



## The mono-wheel robot with dynamic stabilisation

Patryk Cieslak, Tomasz Buratowski\*, Tadeusz Uhl, Mariusz Giergel

*Department of Robotics and Mechatronics, AGH University of Science and Technology, 30 Mickiewicz Alley, 30-059 Krakow, Poland*

### ARTICLE INFO

#### Article history:

Received 26 January 2011

Received in revised form

22 April 2011

Accepted 2 May 2011

Available online 7 May 2011

#### Keywords:

One-wheel robot

Nonholonomic mono-wheel robot

Stabilization of the mono-wheel robot

Control of the mono-wheel robot

Mono-wheel robot design

### ABSTRACT

This article presents a design process of a mono-wheel robot which consists of building a theoretical model, designing a mechanical structure, simulating the design, building a prototype and testing it. It describes the control strategy for this vehicle, developed during the simulation process, and how it works for a ready built prototype. It mainly focuses on the self-stabilisation problem encountered in a mono-wheel structure and shows the test rig results for this case. The design of the robot is under patent protection.

© 2011 Elsevier B.V. All rights reserved.

### 1. Introduction

The dynamic stability problem in robotics is encountered in many types of mechatronic systems. The complexity of achieving good dynamic stability highly depends on the structure of a system, the way it interacts with the environment and the environment itself. The mechatronic system, presented in this paper, is a mono-wheel robot based on the idea of a mono-wheel motorbike where a passenger sits inside a big wheel. All parts of the robot are placed inside the wheel – motor, sensors, radio, power source and stabilising/control system. This robot has to maintain its upright state when it is not moving and when it starts to move forward or backward. When travelling with some speed it has to use the balancing mechanism to change direction. Our design has been registered in the Polish patent office and got a reference no. 390857 [1].

There already exist some other concepts of mono-wheel self-stabilising robots but most of them consider a different kind of structure and their way of stabilisation produces some restrictions which we wanted to avoid. The most similar concept is described in a US Patent 7337862 [2]. It is a vehicle based on the same idea as our design but it does not cover any active stabilisation system. There is also no information whether it has ever been realised. A second similar construction with a few built prototypes is called a Gyrover [3]. It uses a mechanical gyroscope fitted inside the wheel to provide stable position when the vehicle is stopped or moving

slowly and a tilt mechanism changing the axis of the gyroscope so that the robot can turn using the precession effect which is similar to a mechanism described in a paper by Bauer [4]. Other concepts present a different point of view but all involve designing a kind of active stabilisation and dynamic control system. One interesting idea with a prototype construction is described in a paper by Fujimoto and Uchida [5]. The design consists of a small wheel at the bottom and a two-part body with a hinge joint in the middle. The robot stabilisation is provided differently in forward-backward and left-right directions. In the longitudinal plane this mechatronic system is considered an inverted pendulum on a cart and in the lateral plane it is modelled as a double inverted pendulum with an under-actuated first joint. Stabilisation of the robot is achieved by using two separate controllers. All of the presented data is taken from simulation and there is no information whether the controllers actually work for the built prototype as it was considered a future work by the authors. Another design involving a dynamic stabilisation mechanism is described in a paper by Kappeler [6]. It consists of a small wheel and a tall, one piece body at the top of which a mechanical gyroscope is installed. The gyroscope is mounted perpendicularly to the wheel, in the vertical plane. The robot stabilises itself in the lateral plane by changing the angular speed of the gyroscope (value and direction). In the longitudinal plane this robot is treated the same way as the previous construction. The prototype was tested and it works in real world. The most advanced design in means of production readiness was created by Honda and called U3-X [7]. It is a personal transport vehicle which stabilises similarly to the Segway or any other inverted pendulum type robot [8–11]. It has a small wheel at the bottom and a seat above. What makes it possible to maintain its upright position is a special construction

\* Corresponding author.

E-mail address: [tburatow@agh.edu.pl](mailto:tburatow@agh.edu.pl) (T. Buratowski).

of the wheel. It is a kind of omnidirectional wheel the rim of which has a modular structure consisting of small rings rotating perpendicularly to the whole wheel. It enables the wheel to move forward, backward and to the sides. This ability is used to change the fulcrum of the robot structure in order to keep it in balance. The robot movement is controlled by the position of the passenger's body – if he/she leans in any direction the robot moves that way to keep itself in an upright position.

All of the presented solutions have their pros and cons. The most tested is the latter design, based on the Segway concepts, a vehicle that is commercially available. It is a good solution for a personal transport device but it has some serious drawbacks. The first and most important is that it is not designed to move autonomously or to be remotely controlled, it needs a person sitting at the top of the device to force movement in a desired direction. Even as a manned vehicle it is rather slow but of good maneuverability.

The robot consisting of a small wheel and a two-part body [5] shares some cons with Honda's design. Its structure suggests it cannot be very fast and will be rather hard to control between obstacles because of its large dimensions and tilt angles. The tall body is also unlikely to be able to rise from a fall. All possible onboard electronic devices like sensors, cameras and batteries could only be mounted inside it which would make them exposed to damage. The Unicycle Robot also presents similar cons and although it looks like a simpler solution it is likely to be the least energy efficient because of the continuous rotation of the gyroscope with sudden direction changes causing its motor to consume a lot of energy.

Two vehicle concepts [2,3] and our own design have a lot in common. They both can easily achieve high speeds, their structure is compact, all their electronic components fit inside the wheel which protects them in case of crashing into obstacles and they should be able to rise from a fall in a way similar to the one developed by Morimoto and Doya [12].

The purpose of the project presented in this paper is to create a robot with all the good features of the mentioned designs, mostly the last ones – a thin compact body, an ability to achieve high speeds, a simple construction, a good adaptability to different surfaces and environmental conditions. However our design has to be completely autonomous in means of achieving self-stability and following a designated path or even finding its own path if equipped with additional sensors and a trajectory planning algorithm. It has to be able to maintain an upright position for an unlimited time when stopped or moving. There is no design we could find in the literature fulfilling all of the requirements and being tested in the real world. Although a mono-wheel motorbike, which is our main inspiration, is an old concept of a vehicle, it is very rare due to the control difficulties it presents to the potential users and we wanted to change it also as our design could be used as a vehicle steered similarly to a car.

