Mathematical Model and Control Strategy of a Two-wheeled Self-balancing Robot

Bernhard Mahler and Jan Haase
Institute of Computer Technology
Vienna University of Technology, Austria
Email: e0426066@student.tuwien.ac.at / haase@ict.tuwien.ac.at

***Abstract*—In this paper a control strategy and sensor concept for a two-wheeled self-balancing robot is proposed. First a mathematical model of the robot is derived using Lagrangian mechanics. Based on the model a full state feedback controller, in combination with two higher-level controls are deployed for stabilization and drive control. A gyroscope, an accelerometer and rotational encoders are used for state determination, introducing a new method of measurement data fusion for the accelerometer and the gyro by using a drift compensation controller. Furthermore measurement proceedings for the model parameters of a real prototype robot are suggested and the control for this robot is designed. The proposed mathematical model, as well as the control strategy are then verified by comparing the behavior of the constructed robot with model simulations.**

I. Introduction

For nearly two decades research and development of self-balancing vehicles is taking place. In 2001 the Segway, a self-balancing transportation system controlled by balance shift of the driver, was introduced and got commercially available [1]. It has been noticed by the broad public as a modern, futuristic vehicle. Although the Segway evolved over the last years as some kind of lifestyle product it was originally designed as a serious transportation system in urban areas. With the upcoming availability of low-cost, high performance MEMS sensor technology, the use of inertial sensors was no longer restricted to high end applications like avionics or space missions and the self-balancing concept also arrived at robotics.

Even though a two-wheeled self-balancing robot is an inherently unstable system which needs to be stabilized by electronic means, the two-wheeled movement can have also several benefits. Because of the reduced wheel count the robot can provide low footprints, compared to traditional systems and is also capable of turning around stationary for constricted area applications. Equipped with loading platforms or manipulators self-balancing robots, working in human controlled or even completely autonomous operation, might be used in industrial autnomous transportations systems or warehouses in the future.

From a more technical standpoint, a two-wheeled self-balancing robot is representing a mobile inverted pendulum, which is a classic problem in dynamics and control theory. One of the early day robots was "Joe" by Grasser et al., which used state space control based on a linear model derived with Newtonian mechanics [2]. In recent years intensive and diverse research is going on in this field and numerous robot concepts and control strategies have been proposed, ranging from linear state space controllers with pole placement [2] and LQR controller design [3], [4], to nonlinear and intelligent control methods like sliding mode control [5] or fuzzy control [6], [7]. The majority of these studies have their individual focus on trying to implement new control methods which have not been tested yet for self-balancing robots. For development of such a robot numerous further issues like motor drive and sensor characteristics also need to be considered in overall control design. Also the complexity of the control algorithms is important for efficient implementation on small microcontroller systems.

In this paper an in-depth overview, beginning with basic theoretical considerations and ending with a ready-to-implement control system, is given. In Section II a mathematical model is derived, using Lagrangian mechanics. For stabilization a well-proven linear state space control is used and for state variable acquisition a novel gyroscope-accelerometer sensor fusion method is proposed in Section IV. The model parameters of a real prototype robot are determined by using specifically designed measurement proceedings. The control is designed in Section V based on these model parameters. Finally the control system is implemented in the prototype robot to verify the effectiveness and robustness of the control strategy and state variable acquisition.

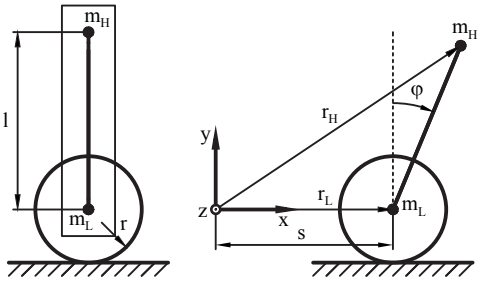

Fig. 1. Structure and definitions of the mathematical model

II. Mathematical Model

A. Overview

A two-wheeled self-balancing robot usually consists of the two wheels connected to a body frame holding the motor drive, the power and control electronics as well as some kind of battery. In order to keep the robot's behavior predictable by analytical means some abstractions of reality have to be made. Especially the nonuniformly distributed mass within the body has to be reduced to point masses.

