**JOURNAL OF THEORETICAL
AND APPLIED MECHANICS**
50, 1, pp. 99-118, Warsaw 2012
50th anniversary of JTAM

A SELF-STABILISING MULTIPURPOSE SINGLE-WHEEL ROBOT

Tomasz Buratowski, Patryk Cieślak,
Mariusz Giergiel, Tadeusz Uhl

AGH University of Science and Technology, Department of Robotics and Mechatronics, Kraków, Poland; e-mail: tburatow@agh.edu.pl; pcieslak@agh.edu.pl; giergiel@agh.edu.pl; tuhl@agh.edu.pl

This article presents a design process of a single-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 the single-wheel structure and shows the test rig results for this case. The design of the robot is under patent protection.

Key words: single-wheel robot, nonholonomic single-wheel robot, stabilization, control design

1. Introduction

The dynamic stability problem in robotics is encountered in many types of mechatronic systems. The complexity of achieving a solid 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 single-wheel robot based on the idea of a single-wheel motorbike where the 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 position 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 the heading angle. Our design has been registered in the Polish patent office and got a reference no. 390857. There already exist some other concepts of single-wheel self-stabilising robots but most of them consider a different kind of structure, and their way of stabilisation produces some restrictions which we want to avoid. The most similar concept is described in a US Patent 7337862 titled "Mono-wheel vehicle with tilt mechanism" (Greenley and Rehkemper, 2008). 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 if it has ever been realised. Another similar construction with a few built prototypes is called Gyrover ("A Single-Wheel, Gyroscopically stabilised Robot") ([11], Xu and Brown, 1997; Xu and Ou, 2000). It uses a mechanical gyroscope fit 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. This way of maintaining upright position and changing the heading angle is also used in a robot presented by Xu and Sun (2000). Two other concepts present a different point of view but both involve designing a kind of an active stabilisation and a dynamic control system. One interesting idea with a prototype construction is described in a paper titled "Three Dimensional Posture Control of Mono-wheel Robot with Roll Rotable Torso" (Fujimoto and Uchidam 2007). 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 if the controllers actually work for the built prototype as it was considered a future work by the authors.

The most advanced prototype in terms of production readiness was created by Honda and called U3-X [6]. It is a personal transport vehicle which stabilises similarly to the Segway. It has a small wheel at the bottom and a seat above. What makes it possible to maintain upright position is a special construction of the wheel. It is a kind of an omnidirectional wheel whose rim has a modular structure build of small rings rotating perpendicularly to the whole wheel. It enables the wheel to move forward, backward and to the sides and it is used to change the fulcrum of the robot structure and 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 himself in 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 by itself, it needs an operator to change its centre of gravity and force movement in a desired direction. Even as a manned vehicle, it is rather slow but of good manoeuvrability. The robot described just before, the model U3-X, consisting of a small wheel and a two-part body, shares some cons with Honda's design. Its structure suggests that it cannot be very fast and rather hard to control between obstacles because of 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, batteries could only be mounted inside it, which would make them exposed to damage. Three vehicle concepts described at the beginning and our own design have a lot in common. They all can easily achieve high speeds, their structure is compact, all their electronic components fit inside the wheel and they should be able to rise from a fall. What is the main difference is that they do not consider any active stabilisation and control system with feedback from sensors, which is one of our main goals. What we want to create is a robot having all of the mentioned features and being able to maintain upright position for unlimited time when stopped or moving. It has to be easily controllable by a radio transmitter, a computer or even move by itself. In a wider perspective, it has to be adaptable to different equipment configurations and environment conditions.

2. Mathematical model of the robot

Mobile robots utilise different methods in order to move i.e. wheels, caterpillars, pedipulators, however the use of the solutions based on wheels is the most energy efficient for such mechatronic systems. A very interesting issue is the motion of an nonholonomic mobile robotic systems (Iagnemma and Dubowsky, 2004). This type of robots have a less quantity of drives with respect to the number of degrees of freedom. If we take the idea of a single-wheel robot into consideration, it is possible to describe it with the use of the following coordinates

$$\boldsymbol{q} = [x, y, \theta]^\top \tag{2.1}$$

where first two coordinates, presented in the fig.1 describe the position of the robot in the plane, while the third shows the orientation angle of the wheel (Dulęba, 2001; Tchoń *et al.*, 2000).

