

Comparison of two efficient control strategies for two-wheeled balancing robot

Michał Majczak, *Student, Member IEEE*, and Paweł Wawrzynski, *Member IEEE*

Institute of Control and Computation Engineering

Warsaw University of Technology

00-665 Warsaw, Poland

m.majczak@stud.elka.pw.edu.pl, p.wawrzynski@elka.pw.edu.pl

Abstract—The subject of this paper is a two-wheeled balancing robot with the center of mass above its wheels. Two control strategies for this robot are analyzed. The first one combines a kinematic model of the robot and a PI controller. The second one is a cascade of two PIDs. These strategies are compared experimentally.

Keywords—mobile robots, inverted pendulum, cost-effective robots.

I. INTRODUCTION

Two motors are necessary and sufficient in a robot to make it move in any given direction on the ground. Each motor is attached to a wheel which makes one of the two robot's supporting points. A robot may have an additional, non-powered wheel (or wheels) to have three (or more) supporting points. Alternatively, a robot may have just two powered wheels. Having no additional supporting point, it has simpler mechanical design but it has to have more elaborate control system, since such a robot is not statically stable.

In this paper we analyze the simplest possible mobile robot architecture: with two wheels powered by two motors, no additional support, and the center of mass above the wheels. The robot is build of standard low-cost components that ensure modularity of the construction making any changes of it relatively easy.

The most known two-wheel balancing robot is Segway™ [1]. A number of control strategies have been applied to such robots, including PID backstepping [2], LQR with neural networks [3], LQR with PID [4] and fuzzy models [5]. Also, a number of specific designs of such robots have been implemented and modeled [6], [7], [8], [9].

In this paper two control strategies for a two-wheel balancing robot are analyzed. The first one combines a PI controller that transforms robot's velocity error into its desired tilt. The tilt is assured by a mechanism designed on the basis of the robot's dynamics. The second control strategy is a cascade of two PIDs. The upper PID determines the desired tilt on the basis of the velocity error, and the lower PID keeps tilt close to the desired. Those strategies are compared experimentally.

The structure of this paper is as follows: Section II specifies the requirements of the robot and its design. Section III presents the dynamic model of the robot. Sections IV and V

present its control systems based on, respectively, dynamics and a cascade of PIDs. Section VI presents experiments that demonstrate the robot's performance, and the last section contains concluding remarks for the paper.

II. ROBOT DESIGN

A. Specification

The purpose of the robot is remote presence with limited resources engaged:

- The robot traverses the ground in the direction given by commands from a human operator. Those are given through a gamepad (basically, a pair of joysticks) via a wireless connection.
- The robot transmits a video stream from its camera to the operator through a wireless connection.
- The robot is build of as few and as simple components as possible.

B. Design

The robot is presented in Fig. 1. It is built of the following components:

- Two servomotors RX-24F.¹ Max torque: 26 kg·cm. Max speed: 125 rpm.
- Several plastic moulders taken from Bioloid Comprehensive Kit.²
- Altronit BB2 controller,³ a small computer with Linux UBuntu OS, Wi-Fi, RS-485 bus.
- Inertial sensors (actually being parts of the controller): accelerometer LIS3DSH⁴, gyroscope L3G4200D⁵.
- Li-Po battery pack of 11.1V and 2.5Ah.

III. DYNAMICS

In order to synthesize a control system for the robot, we first derive a model of its dynamics. It is assumed here that rotating the robot around the vertical axis is a trivial activity that results from a difference between the wheels' angular velocity. The actual problem here is to keep the robots body vertical to the

¹Manufacturer: Robotis.

²Manufacturer: Robotis.

