Design and Control of YAIP – an Inverted Pendulum on Two Wheels Robot

Johan Åkesson, Anders Blomdell and Rolf Braun

***Abstract*— In this paper we describe the design and control of an inverted pendulum type robot on two wheels. The objective of the design is to provide a flexible platform intended for teaching and research, which provides rich opportunities for application of signal processing, control design, distributed control systems and consideration of implementational issues. In addition, a design constraint has been to use low-cost components. Issues such as selection of hardware and sensors, signal processing, modeling and control are treated. Special attention is given to the problem of obtaining high accuracy velocity estimates using analog encoder signals.**

I. Introduction

Inverted pendula have been the subject of numerous studies in automatic control, from the forties and onwards. In this paper, a variation on the theme inspired by the well known Segway robot, [6], is described.

Inverted pendulums mounted on two wheels have been reported in several papers during the last years, see e.g. [5]. Also, there are a number of commercial robots on the market. The present paper describes the design and control of a prototype Segway-type robot, intended as a platform for research and teaching. The design problem is challenging, considering that it has been a primary objective to use components available at a reasonable price, while maintaining acceptable performance. Key design issues includes selection of drives and sensors as well as electronics and choice of microprocessors for signal processing and control.

The resulting robot offers several interesting features regarding sensors, control design, distributed control systems and implementation. The robot is equipped with two drives for actuation, a rate gyro and an accelerometer for measuring the angle and angular velocity of the pendulum body, and encoders for measuring the angle of the wheels. Signal processing and control algorithms are distributed amongst three microprocessors; one for each of the drives and one responsible for stabilizing control. This layout enables hierarchical control design, but also complicates implementation, since processor communication must be considered.

The purpose of the paper is to describe the robot design and to report experiences from the design process. The paper gives two main contributions. Firstly, a design description of the robot is given. Secondly, a novel approach to angular velocity estimation based on analog differentiation of encoder signals is presented.

The paper is organized as follows. In Section II, the robot design is briefly described. Section III treats selection of sensors and associated algorithms. In Section IV, a dynamic model of the system is presented. In Section V control strategies are described and in Section VI implementational issues are covered. Section VII describes results from experiments, and finally, in Section VIII, conclusions and final remarks are given.

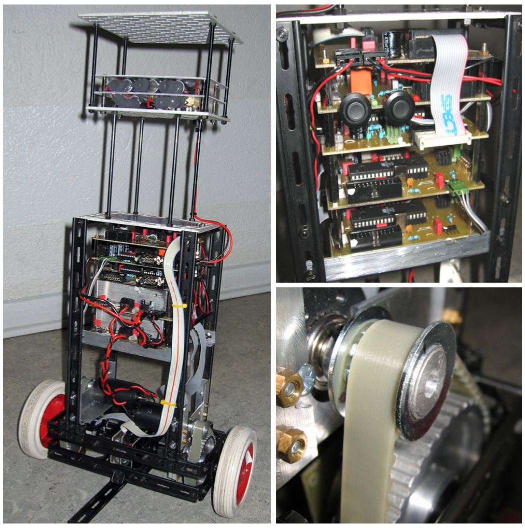

Fig. 1. YAIP – the inverted pendulum robot.

II. System Design

A schematic picture of the robot is shown in Figure 4. The robot consists of the pendulum body, which is attached to axes at which the two wheels are mounted. The pendulum body incorporates two DC-motor drives, transmission, sensors and several circuit boards hosting the micro-processors and sensor related electronics such as filters and amplifiers.

A. Mechanical Design

The actual pendulum body is built using FAC systems meccano/errecter set [3], complemented by some custom made aluminium parts, e.g. the parts used for mounting the drives. The robot is depicted in Figure 1.

B. Sensors and Actuators

In order to enable stabilizing control, the robot must be equipped with appropriate sensors and actuators. In fact, this issue constitutes perhaps the most challenging task in the robot design process. The choices concerning sensors and actuators, and associated algorithms, will inevitably impose constraints on achievable performance.