## 2. Mathematical model of the robot

Mobile robots utilise different methods in order to move i.e. wheels, caterpillars, legs, however the use of the solutions based on wheels is the most energy efficient for such mechatronic systems. A very interesting issue is the control of the nonholonomic mobile robotic systems [13–15]. The naming is connected with the kind of constraints applied to the system. The issue connected with the nonholonomic systems is that the motion velocity constraints are not removable in contrast to the holonomic ones. This type of robot has less drives than the number of degrees of freedom.

If we take the idea of a mono-wheel robot into consideration, it is possible to describe its position using the following coordinates:

$$p = [x \ y \ \varphi]^T \quad (1)$$

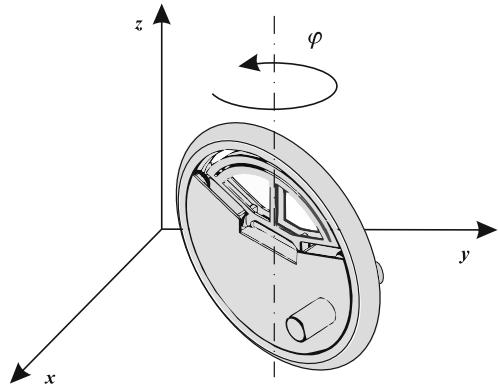


Fig. 1. The position description for the mono-wheel robot.

where first two coordinates, presented in Fig. 1, describe the location of the robot in the plane, while the third describes the orientation of the wheel [16,17]. The required assumption in our system is that there is no skid between the robot wheel and the ground. In such a case the natural limitation of the robot motion can be described as:

$$\frac{dy}{dx} = \tan \varphi. \quad (2)$$

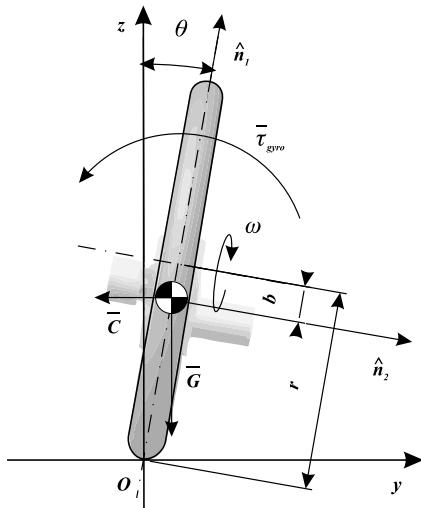
The conclusion from the position description of the mono-wheel robot is that the motion constraints are nonholonomic. It means that the system will be much more difficult to control than a conventional mobile structure. The Eq. (2) shows that the vector  $\dot{p} = [\dot{x}, \dot{y}, \dot{\varphi}]^T$  must always be perpendicular to the vector  $[\sin \varphi, -\cos \varphi, 0]^T$  during motion. If we assume the base in the null space, constructed from the two vectors  $[\cos \varphi, \sin \varphi, 0]^T$  and  $[0, 0, 1]^T$ , where each one is perpendicular to the vector  $[\sin \varphi, -\cos \varphi, 0]^T$ , then we receive a description of the mono-wheel robot motion as follows [18,16,14]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos \varphi \\ \sin \varphi \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_2. \quad (3)$$

The mono-wheel robot is controlled according to the presented equation by influence on its wheel velocity  $u_1$  and the orientation changing speed  $u_2$ . The characteristic feature of the nonholonomic constraints is controllability of the system despite the number of controls being significantly less than the state space dimension. As is shown for our system there are two controls  $u_1, u_2$  and a three-dimensional state space; however the mono-wheel robot is able to reach any location on the surface with any orientation.

The robot kinematics model presented above does not tell us how to change the orientation of the robot during forward motion. To deduce how to make the robot turn, a dynamics model described below was created.

A dynamics model for the robot during forward motion with velocity  $v$  (wheel angular velocity  $\omega$ ) is presented in Fig. 2. When the robot is perfectly upright there is nothing to analyse in the lateral plane as it simply moves straight. It has to lean to start turning which is achieved by changing the whole structure's centre of gravity. When the desired tilt  $\theta$  is achieved the centre of gravity can return to its natural position. The gravity force  $\bar{G}$  causes a circular motion of the robot along a path with a radius of  $R$  as it induces a friction force perpendicular to the wheel forward motion, parallel to the ground. There also arises a centrifugal force  $C$  trying to decrease the wheel inclination. Although we discuss a situation where the robot has a constant inclination it is worth mentioning that there also exists a gyroscopic torque  $\tau_{gyro}$  counteracting changes in the wheel axis orientation which greatly



**Fig. 2.** The model of the system in the lateral plane during forward motion.

helps to stabilise the robot but also limits the turning speed. In this case  $\tau_{gyro} = I_{wheel}\omega\dot{\theta} = 0$ .

The easiest way to describe the dynamics in the presented case is to compare the rate of change of angular momentum of the whole robot in the frame of ground with the torque acting on its centre of gravity [19]:

$$(\dot{L}_{cog})_{ground} = \tau_{cog}. \quad (4)$$

The rate of change of angular momentum in the inertial frame of the ground is:

$$(\dot{L}_{cog})_{ground} = (\dot{L}_{cog})_{body} + \dot{\varphi} \times (L_{cog})_{ground}. \quad (5)$$

In the non-inertial frame  $(\dot{L}_{cog})_{body} = 0$ . If we express the orbital angular velocity as  $\dot{\varphi} = -(v/R)(\cos\theta\hat{n}_1 - \sin\theta\hat{n}_2)$  the angular momentum of the robot equals:

$$(L_{cog})_{ground} = -\frac{v}{R}(\cos\theta I_{robot1}\hat{n}_1 - \sin\theta I_{robot2}\hat{n}_2) - \frac{v}{r}I_{wheel}\hat{n}_2 \quad (6)$$

where  $I_{robot1}$  and  $I_{robot2}$  are the principal moments of inertia of the whole robot around the  $\hat{n}_1$  and  $\hat{n}_2$  axes while the  $I_{wheel}$  is the moment of inertia of the outer wheel around its axis of rotation and  $r$  is the wheel radius. Substituting Eq. (6) in Eq. (5) gives:

$$\begin{aligned} (\dot{L}_{cog})_{ground} &= \frac{v^2}{R^2} \left[ (I_{robot1} - I_{robot2}) \sin\theta \cos\theta \right. \\ &\quad \left. + I_{wheel} \cos\theta \frac{R}{r} \right] \hat{n}_3. \end{aligned} \quad (7)$$