³Manufacturer: Altronit

⁴Manufacturer: STMicroelectronics

⁵Manufacturer: STMicroelectronics

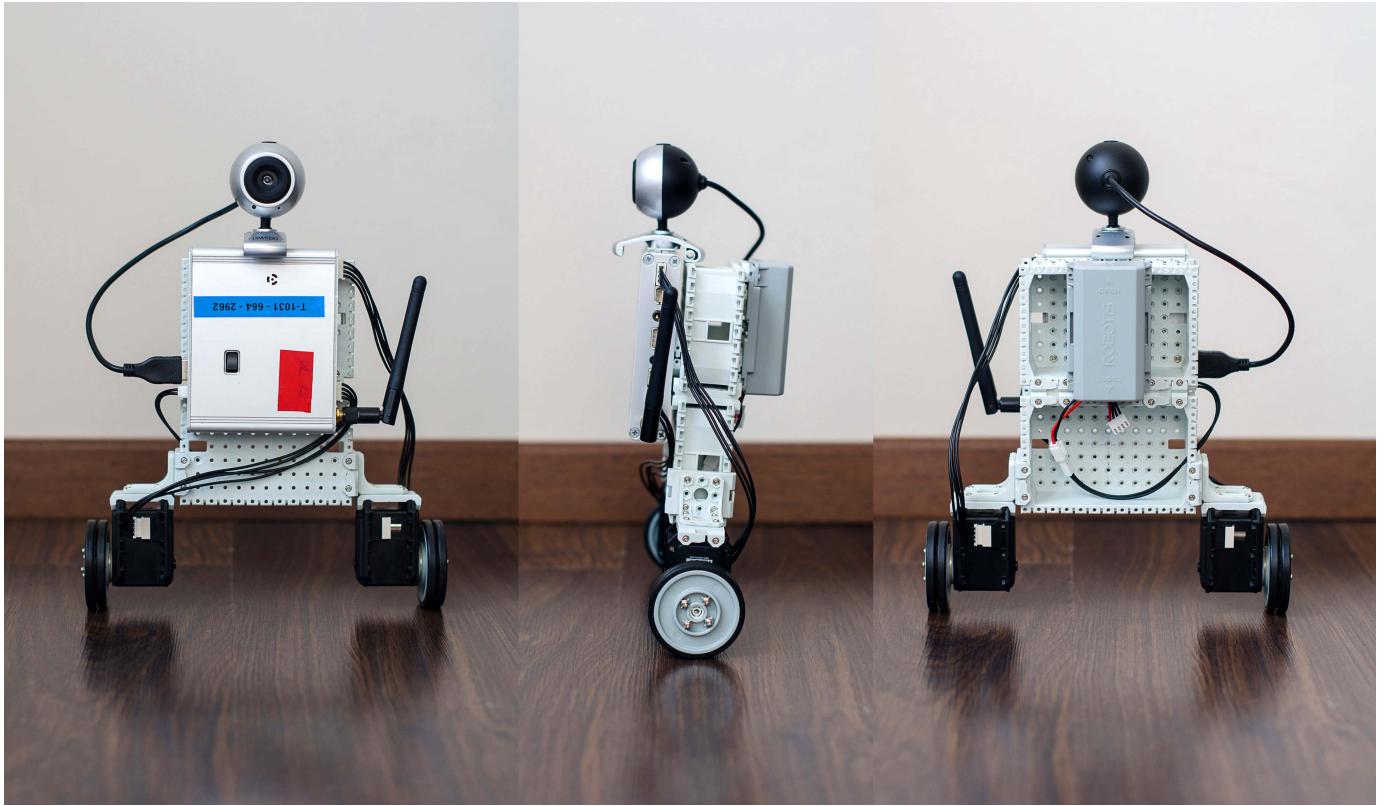


Fig. 1. The robot

ground through appropriate average angular velocity of the wheels.

In order to solve that problem, the robot is analyzed as an inverted pendulum (Fig. 2). Its mass is located in two points: in the center of the wheels, and in the center of the body.