J. Åkesson, A. Blomdell and R. Braun are with the Department of Automatic Control, Faculty of Engineering, Lund University. Corresponding author: johan.akesson@control.lth.se

A pair of DC-drives were used to actuate the robot. Basically, the main trade-off is that between weight and torque, where higher torque comes at the price of more expensive and heavier drives. In order to increase torque, drives equipped with gear-boxes are attractive choices. A standards solution, which is also widely available, is then planetary gear-boxes. However, such devices introduce back-lash, which severely degrades control performance. This was confirmed by early designs in the project, as well as in [5]. Therefore, a gear-box was constructed using timing belts (ratio 4.1:1), which was found to effectively eliminate the back-lash. The drives used were a pair of Faulhaber 3863012C, which produced the maximum torque 0.45 Nm each, when connected to the gear-box. The choice of drives was guided by a preliminary simulation study.

The use of a rate gyro in combination with an accelerometer is a standard configuration for inertial sensing, which is needed in this application to estimate the angle and angular velocity of the pendulum body, see e.g. [8]. The selected accelerometer was an ADX202 and the rate gyro was an ADXRS300, both from Analog Devices. The ADX202 is a two axis accelerometer, although only one channel is needed in this application.

In order to measure the wheel angles, encoders were used. Since good angle and angular velocity information is crucial for stabilization of the system, the analog encoder signals were sampled at a frequency high enough to enable angle estimation based on the analog wave forms produced by the encoders. In addition, in order to further increase the accuracy of the angular velocity measurements, the derivative of the encoder signals were produced using analog filters. The differentiated signals were then sampled together with the original encoder signals. The idea of using analog differentiated encoder signals for velocity estimation is not new, see e.g. [4]. However, the approach presented in this paper, which is based on tabulated mappings calculated from the encoder wave-forms, has not, to the best knowledge of the authors, been reported previously.

C. Micro-processors

The micro-processor Atmel MEGA16 was chosen for signal processing and control algorithms. This processor is a convenient choice since it offers several important features on one chip, such as A/D conversion, RS232 communication, PWM-signal generation for motor control and a protocol for inter-processor communication, I2C. In addition, there is a free C-compiler for the AVR architecture, `avr-gcc`. The robot was equipped with three Atmel MEGA16 processors; one for each drive and one for stabilizing control and coordination.

III. Sensor Processing

A. Encoder Processing

1) Angle Estimation: The use of encoders is a standard method for estimation of angle and angular velocity. The idea is simple. By attaching a disc with alternating transparent and solid fields to the axis of rotation, and mounting a light source, commonly a LED, on one side of the disc and a photocell on the other, the latter component will produce a periodic waveform as the disc rotates with constant velocity. If two such pairs of a LED and a photocell are placed a quarter of a period apart, the angle of rotation can be calculated from the resulting waveforms. Alternatively, the LED and photocell pair may be placed on the same side of the disc, if a solid disc with light and dark fields is used. This configuration was used on the robot presented here. Commonly, the encoder signals are decoded digitally as logic high or low, which yields a resolution of twice the number of fields per disc revolution. This approach works well for high angular velocities, but generally poorly for low velocities. This is because in the latter case, few fields per time unit are passed which gives rise to severe quantization effects.

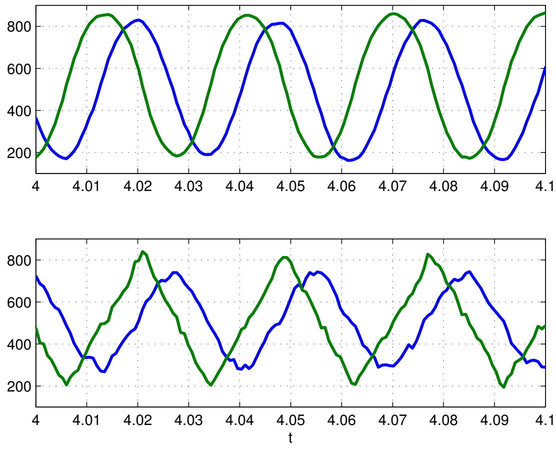

Fig. 2. Typical encoder waveforms. The upper plot shows the original encoder signals and the lower plot shows the signals obtained through analog differentiation.