Figure 1 shows the mechanical model of the robot used in this paper. It consists of the two point masses m_H and m_L connected to each other, having the distance l. The lower point

978-1-4799-0224-8/13/$31.00 ©2013 IEEE

mass m_L represents the motor drive and the upper point mass m_H represents the rest of the body frame. the two mass system is rotatable connected to the wheels at the lower mass point. The moment of inertia of the wheels J_W is also taken into account.

B. Equations of Motion

The equations of motion of the robot can be systematically obtained by using Lagrangian mechanics. With regard to the measurement based determination, the moved distance s of the lower point mass and the body's pitch angle φ are used as generalized coordinates q_1 and q_2 in terms of Lagrangian formalism.
The position of the two point masses from a stationary reference system as shown in Figure 1 can be described by the two position vectors

$$\mathbf{r}_H = [s + l\ sin(\varphi), 0, l\ cos(\varphi)]^T \tag{1}$$

$$\mathbf{r}_L = [s, 0, 0]^T \tag{2}$$

With the identities $v = \dot{s}$ and $\omega = \dot{\varphi}$, the kinetic energy of the two masses and the wheels are given by the equations (3), (4), and (5).

$$T_H = \frac{1}{2}\, m_H\, \dot{\mathbf{r}}_H^T\, \dot{\mathbf{r}}_H = \frac{1}{2}\, m_H\, (v^2 + 2\, l\, v\, \omega\, cos(\varphi) + l^2\, \omega^2) \tag{3}$$

$$T_L = \frac{1}{2}\, m_L\, \dot{\mathbf{r}}_L^T\, \dot{\mathbf{r}}_L = \frac{1}{2}\, m_L\, v^2 \tag{4}$$

$$T_R = \frac{1}{2}\, J_R\, \omega_R^2 = \frac{1}{2}\, J_R\, \left(\frac{v}{r}\right)^2 \tag{5}$$

Assuming that the potential energy of the mass points is zero for $y = 0$, the total potential energy of the two masses m_H and m_L is

$$V = m_H\, g\, l\, cos(\varphi) \tag{6}$$

In the mathematical model of the robot a velocity dependant rolling friction of the wheels, as well as a rotational friction, proportional to the angular velocity of the motor drive, are also taken into account. These dissipative processes can be described as Rayleigh's dissipation function of the following form [10]:

$$D = \frac{1}{2}\, \sum_i c_i\, \dot{q_i}^2 = \frac{1}{2}\, \left(c_v\, v^2 + c_\omega\, \left(\frac{v}{r} - \omega\right)^2\right) \tag{7}$$

The drive torque, which is trying to twist the wheels against the body, can be described as external force of the mechanical system. The generalized external forces can be written as:

$$\tau_\omega = -M_D \tag{8}$$

$$\tau_v = F_D = \frac{M_D}{r} \tag{9}$$

With the use of the Lagrangian $L = T - V$ and Lagrange's equation [10],

$$\frac{d}{dt}\, \frac{\partial L}{\partial \dot{q_i}} - \frac{\partial L}{\partial q_i} + \frac{\partial R}{\partial \dot{q_i}} = \tau_i \tag{10}$$

the nonlinear equations of motion for the robot (11), (12), and (13) can be obtained.

$$\frac{d\varphi}{dt} = \omega \tag{11}$$

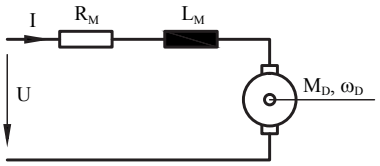

Fig. 2. Equivalent circuit diagram of the motor dive

$$\frac{d\omega}{dt} = \frac{\left((m_H + m_L)\, r^2 + J_R\right)\ (g - \omega\, v)\ sin(\varphi)}{l\ ((m_H + m_L)\, r^2 + J_R - m_H\, r^2\, cos^2(\varphi))} + \frac{\left((M_D + \omega\, c_\omega)\, r - c_v\, v\, r^2 - c_\omega\, v\right)\ cos(\varphi)}{l\ ((m_H + m_L)\, r^2 + J_R - m_H\, r^2\, cos^2(\varphi))} - \frac{((M_D + \omega\, c_\omega)\, r - c_\omega\, v)\ \left((m_H + m_L)\, r^2 + J_R\right)}{m_H\, l^2\, r\ ((m_H + m_L)\, r^2 + J_R - m_H\, r^2\, cos^2(\varphi))} \tag{12}$$