A. Assumption and Notation

The model below is based on the following assumptions:

- The wheels rotate without slippage;
- The whole robot's mass is concentrated in two points: x_1 and x_2 . These masses are, respectively, m_1 and m_2 ;
- The radius of the wheels is r , and their total moment of inertia is I_k ;
- Friction in the system is negligible.

Notation:

- x — the position of the wheels' center on horizontal axis ($x = x_2$ in Fig. 2),
- φ — the angle between the robot's trunk and the vertical axis,
- M_s — the angular velocity of the wheels in reference to the robot's body.

B. Derivation of dynamics

The relation between \dot{x} and $\dot{\varphi}$ takes the form

$$\dot{\varphi} = \dot{x}/r - M_s. \quad (1)$$

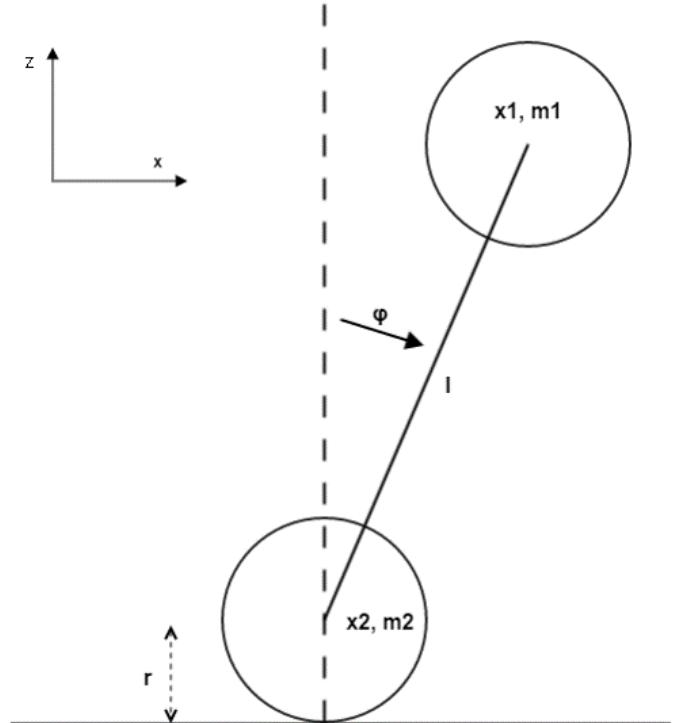


Fig. 2. Inverted pendulum as the robot model.

Next, the axis of the wheels is analyzed as the axis about which the robot rotates. Its angular momentum is equal to

$$Q = I_k \dot{x} / r + m_1 l^2 \dot{\varphi} \quad (2)$$

$$= I_k \dot{x} / r + m_1 l^2 (\dot{x} / r - M_s). \quad (3)$$

Derivative of the angular momentum with respect to time is equal to torque applied to the robot by Earth gravity i.e.,

$$\dot{Q} = I_k \ddot{x} / r + m_1 l^2 (\ddot{x} / r - \dot{M}_s) = \sin \varphi m_1 g l. \quad (4)$$

Therefore,

$$\ddot{x} (I_k / r + m_1 l^2 / r) = m_1 g l \sin \varphi + \dot{M}_s m_1 l^2. \quad (5)$$

Using (1) to replace \ddot{x} in the above equation we obtain

$$(\ddot{\varphi} + \dot{M}_s) r (I_k / r + m_1 l^2 / r) = m_1 g l \sin \varphi + \dot{M}_s m_1 l^2 \quad (6)$$

$$\ddot{\varphi} (I_k + m_1 l^2) = m_1 g l \sin \varphi - \dot{M}_s I_k. \quad (7)$$

The last of the above equations allows to directly set $\ddot{\varphi}$ by imposing appropriate \dot{M}_s i.e., increase or decrease of M_s .