A different approach, described in [10] and references therein, is to sample the analog encoder signals. Using this technique, the shape of the waveforms can be used to increase the resolution of the angle measurement significantly. It is a common misconception that encoder signals are given by

$$\begin{aligned}\bar{z}_x(\varphi) &= V_x \cos(\varphi N)\\ \bar{z}_y(\varphi) &= V_y \sin(\varphi N)\end{aligned} \tag{1}$$

where φ is the disc angle, $\bar{z}_x$ and $\bar{z}_y$ are the encoder signals and N is the number of solid (or transparent) fields of the disc. Mainly, there are two reasons for the encoder signals to be non-ideal. Firstly, the encoder signals are usually not shifted exactly by $\pi/2/N$ rad. Secondly, the waveforms often deviates from the ideal sinusoidal shape.

In order to increase the position estimation accuracy, these phenomena could be compensated for, e.g. using methods presented in [11] and [10]. We have used a method that explores the ideas presented in [10], and the method is described in summary in the following. We assume that the encoder waveforms are given by

$$\begin{aligned} z_x(\phi) &= g_x(\varphi N) = g_x(\phi),\ \ g_x(\phi) = g_x(\phi + 2\pi) \\ z_y(\phi) &= g_y(\varphi N) = g_y(\phi),\ \ g_y(\phi) = g_y(\phi + 2\pi) \end{aligned} \tag{2}$$

where ϕ is the encoder angle, related to the disc angle as $\phi = \varphi N$. Further, z_x corresponds to the cosine-like waveform and z_y corresponds to the sine-like waveform. See Figure 2 for typical waveforms.

Given measurements of z_x and z_y we would like to calculate an estimate of the angle, ϕ. In the ideal case, the expression

$$\hat{\phi} = \arctan\left(\frac{V_x \bar{z}_y}{V_y \bar{z}_x}\right) \tag{3}$$

gives the correct result. In the non-ideal case, however, the arctan-function will not produce a correct result. Now, instead of using the arctan-function, an equivalent mapping function may be used, given by

$$\hat{\phi} = q\left(\frac{z_y}{z_x}\right) \tag{4}$$

where q is calculated using the actual encoder waveforms rather than the ideal ones.

In order to calculate the function q, the actual encoder waveforms need to be estimated. This may be done using signal data recorded while rotating the encoder disc at constant velocity. By calculating the average of all recorded encoder periods, estimates $\hat{g}_x(\phi)$ and $\hat{g}_y(\phi)$, $0 \leq \phi \leq 2\pi$ may be obtained. The mapping function q is then calculated from

$$q^{-1}(\phi) = \frac{\hat{g}_y(\phi)}{\hat{g}_x(\phi)} \tag{5}$$

where $q^{-1}(\phi)$ denotes the inverse of $q(\phi)$. In Figure 3, the function $q(\phi)$ is shown in solid for a particular encoder configuration. The dashed curves shows the angle estimation given by expression (3). As can be seen, there is a significant difference between the angle estimates obtained if ideal curves are assumed, as compared to the case when the actual encoder waveforms are used as a basis for the estimation. Notice that in order for the mapping from z_y/z_x to $\hat{\phi}$ to be unique, the current quadrant, i.e. the signs of the encoder signals, must be considered.

2) Angular Velocity Estimation: In addition to accurate angle estimates, it is desirable to have accurate angular velocity information. The standard approach to this problem is to apply a discrete-time differentiation filter to the sequence of angle estimates. While this approach works well for high velocities, the performance for low velocities is poor. The main reason for this is that the temporal discretization, resulting from sampling, is difficult to compensate for using digital filters. Therefore we propose derivation of the encoder signals using analog filters. The analog derivative signals may then be sampled and used to improve the accuracy of the angular velocity estimate. The main advantage of this approach is that the need for differentiation by means of digital filters is eliminated. Instead, this approach enables velocity estimation by means of mapping-based methods similar to that presented above for angle estimation.