$$\frac{dv}{dt} = \frac{(M_D + \omega\, c_\omega)\, r^2\, cos(\varphi)}{l\ ((m_H + m_L)\, r^2 + J_R - m_H\, r^2\, cos^2(\varphi))} - \frac{(l\, r\, m_H\ (g - \omega\, v)\, sin(\varphi) + k_\omega\, v)\ r\, cos(\varphi)}{l\ ((m_H + m_L)\, r^2 + J_R - m_H\, r^2\, cos^2(\varphi))} + \frac{(M_D + \omega\, c_\omega)\, r - c_v\, v\, r^2 - c_\omega\, v}{(m_H + m_L)\, r^2 + J_R - m_H\, r^2\, cos^2(\varphi)} \tag{13}$$

C. Drive system

Several drive systems can be used for a self-balancing robot. A very simple system - in terms of driver complexity - would be two brushed permanent-magnet DC-motors with the wheels attached through appropriate gearboxes. The use of brushless DC-motors - geared or dircetly attached to the wheels - would also be possible. More complex electronics with space vector control of the brushless DC-motors would be needed for slow speeds and standstill of the robot. Another suitable method especially for smaller robots would be the use of stepper motors directly attached to the wheels.
For brushed permanent-magnet DC-motors a simple electronic model can be used. Figure 2 shows the equivalent circuit diagram of the motor drive. In this model the armature resistance R_M and inductance L_M, together with a motor torque constant k_M are used to describe the drive's dynamics. The equation of state for the motor current can be written as

$$\frac{di}{dt} = \frac{1}{L_M}\, \left(u_M - R_M\, i - k_M\, \left(\frac{v}{r} - \omega\right)\right) \tag{14}$$

and is linked to the mechanical state equations (12) and (13) by using the relationship

$$M_D = k_M\, i \tag{15}$$

Equations (11), (12), (13), and (14) are representing the complete nonlinear system state model of the robot and can be used for simulation and control design.

III. Control Strategy

The self-balancing robot is a variation of an inverted pendulum, which is a classic problem in dynamics and control theory. The control strategy proposed in this paper is a state space control combined with two higher level control loops for speed and differential speed of the two wheels. The differential speed control is needed for turning maneuvers and driving along curves. Figure 3 shows the control scheme of the robot.

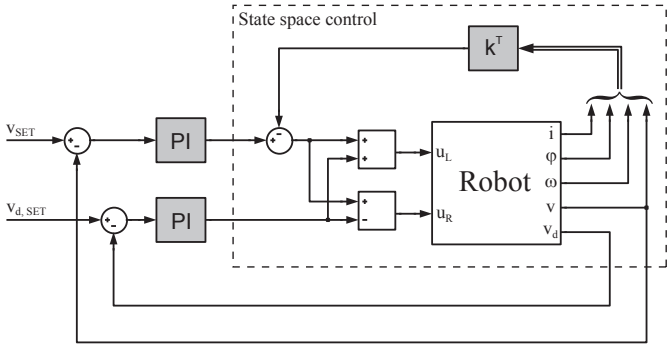

Fig. 3. Basic control scheme for the self-balancing robot

A. Linearization

In order to use linear control techniques, the mathematical model of the robot needs to be linearized at its upper rest position. The system state functions (11), (12), (13), and (14) have the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, where the vector $\mathbf{x}$ represents the state variables (φ, ω, v, i) and the vector $\mathbf{u}$ represents the input signal (u_M).
By linearizing the system, an equivalent linear model (16) can be obtained,

$$\begin{aligned} \mathbf{\Delta\dot{x}} &= \mathbf{A}[\mathbf{x} - \mathbf{x_0}] + \mathbf{B}[\mathbf{u} - \mathbf{u_0}] \\ \mathbf{\Delta y} &= \mathbf{C}[\mathbf{x} - \mathbf{x_0}] + \mathbf{D}[\mathbf{u} - \mathbf{u_0}] \end{aligned} \tag{16}$$

where the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ result from the Jacobians of the nonlinear state functions evaluated at the upper rest position of the robot $(\varphi_0 = \omega_0 = v_0 = i_0 = u_{M,0} = 0)$.