IV. CONTROL BASED ON DYNAMICS

The objective of control is to set forward speed \dot{x} close to the desired value, \dot{x}^d , given on-line by the human operator. The general idea of control is to tilt the robot in the direction indicated by $\dot{x}^d - \dot{x}$. That is, the desired φ is to be proportional to the aforementioned difference:

$$\varphi^d \propto \dot{x}^d - \dot{x}. \quad (8)$$

In order to make φ close to φ^d , $\dot{\varphi}$ is set (with the use of (7)) according to

$$\dot{\varphi} \propto \varphi^d - \varphi. \quad (9)$$

In details, the control system works as follows

$$\varphi^d = K_p (\dot{x}^d - \dot{x}) + K_i \int_0^t (\dot{x}^d - \dot{x}) dt', \quad (10)$$

$$\dot{\varphi}^d = \frac{1}{\tau_1} (\varphi^d - \varphi) \quad (11)$$

$$\ddot{\varphi} = \frac{1}{\tau_2} (\dot{\varphi}^d - \dot{\varphi}) \quad (12)$$

The integral in (10) enables to balance the robot when a value $\varphi \neq 0$ is required to keep constant \dot{x} . K_p and K_i are appropriate gains and τ_1 , τ_2 are small time constants that define the speed of φ approaching φ^d and $\dot{\varphi}$ approaching $\dot{\varphi}^d$. Eqs (7) and (12) determine how M_s should be adjusted for $\ddot{\varphi}$ to have appropriate value.

V. CONTROL BASED ON A CASCADE OF PIDS

An analysis of the model presented in the previous section leads to the following idea of the robot control:

- In order to accelerate/decelerate the robot, it has to be tilted in an appropriate direction.
- The robot is tilted by applying appropriate angular velocity in its wheels.

The above idea translates into a control system sketched in Fig. 3. It is a cascade of two PID controllers. The first one

produces the robot's target tilt on the basis of the difference signal between the robot's target speed, and the current one. The second controller produces the target drive speed on the basis of the difference signal between the robot's target tilt and the actual one.

The actual tilt of the robot is computed with the use of the accelerometer, the gyroscope, and the complementary filter. A current tilt estimate, $\hat{\varphi}(t)$, is computed as

$$\hat{\varphi}(t) = \alpha (\hat{\varphi}(t - \delta) + \delta \omega) + (1 - \alpha) \varphi^A, \quad (13)$$

where δ is the timespan between two measurements, ω is the angular velocity measurement taken by the gyroscope, α is a coefficient of weighted sum and φ^A , is the tilt computed on the basis of the accelerometer measurement and the assumption that this sensor measures only its inclination (that is the gravity vector in the frame attached to the sensor).

Ordinary PID controllers produce control, u , as follows:

$$u = K_p (x^d - x) + K_I \int_0^t (x^d - x) dt + K_d (\dot{x}_d - \dot{x}), \quad (14)$$

where x is the actual value of the signal, x^d is its target value, and K_p , K_I , K_d are coefficients. Thank to derivative component, the controller reacts immediately to changes in the target signal, which produces unwanted "shaky" behaviour when the target signal is given manually by a human operator. Instead, a modified PID controller is applied that assumes $\dot{x}_d \equiv 0$, and produces u in the form

$$u = K_p (x^d - x) + K_I \int_0^t (x^d - x) dt - K_d \dot{x}. \quad (15)$$

VI. EXPERIMENTS

In this section experiments with the robot are reported. Subjectively, the control systems work as expected, the robot moves in accordance to the direction set remotely by a human operator. Several experiments presented below give quantitative demonstration of that.

A. Balancing at zero target speed

In the first experiment it is verified how the robot behaves given zero target velocity. The results are presented in Fig. 4. It is seen that velocity and tilt of the robot slightly oscillate in both control methods. The robot cannot be absolutely immobile since its state is never statically stable.