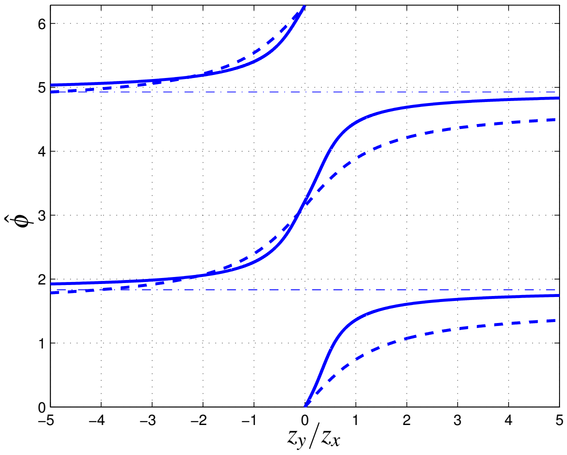

Fig. 3. The solid curves show the function $q(\phi)$. For comparison, the dashed curves shows the angle estimation given by expression (3).

Certainly, there are many ways to use the encoder signals in combination with the derivative signals to obtain angle and angular velocity estimates. In this paper a method producing promising results will be described. The main advantage of the proposed method is its simplicity and low computational complexity, which has been a key issue in this project. A different approach could be to extend the method presented in [11], which is based on an Extended Kalman Filter, to also include analog encoder derivative measurements.

Given the encoder waveform expressions (2), the time derivative signals may be written

$$\begin{aligned} dz_x(\phi,\dot{\phi}) &= \alpha_x \frac{dg_x}{d\phi}(\phi)\dot{\phi} = \alpha_x g_x'(\phi)\dot{\phi} \\ dz_y(\phi,\dot{\phi}) &= \alpha_y \frac{dg_y}{d\phi}(\phi)\dot{\phi} = \alpha_y g_y'(\phi)\dot{\phi} \end{aligned} \tag{6}$$

were α_x and α_y represents the amplification of the analog derivation filters. Using these expressions, estimates of the angular velocity may be calculated from

$$\begin{aligned} \hat{\dot{\phi}}_x &= \frac{dz_x}{\alpha_x g_x'(\phi)} \\ \hat{\dot{\phi}}_y &= \frac{dz_y}{\alpha_y g_y'(\phi)}. \end{aligned} \tag{7}$$

Clearly, $\hat{\dot{\phi}}_x$ and $\hat{\dot{\phi}}_x$ can be expected to be reasonable velocity estimates when, respectively, $g_x'(\phi)$ and $g_y'(\phi)$ are not close to zero. A natural way of combining the estimates is then

$$\begin{aligned} \hat{\dot{\phi}} &= \frac{g_x'(\phi)^2}{g_x'(\phi)^2 + g_y'(\phi)^2} \frac{dz_x}{\alpha_x g_x'(\phi)} \\ &+ \frac{g_y'(\phi)^2}{g_x'(\phi)^2 + g_y'(\phi)^2} \frac{dz_y}{\alpha_y g_y'(\phi)} \\ &= \frac{g_x'(\phi)}{g_x'(\phi)^2 + g_y'(\phi)^2} \frac{dz_x}{\alpha_x} + \frac{g_y'(\phi)}{g_x'(\phi)^2 + g_y'(\phi)^2} \frac{dz_y}{\alpha_y} \\ &= w_x(\phi) dz_x + w_y(\phi) dz_y \end{aligned} \tag{8}$$

where the weights $w_x(\phi)$ and $w_y(\phi)$ are readily calculated using the average waveforms $\hat{g}_x(\phi)$ and $\hat{g}_y(\phi)$. In the expression (8), the angle ϕ is assumed to be known, which is not the case in the real application. Instead, the estimate of the angle is used, which yields

$$\hat{\dot{\phi}} = w_x(\hat{\phi})dz_x + w_y(\hat{\phi})dz_y. \qquad (9)$$

3) Filtering: The estimates obtained from the algorithms described above are subject to noise. There are several sources of noise, including noise at the original encoder signals which is propagated to the estimated variables and noise introduced by the arithmetic operations. The latter source of noise may result if the algorithms are implemented using fixed-point arithmetics, which is common when computing power is scarce. Notice, for example, that a noisy angle estimate will propagate to the angular velocity estimate through the weights w_x and w_y.