B. State Space Control

Without any kind of control mechanism the robot shows inherently unstable behavior, because its center of mass is above its pivot point. This can also be examined by determining the eigenvalues of the system matix $\mathbf{A}$ in (16). As long as the upper mass m_H is different from zero the system matrix has eigenvalues with positive real part and so the system is unstable.
A common method for stabilizing such a system is the use of a state space control. For stabilizing the robot in an upright position, full state feedback is used in this paper to place the closed loop poles, corresponding to eigenvalues of the system at desired locations in the s-plane. Full state feedback is accomplished by weighted feedback of the state variables $\mathbf{x}$ as shown in the following equation.

$$u = r - \mathbf{k^T}\,\mathbf{x} = r - (k_\varphi\,\varphi + k_\omega\,\omega + k_v\,v + k_i\,i) \tag{17}$$

For determination of the appropriate weighting coefficients of the feedback vector $\mathbf{k}$ either manual pole placement by using Ackermann's formula [11] or methods of the LQR (linear quadratic regulator) design can be used. Placing the poles of the system to a more left position on the s-plane will result in faster response of the system, but also higher values of the weighting coefficients which will lead the motor voltage to its range limits.
One major downside of full state feedback control is, that all state variables of a system have to be measured or at least derived from measured variables using some kind of state estimation structure. The acquisition of the necessary state variables for the robot is covered in section IV.

C. Motor dynamics simplification

The motor current dynamics will usually be significantly faster than the rest of the mechanical system. Controller design will result in a low feedback coefficient for the motor current, so the impact of the motor current on the system will be low. Using the limit of the function $L_M/R_M \to 0$ equation (14) can be rewritten as the algebraic equation

$$i = \frac{1}{R_M}\,\left(u - k_M\,\left(\frac{v}{r} - \omega\right)\right) \tag{18}$$

reducing the order of the dynamic system by the motor current. So including the motors current in state space control is not needed in most cases and therefore measurement of the motor current can be neglected.

D. Speed and Steering Control

In most scenarios drive speed and direction is commanded by some kind of higher-level or manual control. With the full state feedback applied, the robot will stay in an upright position and control deviation will be reduced but not brought zero. In order to achieve the set speed and direction two PI-controllers on top of the state feedback are applied, as also shown in Figure 3.
Steering of the robot is done by controlling the differential speed of the two wheels. Beside the control of driving turns, this is also necessary for driving straight ahead. In this case although the same voltage is applied to both motors they will typically run at slight different speeds leading the robot to drive unwanted curves.

E. Curve radius

In automotive engineering a common way of describing turns is the curve radius. For software implementation a relationship between the wheels speed difference and the curve radius is useful. Assuming the robot is driving along a curve with the radius r_c and the right wheel is at the inside of the curve and the two wheels having the distance b, the angular velocity around the centre of the curve is then given by

$$\omega_c = \frac{v}{r_c} = \frac{v_l}{r_c + \frac{b}{2}} = \frac{v_r}{r_c - \frac{b}{2}} \tag{19}$$

Using (19) and the relationships $v_l = v + v_d/2$ and $v_r = v - v_d/2$ the curve radius can be expressed by

$$r_c = \frac{b\,(v_l + v_r)}{2\,(v_r - v_l)} = b \cdot \frac{v}{v_d} \tag{20}$$

IV. Sensorical State Acquisition

For implementation of the full feedback state space control, described in section III, the acquisition of the state variables (body pitch angle φ, pitch angular velocity ω, linear velocity v, and motor current i) of the robot is needed.