B. Rotating about vertical axis

Another experiment presents the robot not moving forward or backward, and rotating about the vertical axis. Within this motion, whenever the center of the body mass is not exactly over the axis of the wheels, a centrifugal force sets in and tends to lean the robot even more. This extra force is handled by the control systems as a disturbance. Fig. 5 shows that it is handled appropriately and similarly in both cases.

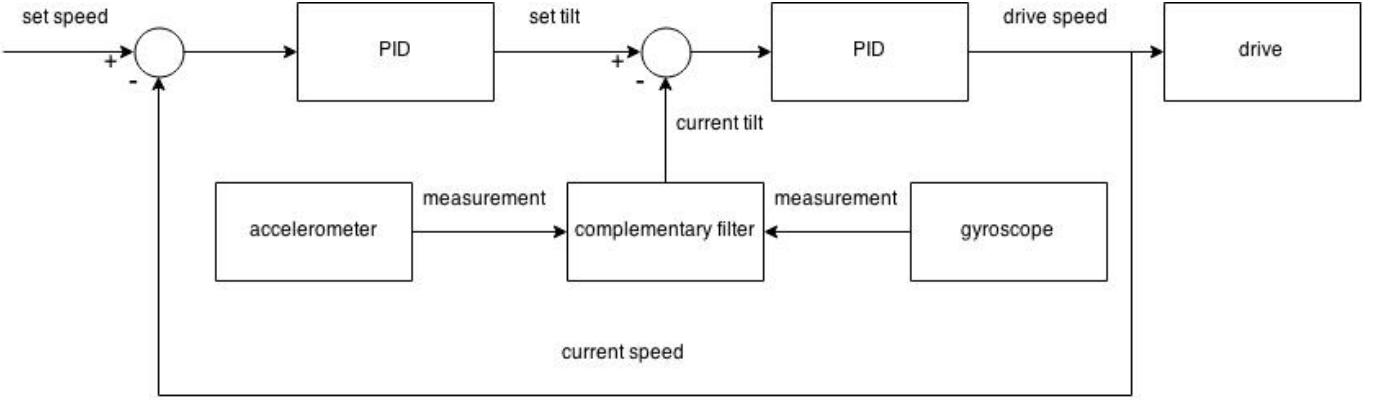


Fig. 3. Control of forward speed of the robot.