In order to reduce the effects of noise, the estimated variables may be filtered. Assuming that a Kalman filter is used, there are two main different approaches for construction of the filter, differentiated by the structure of the underlying dynamic model. It could be argued that the use of a full dynamic model of the robot is advantageous, since it captures the full behaviour of the system. However, such a model may be very complex, and it may also contain severe nonlinearities, e.g. friction, that makes it difficult to use as a basis for a Kalman filter. Instead, a model expressing the kinematic relationships of the variables has been used. In [11], the following stochastic model is proposed

$$dx = Adt + dw = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\alpha \end{pmatrix} dt + dw \qquad (10)$$

where $x = (\phi, \dot{\phi}, \ddot{\phi})^T$, α is the inverse decorrelation time of the acceleration and dw is vector of zero-mean Wiener processes with incremental correlation matrix $R_{1c} = diag(0,0,\sigma^2)$. The continuous time model (10) is readily sampled using the sampling interval $h = 0.00146$ ms, yielding a discrete time equivalent

$$\begin{aligned} x(kh+h) &= \Phi x(kh) + w_d(kh) \\ y(kh) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} x(kh) + v(kh) \end{aligned} \qquad (11)$$

where $\Phi = e^{Ah}$ and the covariance matrix of w_d, R_1, is given by

$$R_1 = \int_0^h e^{A\tau} R_{1c} e^{A^T\tau} d\tau$$

The sampling expressions can be found e.g. in [2]. Also, the measurement noise process v has been introduced. Assuming that the covariance matrix of v, R_2, can be estimated from data, the Kalman filter has two tuning knobs; α and σ^2. These parameters can be used to set bandwidth of the filter – lower values of σ^2 and/or α, yields a filter with lower bandwidth. The calculation and implementation of the Kalman filter is described in [2]. The Kalman filter parameters α and σ^2 were set to 50 and 5×10^5.

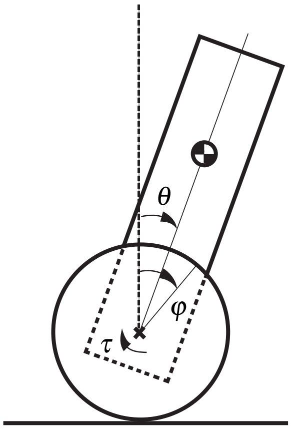

Fig. 4. A schematic picture of the pendulum robot.

A preliminary evaluation of the improvement of the angular velocity estimation produced by the proposed scheme showed improved precision compared to an estimate produced by a digital filter. However, in order to fully evaluate the properties of the method, more studies need to be performed. This, however, is beyond the scope of this paper.

B. Gyro and Accelerometer

The problem of obtaining accurate estimates of the angle and angular velocity of the pendulum body, θ and $\dot{\theta}$, is important in order to enable stabilizing control. The accelerometer, which is mounted on the pendulum body, takes into account the acceleration due to gravity, and may be used as an indicator of the angle of the body. This signal is, however, corrupted by high frequency disturbances. The rate gyro, on the other hand, produces a signal proportional to the angular velocity. While this signal has good high frequency properties, it suffers from drift. The two sensors in combination, however, may be used to obtain angle and angular velocity estimates.