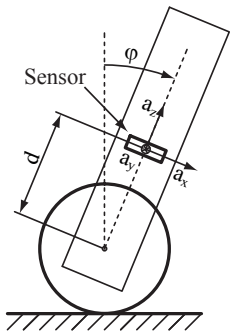

Fig. 4. Position of the accelerometer inside the robot

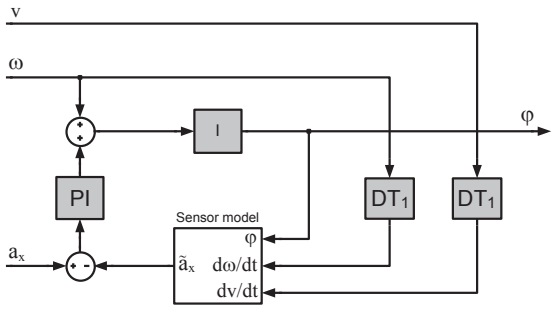

Fig. 5. Scheme for pitch angle determination

A. Motor current

Especially when highly dynamic behavior is demanded from the robot, including the motors current into the control strategy can be reasonable. Measuring the motor current can be accomplished with shunt resistors. If potential-free measurement is needed - especially when bridged motor drivers are used - also hall-effect based current sensors are available. Due to the use of motor bridge drivers the current of each motor will be measured separately. Under normal operating conditions ($v >> v_d$) the two motors will run at approximately equal speeds and so from a stabilization standpoint the two motors can be considered as one unit. The motor current input for the controller can therefore be calculated as the sum of the two individual motor currents.

B. Angular velocity

One of the key factors for stabilization is the accurate measurement of the angular pitch velocity ω. The measurement of the angular velocity can be done with a gyroscopic sensor attached to the body. A large variety of highly integrated MEMS-based gyroscopic sensors are available. One mayor quality feature of the sensor, especially needed for use in a self-balancing robot, is the insensitiveness against linear accelerations.

C. Velocity

The linear velocity v of the robot can be obtained form angular pitch velocity ω and the angular velocity of the two motor drives $\omega_{D,l}$ and $\omega_{D,r}$ using

$$v = (\omega_D + \omega)\ r = \left(\frac{\omega_{D,l} + \omega_{D,r}}{2} + \omega\right)\ r. \qquad (21)$$

For steering control the difference speed v_d is given by

$$v_d = (\omega_{D,l} - \omega_{D,r})\ r. \qquad (22)$$

Depending on the motor type, the motor speed ω_D can be measured with a rotary encoder attached to the motor shaft (for brushed DC-motors) or the drive speed is directly available from the motor control unit (for brushless DC-motors with space vector control or stepper motors).

D. Pitch angle - Sensor fusion

The main factor for stabilization of the robot is a dependable, noisefree pitch angle signal. The pitch angle typically has the highest weighting factor in state feedback.
Direct angular measurement with an accelerometer sensor is relying on vertical gravity. During acceleration maneuvers of the robot, horizontal acceleration and gravitational acceleration will add vectorially, resulting in wrong angular outputs. Another method would be the integration of the angular velocity signal from the gyroscope over time. The unavoidable offset in the gyro signal leads to a drift in the angle signal and makes this approach unusable.
In practice the acquisition of the pitch angle can be done by combination of the two sensors, benefitting from the drift free angular signal of the accelerometer as well as the noiseless, acceleration uninfluenced angular signal of the gyroscope. These kind of approach is also called sensor fusion and different implementations such as the use of IIR filters [8] or the Kalman filter [9] exist. The method proposed in this paper is based on drift compensation of the angular signal by comparing the output of the accelerometer sensor with the output of a mathematical sensor model.
The position of the accelerometer inside the body of the robot is shown in Figure 4. The acceleration $\widetilde{a}_x$ along the sensors x-axis is given by

$$\widetilde{a}_x = g\ sin(\varphi) + \dot{v}\ cos(\varphi) + \dot{\omega}\ d\ , \qquad (23)$$