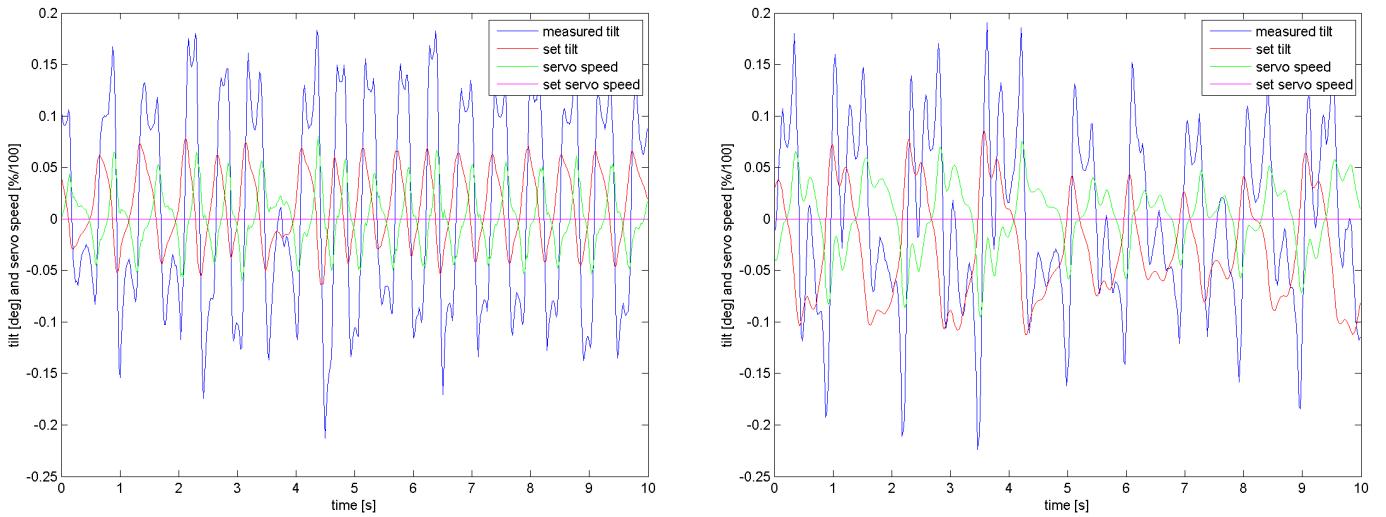


Fig. 4. Standing still; left: cascade of PIDs, right: PI and model

C. Rapid movement forward and backward

In this experiment the target velocity of the robot alternates periodically between its highest possible and its lowest possible value. The results are presented in Fig. 6. For both control methods it is seen that the robot leans forward and backward to a large extend. Although, there is some overshoot in the robot's speed, generally it behaves as expected. Both behaviors are very similar to each other.

D. Overriding an obstacle

In this experiment the robot overrides an obstacle that is 15 cm wide and 0.7 cm high (as compared to its wheel radius of 2.5 cm). The results are presented in Fig. 7. For both control methods the robot needs two attempts to climb the obstacle. When the robot hits the obstacle for the first time, its wheels get blocked, and instead of climbing the obstacle, they are tilting the robot backwards. In response the robot moves back

to regain balance, leans forward, and approaches the obstacle again. This time it is tilted forward more due to a larger value of the integral part of the first controller in cascade (velocity has been smaller than desired). Consequently, the second attempt to get on the obstacle is successful in both cases of control systems.

VII. CONCLUSIONS

In this paper a mobile robot is presented along with its two control systems. The robot is modular and simple to reproduce. The first control system is based on a cascade of two modified PID controllers. The second control system is based on a cascade of a PI controller and a mathematical model of robot dynamics. The experiments demonstrate robustness and versatility of both control methods. Each control system is efficient and immune to limited disturbances. The control system based on a PI controller and a mathematical model is