As mentioned earlier the friction component of the reaction force of the ground is a centripetal force and it equals  $F = Mv^2/R$  where  $M$  is the total mass. The normal reaction is equal to the gravity force  $N = Mg$ . The torque around the  $\hat{n}_3$  axis can be expressed as:

$$\begin{aligned} \tau_{cog} &= [N(r - b) \sin\theta - F(r - b) \cos\theta] \hat{n}_3 \\ &= \left[ Mg(r - b) \sin\theta - \frac{Mv^2}{R}(r - b) \cos\theta \right] \hat{n}_3. \end{aligned} \quad (8)$$

Finally substituting Eqs. (7) and (8) in Eq. (4) gives:

$$\begin{aligned} Mg(r - b) \tan\theta - \frac{Mv^2}{R}(r - b) \\ = \frac{v^2}{R^2} \left[ (I_{robot1} - I_{robot2}) \sin\theta + I_{wheel} \frac{R}{r} \right] \\ = (\text{orbital term}) + (\text{spin term}). \end{aligned} \quad (9)$$

This equation shows the importance of the rate of change of the wheel spin and the orbital angular momentum – the spin term dominates the orbital term because  $R/r \gg \sin\theta$  since it is assumed that  $\theta < 30^\circ$  and  $R \gg r$ . If the relatively small orbital term is ignored and an assumption that  $I_{wheel} = mr^2$  and  $(r - b) = c \cdot r$  is made, Eq. (9) gives:

$$\tan\theta = \frac{v^2}{gR} \left( 1 + \frac{m}{Mc} \right). \quad (10)$$

The dynamics model presented above can be combined with the kinematics model presented earlier in this chapter to better reflect the orientation changes of the real robot:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi r \\ \sin\varphi r \\ 0 \end{bmatrix} \omega + \begin{bmatrix} 0 \\ 0 \\ g/\left[r\left(1 + \frac{m}{Mc}\right)\right] \end{bmatrix} \frac{\tan\theta}{\omega}. \quad (11)$$

The resulting equation shows that the robot can be controlled by changing the wheel velocity and tilting the whole structure. From the dynamics model it can also be deduced that when the robot is not moving (or moving very slowly) there is no centrifugal force and no gyroscopic torque that could prevent it from falling on one of the sides. That is why there has to be an active stabilisation mechanism as it was assumed that the robot has to be able to maintain an upright position for an unlimited time regardless of its velocity. To solve all the problems – forward motion, turning, maintaining an upright position – we decided to use an electric motor with a gear drive to propel the wheel and designed a kind of a balancing mechanism consisting of a one-side lever with ballast at the end, powered by a servomechanism, to be able to change the centre of gravity of the whole device.

To understand the dynamics of the balancing mechanism a dynamics model of the mono-wheel robot in the lateral plane, when no forward motion occurs, is presented below. It is equivalent to a model of a double inverted pendulum whose first joint is underactuated as shown in Fig. 3. The motion equation of the model for the assumptions:  $x = [\theta, \alpha, \dot{\theta}, \dot{\alpha}]^T$ ,  $y = [\theta, \alpha]^T$ ,  $\tau = [0, \tau_2]^T$  in the lateral plane, can be written as follows [5]:

$$\dot{x} = \begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ M(\gamma)^{-1}(\tau - h(\gamma, \dot{\gamma}) - G(\gamma)) \end{bmatrix} \quad (12)$$

where:

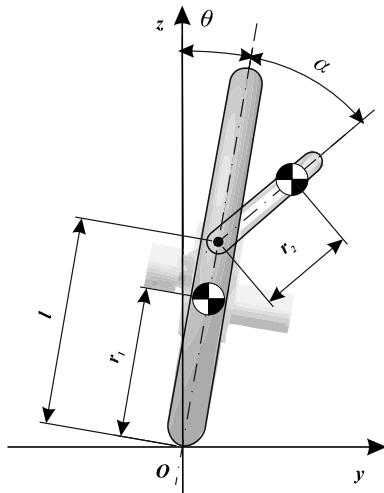
$$M(\gamma)$$

$$= \begin{bmatrix} I_1 + I_2 + m_2(l^2 + 2lr_2 \cos\alpha) & I_2 + m_2lr_2 \cos\alpha \\ I_2 + m_2lr_2 \cos\alpha & I_2 \end{bmatrix} \quad (13)$$

$$h(\gamma, \dot{\gamma}) = \begin{bmatrix} -m_2lr_2 \sin\alpha(2\dot{\theta}\dot{\alpha} + \dot{\alpha}^2) + D_1\dot{\theta} \\ m_2lr_2\dot{\theta}^2 \sin\alpha + D_2\dot{\alpha} \end{bmatrix} \quad (14)$$

$$\begin{aligned} G(\gamma) &= -g \left[ \begin{array}{c} (m_1r_1 + m_2l) \sin\theta + m_2r_2 \sin(\theta + \alpha) \\ m_2r_2 \sin(\theta + \alpha) \end{array} \right] \\ &\quad + g_2 \left[ \begin{array}{c} (m_1r_1 + m_2l) \cos\theta + m_2r_2 \cos(\theta + \alpha) \\ m_2r_2 \cos(\theta + \alpha) \end{array} \right]. \end{aligned} \quad (15)$$

The construction data necessary to calculate the motion Eq. (12) consist of:  $l$  – distance from the ground to the rotational joint of the lever,  $m_1, m_2$  – mass of the first and second elements,  $\theta$  – angle of wheel inclination,  $\alpha$  – angle of lever rotation,  $\tau_2$  – input torque to the rotational joint,  $r_1, r_2$  – position of the centre of mass of the wheel and the lever,  $I_1, I_2$  – inertia of the wheel and the lever,  $D_1, D_2$  – damping factor of the wheel inclination and the lever rotation,  $g$  – gravity acceleration,  $g_2$  – centrifugal acceleration.