where d is the distance of the sensor from the wheel axle.
As shown in Figure 5, the measured value from the accelerometer a_x is compared to the calculated value from the sensor model $\widetilde{a}_x$. Any drift in the angle signal φ, due to integration of the angular velocity signal ω, leads to a deviation between a_x and $\widetilde{a}_x$, which is compensated via the PI-controller. For a precise calculation of $\widetilde{a}_x$ also the time derivatives of ω and v are needed. They can either be obtained through state estimation, based on the model of the robot, or through derivative filters as shown in Figure 5.
Adjusting the response speed of the drift compensation control loop is critical for proper angle determination. If the speed is chosen too fast, noise from the accelerometer is fed through the angle signal. If the speed is too slow, drift from the gyroscope will not be compensated properly.

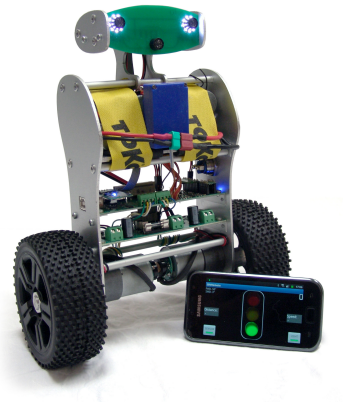

Fig. 6. Prototype of the robot with optional Smartphone control

V. Experimental Results

A. Prototype robot

A prototype robot was developed at Technical University of Vienna. Figure 6 shows a photo of the robot. It can be controlled either with a commercially available model sports radio control or with a Smartphone via Bluetooth. Two ultrasonic distance sensors in the head of the robot can be used for collision avoidance or rudimental autonomous operation.
The body is built up out of two aluminum sheets, joint together with aluminum tubes. The robot is equipped with two 12V brushed permanent-magnet DC-motors with attached gearboxes. The motors have a no-load speed of about 500RPM and a stall torque of about 0.6Nm (both on the wheel side of the gearbox). Two encoders for rotational speed measurement are directly attached to the motor shafts. The encoders have a resolution of 64CPR resulting in 1200CPR for the wheels.
The electronics is located in the middle part of the body. One PCB contains the motor drivers as well as a DC/DC converter for power supply. Also two hall effect based current sensors are integrated on the power PCB. The other PCB holds the microcontroller with peripherals, the gyroscope and the accelerometer as well as the Bluetooth module and USB in-system-programming circuitry. A 12V rechargeable battery pack is placed in the upper part of the body.

B. Model parameters

In order to design the control of the robot in a deterministic sort of way, the parameters of the mathematical model need to be determined. Nearly every parameter (except the total mass) can be calculated from measurements using the onboard sensors.
The static drive parameters R_M, k_M, and c_ω can be obtained by measuring the motor current and rotational speed at different voltages with no load and utilizing equation (14) as well as the equilibrium torque relationship (24) under static conditions.

$$M_D = k_M \cdot i = c_\omega \cdot \omega_D \tag{24}$$

The dynamic drive parameters L_M and J_R can be obtained by matching the measured step response of the motor current and the rotational speed with the responses of a simulated linear model of the free running motor/wheel combination. With the state vector $\mathbf{x} = [\omega_D, i]^T$ and the input vector $\mathbf{u} = [u_M]$ the motor/wheel system is given by the matrices

$$\mathbf{A} = \begin{bmatrix} -\frac{c_\omega}{J_R} & \frac{k_M}{J_R} \\ -\frac{k_M}{L_M} & \frac{R_M}{L_R} \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{L_M} \end{bmatrix} \tag{25}$$

The distance of the two masses l cannot be measured by using a ruler because the real mass is nonuniformly distributed in the body making it difficult to determine where the upper mass point actually is. The method suggested is to consider the robot as pendulum. By linearizing the pendulum's state equations at its lower rest point, the autonomous system is given by

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix} \tag{26}$$

with the state vector $\mathbf{x} = [\varphi, \omega]^T$. On stimulation this system is showing undampened oscillation with an angular frequency of

$$\omega_0 = \sqrt{\frac{g}{l}} \tag{27}$$

The angular frequency is only dependant on the length of the pendulum. By letting the robot swing while hanging downwards (without the wheels) the effective pendulum length l of the body can be calculated by measuring the angular velocity of the oscillation.
The isolated determination of the missing parameters m_H, m_L, and c_v by measurement is difficult due to heavy coupling of these parameters. Measuring the total mass $m_H + m_L$ is trivial. The best way found to determine mass distribution and c_v is to record the dampened swinging of the robot, while hanging downwards with the wheels rolling on two wooden strips and the motors blocked. The recorded angle and angular velocity waveforms are then matched with the ones form a simulated model trying to find the best fitting combination of the parameters.