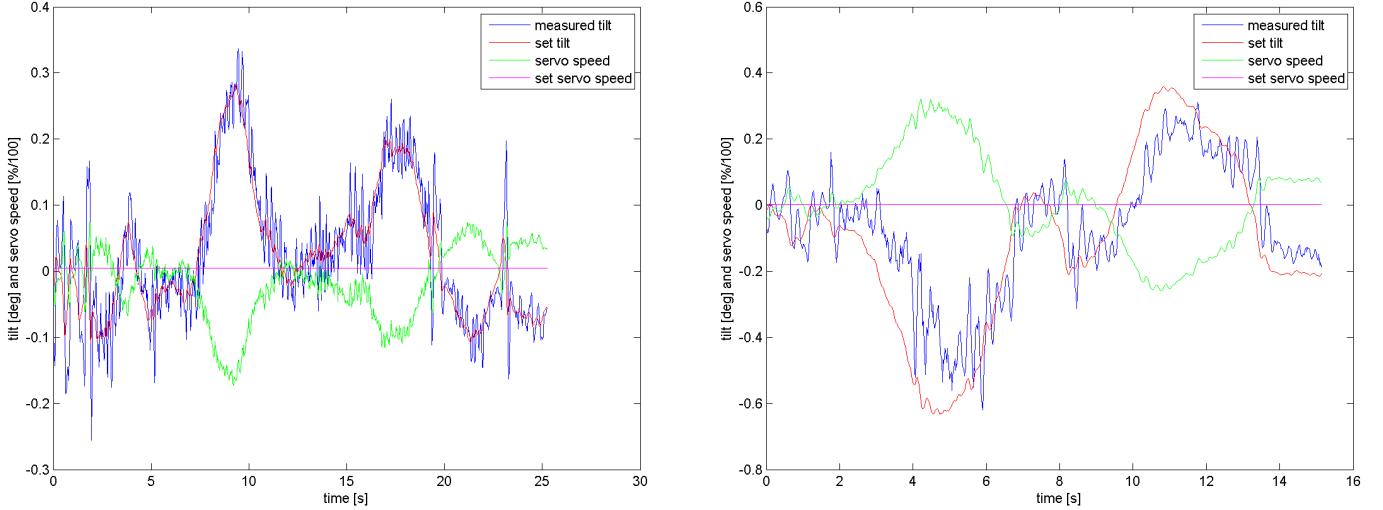


Fig. 5. Rotating about vertical axis; *left*: cascade of PIDs, *right*: PI and model

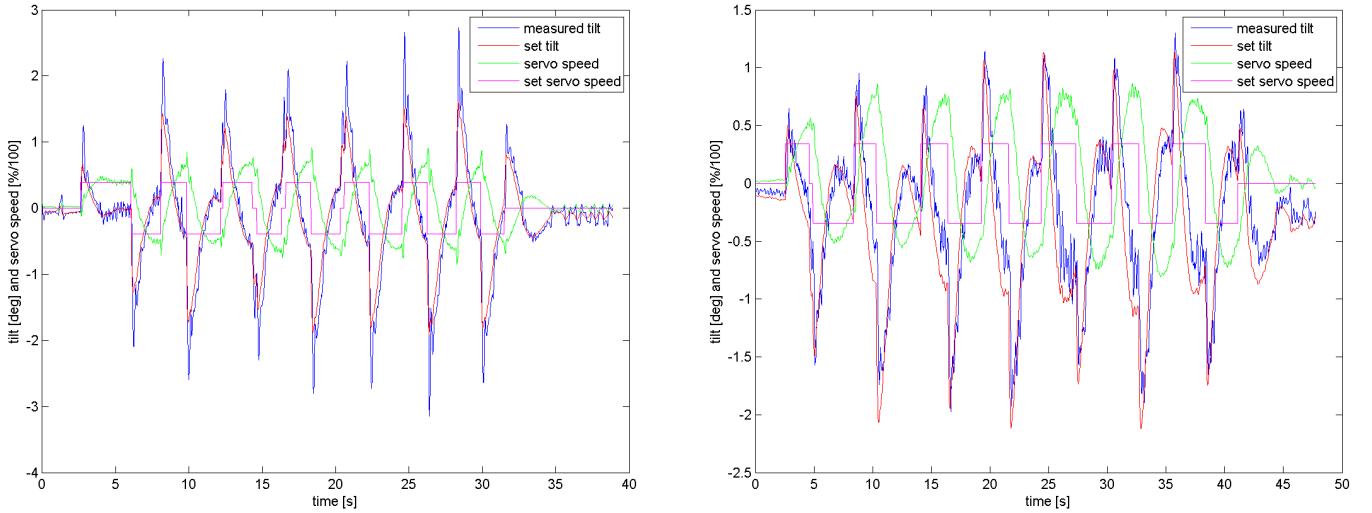


Fig. 6. Rapid movement forward and backward; *left*: cascade of PIDs, *right*: PI and model

also more responsive to user commands and a lot easier to tune, as it encompasses fewer parameters.