**Fig. 3.** The model of the system in the lateral plane during halt.

After linearisation of the nonlinear motion Eq. (12) around the equilibrium point, where  $\theta = \arctan(g_2/g)$ ,  $\dot{\theta} = \alpha = \dot{\varphi} = 0$ , the control law for the balancing mechanism can be presented as follows [5]:

$$\dot{x} = Ax + B\tau_2 \quad (16a)$$

$$y = Cx. \quad (16b)$$

The control law connected with the balancing mechanism is significant to the motion of the mono-wheel robot because it is responsible for the self-stabilisation during a halt and for orientation changing in forward motion. However the above model was provided mainly to present the kind of dynamics we deal with and to prove that the designed balancing mechanism can be controlled with only one drive because we decided to use the multi-body approach where it is not possible to access any analytical results. In this paper we have focused on the self-stabilisation problem when there is no forward motion. It is not mentioned how the two separate parts – the wheel and the cart with the balancing mechanism – work together but it is assumed that the balancing mechanism is always kept in the upper part of the wheel. The last assumption is always true when there is no forward motion as the centre of gravity of the inner part of robot is below the centre of the outer wheel.

### 3. Mechanical structure

The mechanical structure of the robot described in this paper consists of several major components, presented in Fig. 4, these are: a big wheel, a cart moving freely inside the wheel and a balancing mechanism. The wheel is equipped with a solid rubber tire, an internal gear and two guides for rolls. The cart is the body of the vehicle containing all of the electronics, a battery, a digital

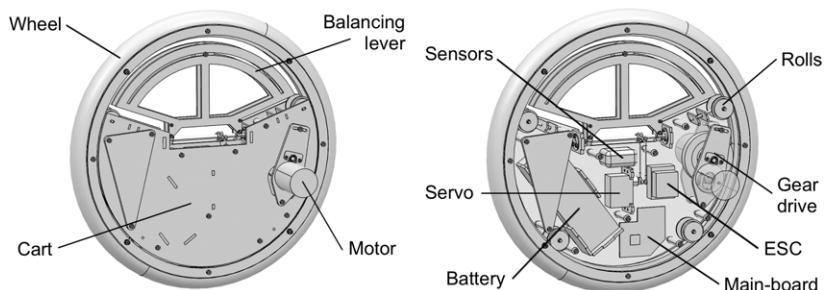
servomechanism and a gear drive, powered by a brushless DC motor, equipped with an optical encoder. This body is attached to the wheel by four pairs of rolls with bearings which ensure independent motion of these two parts. The balancing mechanism consists of an arc shaped lever with a slot for mounting ballast and a shaft attached to the robot body by bearings, equipped with a small two side lever connected with the servomechanism arms by rods with ball joints. The lever rotation, powered by the servomechanism, enables us to move the ballast to the left or right side of the wheel plane. This action produces moment and changes the centre of gravity of the robot which is used to keep the vehicle in upright position and turn. The two-step gear drive placed inside the cart consists of three spur gears and the wheel internal gear. The first is placed on the motor shaft and it meshes with the second one placed on the main shaft of the gear drive. The main shaft is attached to the cart by bearings sitting inside special gear drive casing plates. These plates are attached to the cart sides in a way which enables their rotation about one of the mounting screws. There is also an optical encoder on the shaft mounted to one of the plates from the outside. The third gear is also placed on this shaft and it meshes with the internal gear of the wheel. The purpose of the special construction of casing plates and their mounting is to enable us to change the gear drive ratio by exchanging gears and setting up the correct shaft axis distances.

The prototype size is about 350 mm in diameter due to the use of radio controlled model electronics. It is made of aluminium and steel. The second-step gears are cut of PTFE and the other two come from a model car, so that it is easy to replace them with new ones. The ready made construction of the robot is presented in Fig. 5, it is tethered to a personal computer for control and monitoring purposes.

### 4. Electronic structure

The electronic structure consists mainly of elements taken from radio controlled models, these are: the digital, high torque, high speed servomechanism, the brushless DC motor with its electronic speed controller (sensored design), the Lithium-Polymer battery with a high discharge rate and the Spektrum receiver (2, 4 GHz).

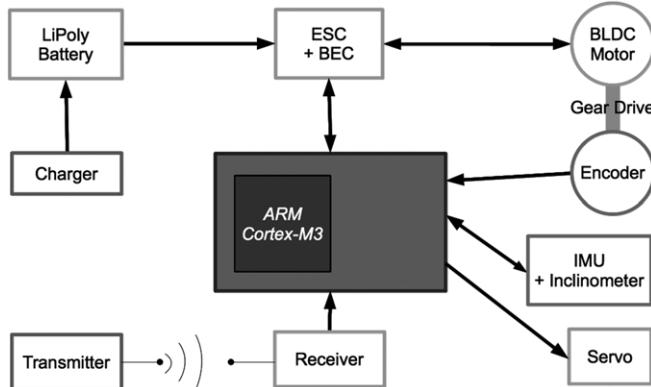
Unlike in the radio controlled model, these elements are not connected directly to each other but to a custom made main-board equipped with an ARM CortexM3 micro-controller. This electronic circuit is the brain of the robot and it contains the control algorithm. There are sensors, connected to the main-board, which provide feedback to the implemented controller. These are a 3-axis MEMS accelerometer and a 2-axis MEMS gyroscope mounted on a single board, forming an IMU, a 2-axis inclinometer and an optical quadrature encoder. Thanks to a wide range of peripherals integrated into the ARM CortexM3 micro-controller it can be used alone to read analogue signals from the IMU sensors with a 12-bit ADC, communicate with the inclinometer through an USART interface and capture the encoder signals. With the addition of a few simple electronic parts it is also able to