A method developed for the combination of an inclinator and a rate gyro based on filters designed by shaping of frequency responses is presented in [7]. Alternatively, the problem may be addressed by postulating a Kalman filter based on the system

$$\begin{aligned} \dot{x} &= \begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \end{pmatrix} y_{gyro} \\ y_{acc} &= \begin{pmatrix} 1 & 0 \end{pmatrix} \end{aligned} \qquad (12)$$

where x_1 and x_2 $(x = (x_1, x_2)^T)$ represents the body angle, θ, and the gyro signal offset respectively. The bandwidth of the Kalman filter is tuned by adjusting the measurement and process noise covariances.

The choice of method does not seem to be critical – both methods work well in practice. However, the Kalman filter method was used since it gives an explicit estimate of the gyro offset.

IV. Dynamic System Model

Consider the schematic picture of the pendulum robot shown in Figure 4. Let the angle of the pendulum body relative to the vertical plane be θ, and let the angle of rotation of the wheels be φ. The applied torque is denoted by τ. Derivation of the equations of motion of the system by means of the Euler-Lagrange equations gives the system:

$$\begin{pmatrix} J_p & m_p l r_w \cos\theta \\ m_p l r_w \cos\theta & J_w + m_w r_w^2 + m_p r_w^2 \end{pmatrix} \begin{pmatrix} \ddot{\theta} \\ \ddot{\varphi} \end{pmatrix} + \begin{pmatrix} -m_p g l \sin\theta \\ -m_p l r_w \dot{\theta}^2 \sin\theta \end{pmatrix} = \begin{pmatrix} -\tau \\ \tau \end{pmatrix} \tag{13}$$

where $J_p = 0.16$ Nm is the moment of inertia of the body w.r.t. the pivot, $m_p = 2.94$ kg is the mass of the pendulum, $l = 0.168$ m is the distance from the pivot to the center of mass of the body, $m_w = 0.46$ kg, $r_w = 0.0515$ m and $J_w = 0.00045$ Nm are the mass, radius and moment of inertia of the wheel assembly respectively and $g = 9.81$ m/s^2 is the acceleration due to gravity.

The stabilizing control strategy described in Section (V) requires a linear state space model. Introducing the state vector $x = (\theta, \dot{\theta}, \varphi, \dot{\varphi})^T$, and linearizing the model (13) around $x^0 = (0,0,0,0)^T$ gives

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{\alpha\delta}{\alpha\beta-\gamma^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{\gamma\delta}{\alpha\beta-\gamma^2} & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ -\frac{\alpha+\gamma}{\alpha\beta-\gamma^2} \\ 0 \\ \frac{\beta+\gamma}{\alpha\beta-\gamma^2} \end{pmatrix} \tau \tag{14}$$

where $\alpha = J_w + m_w r_w^2 + m_p r_w^2$, $\beta = J_p$, $\gamma = m_p l r_w$ and $\delta = m_p g l$.

The input of the model (13) is the torque applied by the drives, τ. However, the actual control signal is the commanded voltage to the drives. A simple motor model relating torque, voltage and angular velocity of the motor was then introduced. Assuming that the current dynamics is fast compared to the rest of the system, the following relation holds

$$\tau = -\beta_1(\dot{\varphi} - \dot{\theta}) + \beta_2 u \tag{15}$$

where $\beta_1 = 0.015$ and $\beta_2 = 0.27$ are parameters calculated from the data-sheet of the drives, and u is the applied voltage. The negative term results from the back EMF produced by a DC-motor in motion. Notice that it is the *relative* angular velocity between the body and the wheels that gives rise to the back EMF.

V. Stabilizing Control

Since estimates of all states of the robot are available, state feedback is conveniently used to stabilize the system. The control law is then given by

$$u = -Lx. \tag{16}$$

where feedback vector L was calculated using LQR design, see e.g. [2].

In addition to the stabilizing controller (16), it was necessary to introduce an additional controller responsible for control of the robot heading. If the heading is not controlled, the robot will inevitable start rotating, resulting in lost stability. While the heading dynamics is not included in the model (13), the dynamics relating the difference in commanded voltages to the drives and difference in wheel velocities is well approximated by an integrator. A simple, yet effective, strategy is then to introduce a proportional controller acting on the difference in the angular velocities between the wheels,

$$u_d = -K(\dot{\varphi}_a - \dot{\varphi}_b) \tag{17}$$

where K is the controller gain and $\dot{\varphi}_a$ and $\dot{\varphi}_b$ are the angular velocity estimates of the two wheels respectively.