TABLE I. Determined model parameters of the robot

Parameter	Value	
Body (Pendulum-)length l	0.147	m
Wheel radius r	0.056	m
Upper mass m_H	1.1	kg
Lower mass m_L	0.7	kg
Moment of inertia wheels J_R	$2.2 \cdot 10^{-3}$	kgm^2
Torque coefficient motor k_M	0.167	Nm/A
Armature resistance motor R_M	1.26	Ω
Armature inductance motor L_M	25	mH
Rotational friction coefficient c_ω	$2.6 \cdot 10^{-3}$	Nms/rad
Rolling friction coefficient c_v	2	Ns/m

C. Pitch angle tracking

As stated in the previous section, reliable determination of the robots pitch angle is essential for proper stabilization, so special attention in designing the drift compensation controller is recommended. When linearizing the control path in Figure 5 only the integrator remains. As there is no limiting factor for the controller output, the response time of the closed control loop can be set at any speed in principle.
Designing the controller for a rise time of 1s at maximum overshooting of 10%, led to a good tradeoff between noise

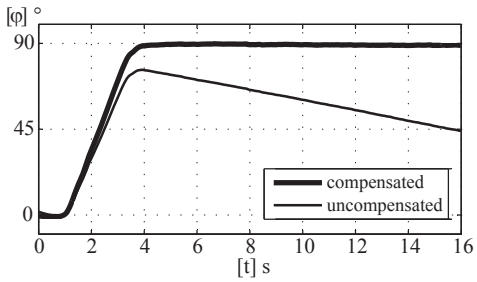

Fig. 7. Compensation of the angle drift caused by gyro offset

reduction and drift compensation, for the prototype robot. Figure 7 shows the angle obtained through integration of the angular velocity measured by the gyroscope with and without drift compensation when tilting the robot by 90 degrees.

D. State space controller design

Evaluating the linearized robot model (16) with the determined model parameters from Table 1 gives the system matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 119.1 & -0.4206 & 17.22 & -27.01 \\ -7.699 & 0.0457 & -2.244 & 2.938 \\ 0 & 6.68 & -119.3 & -50.4 \end{bmatrix} . \quad (28)$$

The system matrix has the eigenvalues $\lambda_1 = -29.1 + j5.06$, $\lambda_2 = -29.1 - j5.06$, $\lambda_3 = -3.25$, and $\lambda_4 = 8.38$, with λ_4 being unstable.
State space controller design was done by manual pole placement. Main goal of the controller design was to stabilize the robot, changing the overall system dynamics was not intended at this point. Placing the poles at

$$p_1 = -30s^{-1};\ p_2 = -30s^{-1};\ p_3 = -6s^{-1};\ p_4 = -3s^{-1} \quad (29)$$

led to satisfying results in terms of stabillity and response time with the prototype robot. Using Ackermann's formula the following feedback weighting coefficients were obtained:

$$k_\varphi = -15.3;\ k_\omega = -1.49;\ k_v = -6.33;\ k_i = 0.4 \quad (30)$$

Figure 8 finally shows the response of the stabilized prototype robot on a step sequence, compared to the model simulation. It should be noted, that there are some oscillations occuring on the prototype robot after sudden changes of direction. This is caused by backlash of the gearboxes, which is also bringing the robot to continuously slight oscillations during standstill. The deviations occouring between 3s and 5s are mainly caused by nonlinear friction and slipping effects between floor and tire after drive reverse, which are not taken into account by the mathematical model.