**Fig. 4.** The mechanical construction of the one-wheel robot.



**Fig. 5.** The robot prototype during the self-stabilisation phase.



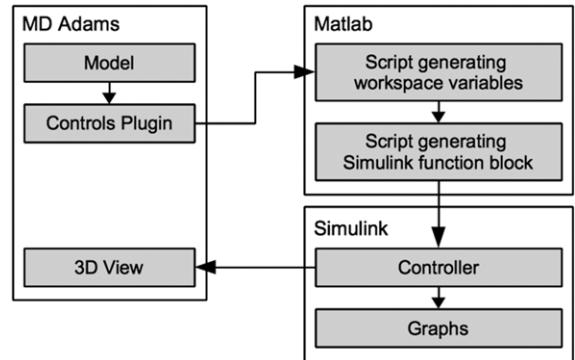
**Fig. 6.** The scheme of the electro-mechanical system.

read and generate PWM signals for the servomechanism, the electronic speed controller and the receiver. The scheme of the electro-mechanical system responsible for the control and self-stabilisation of the robot is shown in Fig. 6. A very important ability of the presented structure is to allow wireless control by a personal computer or a human operator with a handheld transmitter [20,21]. Everything is powered by the robot battery.

## 5. Simulation

A simulation of the design was done to check whether it is correct, prior to building a prototype. The whole simulation process could be performed only in Matlab [22,23] using the described mathematical model but because of the simplicity of its equations it would be hard to include all of the geometry data of the designed structure in it. To simplify the part design and the correlation between the model and the real structure a more sophisticated simulation environment was chosen. The simulation process involved using two connected computational environments – MD Adams and Matlab/Simulink. The idea of the simulation workflow is shown in Fig. 7.

The first stage was creating a mechanical model in Adams View and determining the state variables which we wanted to control and measure (inputs and outputs of the model). As one can see, in the described mechatronic structure, there are two ways to generate forces which can change the state of the robot. The first



**Fig. 7.** The scheme of the simulation procedure.

is using the balancing lever rotated by the servomechanism and the second is using the motor to move the cart inside the wheel. That is why our inputs to the model are the servomechanism torque and the motor velocity. The outputs of the model are its tilt in the longitudinal and lateral planes, the wheel orientation in reference to the ground plane, the angle and velocity of the servomechanism and the torque acting on the motor shaft. Adams View software is able to simulate the model by itself but we wanted to design a custom controller using Matlab/Simulink so a plugin called Adams Controls was used to generate a Simulink function block and a compiled model library. The next stage was building a controller in Simulink which connected to the inputs and outputs of the generated model function block. This way it was possible to build a closed feedback loop where the controller sets the servomechanism position and the motor rotating speed based on the orientation of the vehicle model. The result of the simulation is displayed as a 3D model view and a set of Simulink graphs. Below, in Fig. 8, there is a presentation of the Simulink model.

Although a full mechanical model of the vehicle was built we mainly focused on the problem of maintaining its upright position when not moving because it is the hardest case in our scope. From other designs presented in the state of the art we can see that keeping upright position while the vehicle is moving and controlling it should be rather simple. What is more, the robot has to be stable when stopped because otherwise it will not be able to start moving safely. As the S-function controller is designed to stabilise the device during a halt it does not have a motor velocity input and the Adams model motor velocity input is connected to a constant. The controller inputs are fed with tilt angle and tilt velocity outputs of the Adams model and with constants representing the controller formula factors and the desired tilt angle. There are a few important elements in the Simulink block diagram which simulate parameters of the real electronics used in the design. These are rate transition blocks which change the sampling frequency of the model tilt signals so that the controller works with a speed possible to achieve using the micro-controller we chose, some noise generating blocks to simulate the real output noise of the Kalman filtering algorithm and a full dynamics model of the servomechanism tuned to reflect the characteristics of the drive installed in the real structure.

Different controller types were tested and it turned out they all have problems stabilising the robot. The first that we used was a proportional controller, then we added the integral and derivative parts but it did not work either. It was observed that a controller taking only control deviation into account can provide stabilisation when there is no servomechanism model and the control frequency is much higher than possible to achieve. Even if it provides stabilisation in this theoretical configuration it fails when the robot tilts more to one side as the control values start to rise

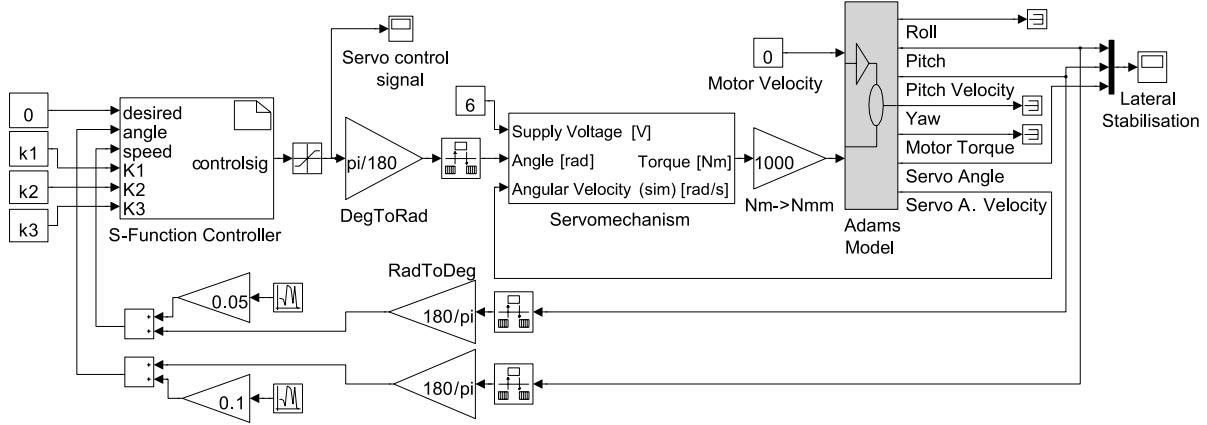


Fig. 8. The coupled Matlab/Simulink and MD Adams simulation of the robot dynamics.