The control signals for the two motors are then given by the relations

$$\begin{aligned} u_a &= \frac{1}{2}(u + u_d) \\ u_b &= \frac{1}{2}(u - u_d). \end{aligned} \tag{18}$$

A. Friction Compensation

As most mechanical systems, the robot suffers from friction. In order to successfully stabilize the system, friction compensation proved necessary. There are several methods to compensate for friction, see, e.g. [9]. In this work, a friction compensation scheme based on Coulomb friction was used. The friction force may be approximated by the following model

$$F_f(\omega, u) = \begin{cases} F_c^+ & \dot{\omega} > 0 \\ F_c^+ & \dot{\omega} = 0, \quad u > F_c^+ \\ u & \dot{\omega} = 0, \quad F_c^- < u < F_c^+ \\ F_c^- & \dot{\omega} = 0, \quad u < F_c^- \\ F_c^- & \dot{\omega} < 0 \end{cases} \tag{19}$$

where $\omega = \dot{\varphi} - \dot{\theta}$ is the relative angular velocity between the body and the wheels. The coefficients F_c^+ and F_c^- are coefficients that may be calculated from simple experiments. An estimate of the friction force is then calculated using this model and added to the control signal computed from the control law (16). The control signal is then

$$v = -Lx + \hat{F}_f \tag{20}$$

VI. Implementational Issues

The signal processing and control algorithms described in previous sections were implemented in the Atmel AVR micro-processors mounted on the robot. The software was implemented in C, compiled using `avr-gcc`, and downloaded to the processors.

Since the computing power of the chosen AVR model is limited, floating point implementation of algorithms was infeasible, which rendered fixed point implementations necessary. This introduces additional overhead for the programmer, and may degrade algorithm performance. However, all algorithms were successfully implemented using fixed point arithmetics, with little performance loss. The most sensitive algorithms proved to be the encoder angle and angular velocity estimation, where a slight decrease in accuracy was noted.

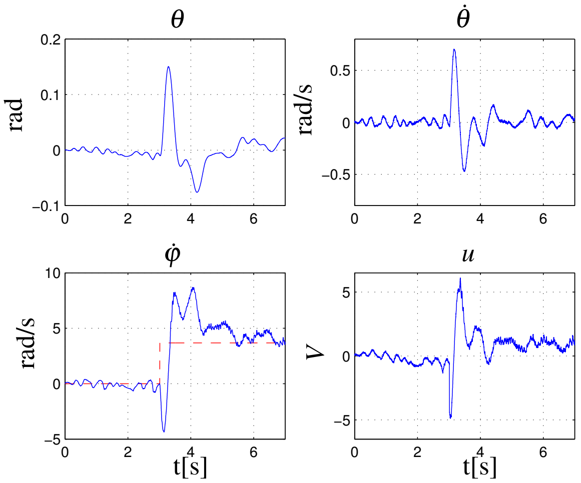

Fig. 5. Stabilizing control and system response for a start command.

Since the control system was distributed amongst three micro-processors, communication and synchronization was necessary. For this matter, the I2C protocol, which is supported by the Atmel AVR, was used. I2C is a serial bus-based communication protocol, which uses only two signal wires. Several units may communicate on the same bus, acting as masters and/or slaves. In this application, the processor responsible for stabilizing control acted as master and the two processors managing the encoders acted as slaves. During run-time, the master issues requests for measurement readings from, and transmits the commanded voltages to the slaves.

VII. Experimental Results

Using the control law (20), the system may be stabilized. As can be seen in Figure 5, up to 3 s, the robot is indeed successfully stabilized, although there is a small, but visible, remaining limit cycle in the wheel velocity, $\dot{\varphi}$. This limit cycle is due to friction, which is not fully compensated for by the friction compensation scheme.