VI. Conclusion

In this paper a mathematical model and control strategy for a self-balancing robot was proposed. Despite the large number of model parameters, the output of the simulated model corresponds fairly well with the measured results of the real robot, as shown exemplarily in Figure 8. This indicates that the model proposed is capable of describing the robot's dynamic behavior in an appropriate way.
The prototype robot is able to accelerate to its maximum speed

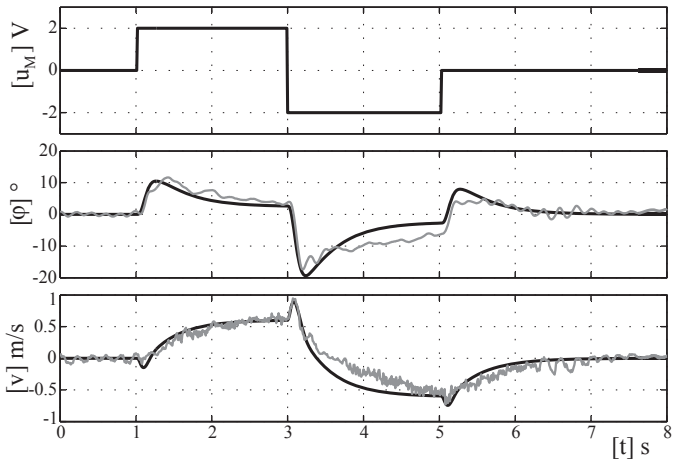

Fig. 8. Simulation and measured step response of stabilized robot

of $2m/s$ in about $1s$, and also strong manual disturbance or uneven terrain (e.g. lawn) are handled without problems. So the deployed control strategy can also be used in applications where high dynamics and robust behavior is required.
For further improvement of the driving dynamics the impact of driving curves on the stabilization process is beeing analyzed. Therefore a more sophisticated model is needed, which also takes upright rotations of the robot and movement in three-dimensional space into account. Another starting point for improvements would be the capability of the robot on adapting to different payloads for transporting goods. Such an adaptive control could maybe implemented by some higher level fuzzy logic, adjusting the state control parameters.

References

[1] Segway Inc. Website. [Online]. Available: http://www.segway.com/

[2] F. Grasser, et al., "JOE: a mobile, inverted pendulum", IEEE Transactions on Industrial Electronics, vol. 49, pp. 107-114, Feb. 2002.

[3] V. Kongratana, et al., "Servo State Feedback Control of the Self Balancing Robot using MATLAB", 12th International Conference on Control, Automation and Systems (ICCAS), pp. 414-417, Oct. 2012.

[4] K. Peng, et al., "Dynamic Model and Balancing Control for Two-Wheeled Self-Balancing Mobile Robot on the Slopes", 10th World Congress on Intelligent Control and Automation (WCICA), pp. 3681-3685, Jul. 2012.

[5] F. Dai, et al., "Development of a Coaxial Self-Balancing Robot Based on Sliding Mode Control", International Conference on Mechatronics and Automation (ICMA), pp. 1241-1246, Aug. 2012.

[6] C.-H. Huang, et al., "Velocity control realisation for a self-balancing transporter", IET Control Theory & Applications, vol. 5, issue 13, pp. 1551-1560, Feb. 2011.

[7] J. Wu, W. Zhang, "Design of Fuzzy Logic Controller for Two-wheeled Self-balancing Robot", 6th International Forum on Strategic Technology (IFOST), vol. 2, pp. 1266-1270, 2011.

[8] H.-W. Lee, "Optimal posture of Mobile Inverted Pendulum using a single gyroscope and tilt sensor", ICROS-SICE International Joint Conference, pp. 865-870, Aug 2009.

[9] H.-J. Lee, S. Jung, "Gyro Sensor Drift Compensation by Kalman Filter to Control a Mobile Inverted Pendulum Robot System", IEEE International Conference on Industrial Technology, pp. 1-6, 2009.

[10] L. Meirovitch, "Methods of Analytical Dynamics", pp. 72-91, New York: McGraw-Hill, 1970.

[11] J. Ackermann, "Der Entwurf linearer Regelungssysteme im Zustandsraum", Regelungstechnik, vol. 20, pp. 297-300, 1972, in German.