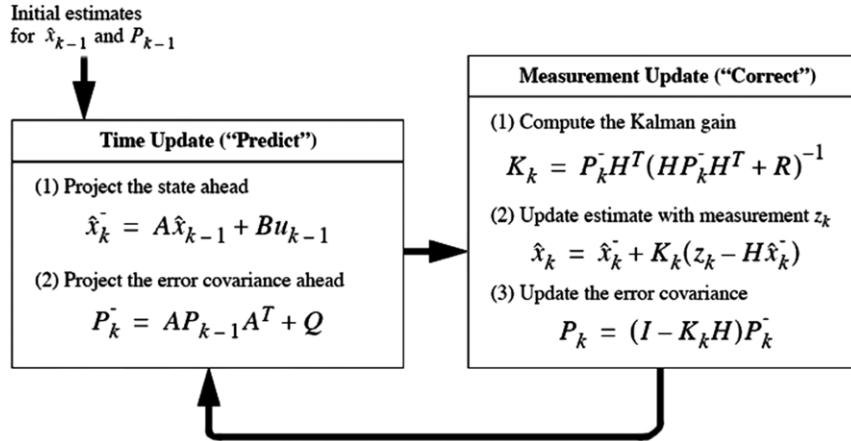


Fig. 9. The Kalman filter algorithm.

quickly. The next attempt was to use a fuzzy logic controller but it was hard to determine the correct rules. Finally we decided to built a custom controller using an S-function block and programming it with a kind of a math formula with some additional logic. After a few different approaches we managed to create a working controller.

The controller formula consists of components representing: a constant step the value of which depends on the tilt angle deviation sign, a fraction that depends on the tilt angle deviation value and a fraction that depends on the tilt angular velocity, multiplied by experimentally received factors:

$$\alpha_k = \alpha_{k-1} + \text{sgn}(\theta_{\text{desired}} - \theta_k) \cdot K_1 \\ + (\theta_{\text{desired}} - \theta_k) \cdot K_2 + \dot{\theta}_k \cdot K_3. \quad (17)$$

What is significant in this formula is that it takes the prior control value into account, not only the control deviation. Using the developed custom controller we were able to achieve an upright position for an unlimited period of time. A full correlation between the simulation and the real structure's behaviour was hard to achieve due to the unwanted clearance that existed mainly in the balancing mechanism but only the custom controller based on observations was able to provide stabilisation in both cases.

## 6. Control system implementation

The control algorithm was implemented on the ARM CortexM3 micro-controller. It consists of a sensor calibration procedure, a closed control loop, with the custom controller developed during the simulation, and a serial communication interface which allows

us to monitor the robot state and send it commands from a personal computer.

The sensor calibration procedure is used to initialise a 3-axis accelerometer, a 2-axis gyroscope and a 2-axis inclinometer. The inclinometer is calibrated on demand and its zero levels are written in its flash memory. It is used as a reference for the accelerometer during the calibration procedure, providing information about how the sensor is mounted in the robot. The accelerometer and gyroscope zero levels are calibrated every time the system is turned on or reset. Only these two sensors are constantly read in the control loop.

For such a complex mechatronic system as our design it is necessary to use the sensor fusion technique, because the noise ratio of each single sensor is too high to give us consistent information about the robot tilt, which is the value we want to control. To calculate the tilt values on both the longitudinal and lateral axes we use two identical Kalman filters [24]. Such filters easily compensate for the gyroscope bias, the accelerometer gravity acceleration measurement error, connected with the robot movement, and the electrical noise. The gyroscope angular velocity readings are integrated over time and used for the prediction part of the filters and the accelerometer readings, providing information about the gravity vector orientation, are used for the correction part. A typical Kalman filter algorithm [15,24] is presented in Fig. 9.

As one can see, this is a two-stage recurrent algorithm. In the first stage we predict new values of the monitored parameters based on the last values and the measurements  $u_{k-1}$  from the first sensor which is a gyroscope in our case. We also predict new values of the covariance matrix.

In the second stage we first compute the Kalman gain matrix which informs us how big the influence is of the measurements from the second sensor on the values of the monitored parameters. In our case the second sensor is an accelerometer. Next we compute corrected values of the monitored parameters based on the measurement innovation ( $z_k - H\hat{x}_k$ ). Finally we compute corrected values of the covariance matrix.

In our implementation of the described algorithm we decided to monitor three parameters: the tilt angle and angular velocity and the gyroscope bias, therefore our state vector is equal:

$$x = [\theta, \dot{\theta}, g_{bias}]^T. \quad (18)$$

The values of the particular matrices are as follows:

$$A = \begin{bmatrix} 1 & 0 & -dt \\ 0 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} dt \\ 1 \\ 0 \end{bmatrix}, \quad (19)$$

$$H = [1 \ 0 \ 0]$$

where  $dt$  is the sampling time.

The variance matrices look like this:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot q, \quad R = [r] \quad (20)$$

where  $q$  is the gyroscope noise and  $r$  is the accelerometer noise.

The covariance matrix is initialised as follows:

$$P = \begin{bmatrix} r & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (21)$$

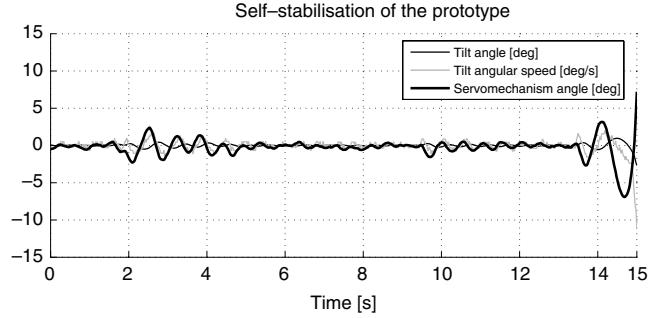
and the state vector  $x$  is filled with zeros.

The Kalman filter output is used as the input to the control function which uses a mathematical formula to generate control signals for the servomechanism. These signals cannot be used directly to set the servomechanism position, because they contain some sudden changes of direction which could break the servomechanism arms or gears. That is why we use a control signal smoothing algorithm constructing a B-spline trajectory from the raw control values. It introduces a slight delay in servomechanism movements but it effectively protects the drive from overstress. The B-spline degree was chosen so that it is the lowest possible value that adjusts the control signal enough to minimise the vibration due to signal spikes and still not cause the control algorithm to fail.