REFERENCES

- [1] M. Reynolds, "Balancing a two-wheeled segway robot," 2011, bachelor's thesis, Massachusetts Institute of Technology.
- [2] N. G. M. Thao, D. H. Nghia, and N. H. Phuc, "A pid backstepping controller for two-wheeled self-balancing robot," in *International Forum on Strategic Technology*, 2010, pp. 76–81.
- [3] C. Sun, T. Lu, and K. Yuan, "Balance control of two-wheeled self-balancing robot based on linear quadratic regulator and neural network," in *International Conference on Intelligent Control and Information Processing*, 2013, pp. 862–867.
- [4] L. Sun and J. Gan, "Researching of two-wheeled self-balancing robot base on lqr combined with pid," in *International Workshop on Intelligent Systems and Applications*, 2010, pp. 1–5.
- [5] Q. Yong, L. Yanlong, Z. Xizhe, and L. Ji, "Balance control of two-wheeled self-balancing mobile robot based on ts fuzzy model," in *International Forum on Strategic Research*, 2011, pp. 406–409.
- [6] H. Bin, L. W. Zhen, and L. H. Feng, "The kinematics model of a two-wheeled self-balancing autonomous mobile robot and its simulation," in *International conference on Computer Engineering and Applications*, 2010, pp. 64–68.
- [7] M. Mahler and J. Haase, "Mathematical model and control strategy of a two-wheeled self-balancing robot," in *Annual Conference of the IEEE Industrial Electronics Society*, 2013, pp. 4198–4203.
- [8] V. Kongratana, S. Gulphanich, V. Tipsuwanporn, and P. Huantham, "Servo state feedback control of the self balancing robot using matlab," in *International Conference on Control, Automation and Systems*, 2012, pp. 414–417.
- [9] S. Kim, J. Seo, and S. Kwon, "Development of a two-wheeled mobile tilting & balancing (mtb) robot," in *International Conference on Control, Automation and Systems*, 2011, pp. 1–6.

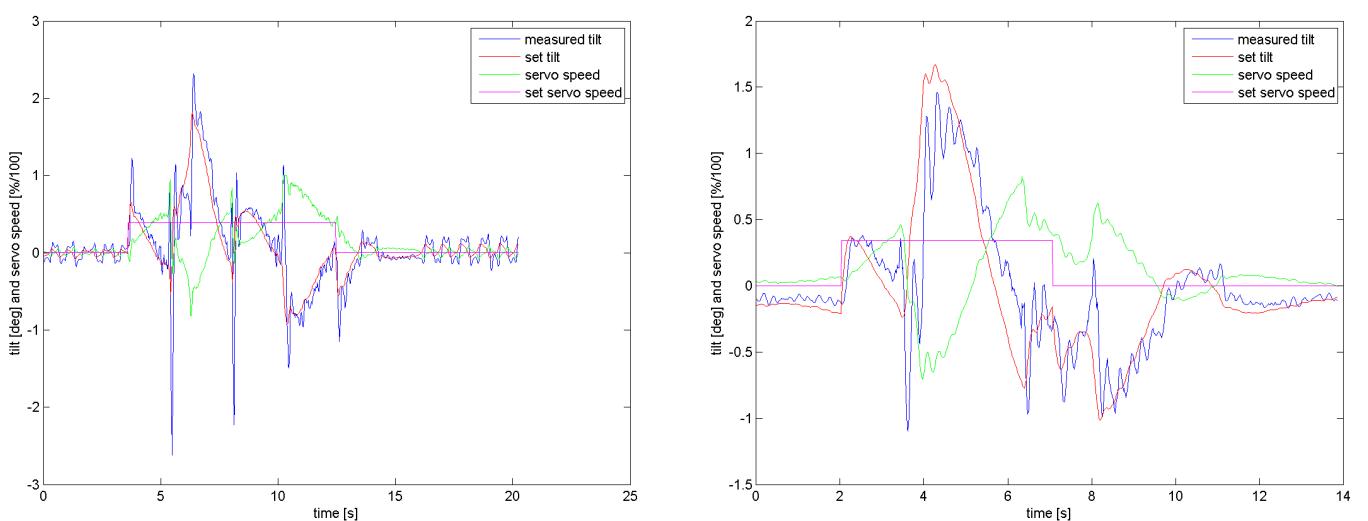


Fig. 7. Overriding an obstacle; *left*: cascade of PIDs, *right*: PI and model