springed pendubot,” *IEEE Trans. Contr. Syst. Technol.*, vol. 21, pp. 950–957, May 2013.
- [10] M. W. Spong, D. J. Block, and K. J. Astrom, “The Mechatronics Control Kit for education and research,” in *Proc. IEEE Int. Conf. Contr. Appl. (CCA’01)*, pp. 105–110, 2001.
- [11] PendCon, “Pendcon advanced: Double pendulum.” available online at <http://pendcon.de/double-pendulum.html>.
- [12] Quanser Consulting Inc., “SRV03 direct drive rotary plant with slipring: Position servo & planar self-erecting ip.”
- [13] Mechatronic Systems Inc., “Pendubot model p-1: User’s manual.”
- [14] D. L. Kleinman and P. Rao, “Continuous-discrete gain transformation methods for linear feedback control,” *Automatica*, vol. 13, no. 4, pp. 425 – 428, 1977.
- [15] M. Tsakiris, *Development and control of an underactuated robotic system (pendubot)*. Undergraduate thesis, Dept. of Electrical Engineering, Technological Educational Institute of Crete, 2010. [in Greek].