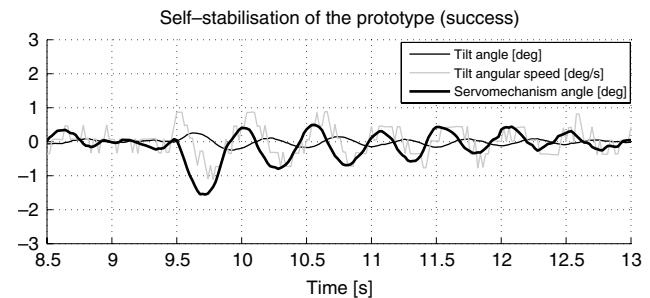
Besides the measurement and control parts of the implementation there is also a communication procedure which is used to read data from the micro-controller and set various parameters of the control algorithm using a personal computer. The data monitoring is not true real-time because of operating system lag but it is a valuable source of information when tweaking the control algorithm parameters and analysing the robot behaviour. The on-board control algorithm is true real-time because it is interrupt driven – not affected by the communication. Problems with task scheduling were not met because the micro-controller used has a much higher computational power than presently required.

## 7. Prototype testing

To efficiently test the prototype we constructed a kind of a table which can be easily leveled and used a personal computer connected to the robot's electronic circuit. Through the computer we can not only program the micro-controller but also change the control function parameters and monitor the state of the vehicle in realtime.



**Fig. 10.** The time courses of the robot tilt angle, tilt angular speed and servomechanism angular position during the conducted test.



**Fig. 11.** A fragment of the time courses connected with a low value force acting on the wheel. Success of the self-stabilisation algorithm.

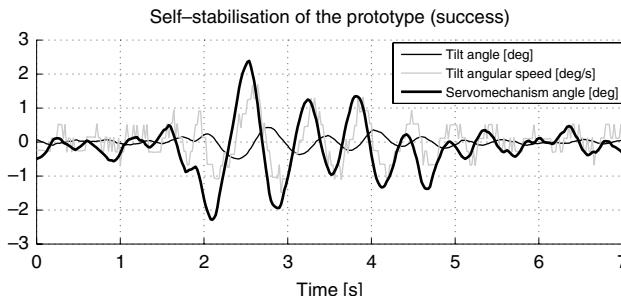
The main-board programming is done using a JTAG interface working with Eclipse IDE. The monitoring of the vehicle state is done using a USB cable connected to the main-board, by a serial interface, emulated by the FTDI driver. The serial communication is used by a control panel designed in LabView. It shows graphs of values measured by the sensors and generated by the Kalman filter and the control function. All of these data can be captured to a file. It also displays calibration information and let us start a recalibration procedure. The control panel can be used to change the control function gain values and to tweak the sensors and servomechanism.

In this paper we focus on the self-stabilisation of the one-wheel robot in the lateral plane when it is stopped or moving with a speed lower than 0.3 [m/s] in the forward/backward direction. In these cases it is the most difficult to maintain upright position because of the lack of or negligible influence of the gyroscopic effect helping to keep the wheel upright when it rotates.

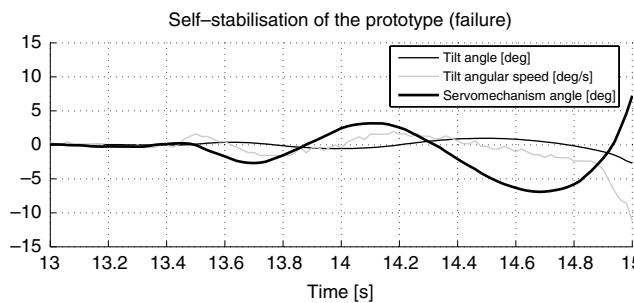
In the following figures in this section the time courses of a simple test of the robot are presented. The testing procedure consisted of setting the robot on a table, turning it on and firing the on-board controller by the control panel described earlier. When the robot stabilised itself it was disturbed by a force acting at the highest point of the wheel and perpendicular to the wheel plane. The forces of different values were used and the time courses of the robot tilt angle, tilt angular speed and servomechanism angular position were captured by the control and monitoring panel.

Fig. 10 presents the time courses of one of the tests where different forces were used and the robot tried to keep its upright position. As one can see the forces disturbed the robot three times during this test. The force value did not exceed the critical force value the first two times and the robot managed to maintain upright position. The third time the force value was too high and the robot fell on its side.

In Fig. 11 there is a fragment of the described test where a force with a low value acted at time 9.5 [s]. It is clearly visible that the robot tilt angular speed rises rapidly in this moment and the balancing lever is rotated by the servomechanism to compensate



**Fig. 12.** A fragment of the time courses connected with a high value force acting on the wheel. Success of the self-stabilisation algorithm.



**Fig. 13.** A fragment of the time courses connected with a force higher than critical acting on the wheel. Failure of the self-stabilisation algorithm.

for it. The robot tilt angle changes in a sinusoidal manner and attenuates to a level of stable work.

In the next Fig. 12 there is another fragment of the test where a force with a higher value acted at time 2 [s]. This time it caused a slightly different behaviour of the robot balancing mechanism suggesting that the force value used was close to the critical value because the oscillations of the structure tend to rise and fall before they finally attenuate in a manner similar to the first case.

The last fragment presented in Fig. 13 shows the behaviour of the robot when a force higher than critical acts at time 13.5 [s]. All of the time courses rise exponentially like in a classic unstable second order system.

The presented results prove that the robot achieves dynamic stability by using its on-board controller. The range of the values of disturbing forces that do not cause a failure of the self-stabilisation algorithm can be further expanded by using larger mass of the balancing lever ballast and eliminating the clearance of the servomechanism and balancing lever connections which is one of our primary tasks in the future.

## 8. Conclusions

In this paper a mathematical description of a mono-wheel robot was presented. Then a mechanical design of a particular robot was described. The electronic structure was also discussed to complete the robot description. The whole simulation process and its results were presented and the implementation of the developed control strategy was described. Finally the prototype testing phase and its results were discussed for the case of a self-stabilisation problem encountered in the design. The dynamic stability of the robot was achieved.