In order to evaluate the system response to start and stop maneuvers, the modified control law

$$v = -l_1\theta - l_2\dot{\theta} - l_3(\dot{\varphi} - r) + \hat{F}_f \qquad (21)$$

was introduced, where r represents the reference wheel velocity. At run time, the control law (20) is used when $r = 0$ and (21) otherwise. Using this strategy, drift is avoided in the case of $r = 0$.

In Figure 5, a start command, represented by a step in the wheel velocity reference r (dashed), is initiated at 3 s. As can be seen, after an initial overshoot, the wheel velocity settles at the desired value. While the transient response of the system controlled by the controller (21) may be improved, it should be noted that introduction of reference signals requires special attention, in particular for unstable systems, see [1]. This, however, is beyond the scope of this paper.

VIII. Conclusions and Suggested Improvements

In this paper, design and control of an inverted pendulum robot on two wheels has been described. The aim of this project has been to build a flexible platform suitable for teaching and research using low cost components. During the course of the project, several preliminary designs were evaluated. The two most important improvements motivated by experiences from early attempts were the use of drives without planetary gear-boxes (which eliminated back-lash) and high resolution wheel angle measurements using analog encoder signals.

A main challenge facing the control system designer is that of obtaining accurate estimates of the states of the system, while designing a stabilizing controller is comparatively straight forward. This is reflected by the fact that the main part of this paper is devoted to state estimation algorithms.

The current design offers several opportunities for further improvements. Firstly, the encoder hardware design may be improved. The current implementation, based on reflection, is sensitive to irregularities of the reflecting surface. Also, if the distance between the LED/photo cell pair varies slightly, the resulting variation in the encoder signal offsets degrade the performance of the algorithms. Secondly, current control of the drives would be desirable, since this would effectively eliminate the uncertainties associated with these components. The resulting system would have a torque reference as input, which also gives a convenient structure of the control system. Another interesting feature would be a remote control facility, enabling a “driver” to maneuver the robot. Finally, more sophisticated control structures targeting manual control aspects would improve control performance, see e.g. [1].

References

[1] Johan Åkesson and Karl Johan Åström. Manual control and stabilization of an inverted pendulum. In *Proc. 16th IFAC World Congress*, Prague, Czech Republic, July 2005.

[2] Karl Johan Åström and Björn Wittenmark. *Computer-Controlled Systems*. Prentice Hall, 1997.

[3] FAC-system. Fac homepage, 2006. `http://www.facsystem.se/index.asp?lang=eng`.

[4] A. Gabor. Apparatus measuring relative velocity of movable members including means to detect velocity from the position encoder. Patent, 1974. Pat. no. US3839665.

[5] Felix Grasser, Aldo D’Arrigo, and Silvio Colombi. Joe: A mobile, inverted pendulum. *IEEE Transactions on Industrial Electronics*, 49(1):107–115, 2002.

[6] Segway Inc. Segway homepage, 2006. `http://www.segway.com/`.

[7] Baerveldt A. J. and Klang R. A low-cost and low-weight attitude estimation system for an autonomous helicopter. *Intelligent Engineering Systems, 1997. INES ’97. Proceedings., 1997 IEEE International Conference on*, pages 391–395, 1997.

[8] Grewal M.S., Henderson V.D., and Miyasako R.S. Application of kalman filtering to the calibration and alignment of inertial navigation systems. *Automatic Control, IEEE Transactions on*, 36(1):3–13, 1991.

[9] Henrik Olsson, Karl Johan Åström, Carlos Canudas de Wit, Magnus Gäfvert, and Pablo Lischinsky. Friction models and friction compensation. *European Journal of Control*, 1998.

[10] S. Venema. A kalman filter calibration method for analog quadrature position encoders. Master’s thesis, University of Washington, 1994.

[11] Y. Zimmerman, Y. Oshman, and A. Brandes. Improving the accuracy of analog encoders via kalman filtering. *Control Engineering Practice*, 14(4):337–350, 2006.