The future works on the described design consist of working out a turning algorithm and increasing its speed and reliability. It is also necessary to increase the robot autonomy by using exteroceptive and proprioceptive sensors and a vision system which would allow us to implement a trajectory planner. This kind of movement control would allow us to exclude the human operator equipped with a radio transmitter controlling the speed and orientation of the robot.

## References

- [1] T. Uhl, T. Buratowski, P. Cieslak, K. Jasinski, Remotely controlled, single wheel vehicle with the engine drive, PL 390857 A1, Biuletyn Urzedu Patentowego nr 20, Poland, 2010.
- [2] P. Greenley, J. Rehkemper, Mono-wheel Vehicle with Tilt Mechanism, United States Patent 7337862, 2008.
- [3] Y. Xu, B. Brown, A Single-Wheel, Gyroscopically Stabilized Robot (GYROVER), <http://www.cs.cmu.edu/afs/cs/project/space/www/gyrover/gyrover.html>.
- [4] R.J. Bauer, Kinematics and dynamics of a double-gimbaled control moment gyroscope, Mechanism and Machine Theory 37 (12) (2002).
- [5] Y. Fujimoto, S. Uchida, Three dimensional posture control of mono-wheel robot with roll rotatable torso in: Proceedings of International Conference on Mechatronics, Kumamoto, Japan, 2007.
- [6] F. Kappeler, Unicycle Robot, EPFL, Automatic control Laboratory, Switzerland, <http://lawww.epfl.ch/webdav/site/la/users/139973/public/reports/Kappeler.Rapport.pdf.pdf>, 2007.
- [7] Honda Company, Honda U3-X: Reinventing the wheel, <http://www.fasterandfaster.net/2009/09/honda-u3-x-reinventing-wheel.html>.
- [8] H. Yun-Su, Y. Shinichi, Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot, Robotics and Autonomous Systems 17 (1–2) (1996).
- [9] F. Grasser, A. D'Arrigo, S. Colombi, JOE: A mobile, inverted pendulum, IEEE Transactions on Industrial electronics 49 (1) (2002).
- [10] P. Deegan, B.J. Thibodeau, R. Grupen, Designing a Self-Stabilizing Robot For Dynamic Mobile Manipulation, University of Massachusetts Amherst, Laboratory for Perceptual Robotics, 2006.
- [11] K. Yeonhoon, K. Soo Hyun, K. Yoon Keun, Dynamic analysis of a nonholonomic two-wheeled inverted pendulum robot, Journal of Intelligent and Robotic Systems 44 (1) (2005).
- [12] J. Morimoto, K. Doya, Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning, Robotics and Autonomous Systems 36 (1) (2001).
- [13] J.A. Batlle, A. Barjau, Holonomy in mobile robots, Robotics and Autonomous Systems 57 (4) (2009).
- [14] H. Guang-Ping, G. Zhi-Yong, The nonholonomic redundancy of second-order nonholonomic mechanical systems, Robotics and Autonomous Systems 56 (7) (2008).
- [15] K. Iagnemma, S. Dubowsky, Mobile Robots in Rough Terrain, in: Springer Tracts in Advanced Robotics, vol. 12, Springer-Verlag, Berlin, Heidelberg, 2004.
- [16] I. Duleba, Metody i Algorytmy Planowania Ruchu Robotow Mobilnych i Manipulacyjnych, Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2001.
- [17] K. Tchon, A. Mazur, I. Duleba, R. Hossa, R. Muszynski, Manipulatory i Roboty Mobilne Modele, Planowanie Ruchu, Sterowanie, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 2000.
- [18] L. Biagiotti, C. Melchiorri, Trajectory Planning for Automatic Machines and Robots, Springer-Verlag, Berlin, Heidelberg, 2008.
- [19] E.C. Horsfield, Pitfalls of prosaic precession—due both to spin and rotational motion, European Journal of Physics 19 (1998).
- [20] T. Brauni, Embedded Robotics—Mobile Robot Design and Application with Embedded Systems, Springer-Verlag, Berlin, Heidelberg, 2008.
- [21] F. Fahimi, Autonomous Robots—Modeling, Path Planning and Control, Springer, 2009.
- [22] J. Garcia de Jahn, A. Callejo, A straight methodology to include multibody dynamics in graduate and undergraduate subjects, Mechanism and Machine Theory 46 (2) (2011).
- [23] C. Vibet, Symbolic modeling of robot kinematics and dynamics, Robotics and Autonomous Systems 14 (4) (1995).
- [24] S.G. Mohinder, P.A. Angus, Kalman Filtering: Theory and Practise Using MATLAB, Wiley, 2008.



**Patryk Cieslak** received a M.Sc. degree in Automatics and Robotics from the Faculty of Mechanical Engineering and Robotics, AGH University of Science and Technology in 2010. He is a Ph.D. student in the Department of Robotics and Mechatronics. His main research interest is mobile robotics and autonomous systems.



**Tomasz Buratowski** received a M.Sc. degree in Robotics and Automatics from the Faculty of Mechanical Engineering and Robotics, AGH University of Science and Technology in 1999, the Ph.D. degree in 2003 also at AGH University. He is currently employed at AGH University as an assistant professor. His main research area is connected with industrial and mobile robots and also human–robot interaction, fuzzy logic applications, modelling and identification of mechatronic systems.



**Tadeusz Uhl** received a M.Sc. degree in 1979 at AGH University of Science and Technology, the Ph.D. in 1983. He is currently employed at AGH University as a professor and Head of Department of Robotics and Mechatronics. His main area of research is construction dynamics, especially modal analysis. His field of scientific interest concerns the active reduction of vibration, control systems and mechatronic systems.



**Mariusz Giergiel** received a M.Sc. degree in Electronics from the Faculty of Electrotechnics Automatics and Electronics, AGH University of Science and Technology in 1985, and a Ph.D. degree in 1992 also at AGH University. He is currently employed at AGH University as a professor. His area of research is based on the application of mobile robots for industrial solutions, and also vibration isolation phenomena is in his fields of interest.