The control of the robot will be carried out by changing the forward velocity u_1 and the speed of orientation angle u_2. The required assumption in

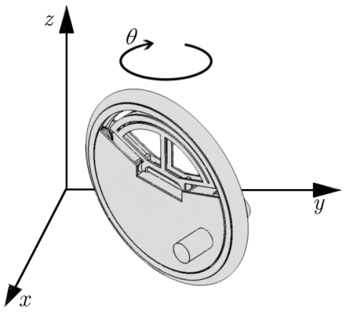

Fig. 1. Description of the position and orientation angle for the single-wheel robot

our system is that there is no skid between the robot wheel and the ground. In such a case, the natural limitation of motion can be described as

$$\dot{x}\sin\theta - \dot{y}\cos\theta(+d\theta\cdot 0) = 0 \tag{2.2}$$

Constraint (2.2) is connected with the class of the phase constraints in a form of Pfaff's transformation

$$\mathbf{A}(\boldsymbol{q})\dot{\boldsymbol{q}} = \mathbf{0} \tag{2.3}$$

where

$$\mathbf{A}\left(\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}\right) = \begin{bmatrix} \sin\theta \\ -\cos\theta \\ 0 \end{bmatrix} \tag{2.4}$$

The constraints are holonomic if the vector of the function $\boldsymbol{\Phi}(\boldsymbol{q})$ exists, which meets the following condition, where the non-singular matrix $\mathbf{M}(\boldsymbol{q})$ is the precision

$$\frac{\partial\boldsymbol{\Phi}(\boldsymbol{q})}{\partial\boldsymbol{q}} = \mathbf{M}(\boldsymbol{q})\mathbf{A}(\boldsymbol{q}) \tag{2.5}$$

If condition (2.5) is not met, the constraint is called nonholonomic. For the presented structure of the single-wheel robot, if the function $\boldsymbol{\Phi}(\boldsymbol{q})$ exists, it should conform to the following dependence

$$\frac{\partial\boldsymbol{\Phi}(\boldsymbol{q})}{\partial\boldsymbol{q}} = \begin{bmatrix} \dfrac{\partial\boldsymbol{\Phi}}{\partial x} \\ \dfrac{\partial\boldsymbol{\Phi}}{\partial y} \\ \dfrac{\partial\boldsymbol{\Phi}}{\partial\theta} \end{bmatrix} = m(\boldsymbol{q})\begin{bmatrix} \sin\theta \\ -\cos\theta \\ 0 \end{bmatrix} \tag{2.6}$$

where the function $m(\boldsymbol{q}) \neq 0$.

Comparing the mixed partial derivatives calculated for the first two conditions of dependence (2.6) yields a false result

$$\frac{\partial^2 \boldsymbol{\Phi}(\boldsymbol{q})}{\partial x \partial \theta} = \frac{\partial^2 \boldsymbol{\Phi}(\boldsymbol{q})}{\partial \theta \partial x} \tag{2.7}$$

The conclusion from the mathematical description of the single-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. Our assumption that the single-wheel robot structure is nonholonomic has been proven on the basis of formulas and transformations (2.3)-(2.7). Equation (2.2) shows that the vector $\dot{\boldsymbol{q}} = [\dot{x}, \dot{y}, \dot{\theta}]^\top$ must be always perpendicular to the vector $[\sin\theta, -\cos\theta, 0]^\top$ during motion. It means that the velocity vector $\dot{\boldsymbol{q}}$ belongs to the null space of the matrix $\mathbf{A}(\boldsymbol{q})$. If we assume the base in the null space, constructed from the two vectors $[\cos\theta, \sin\theta, 0]^\top$ and $[0, 0, 1]^\top$, where each one is perpendicular to the vector $[\sin\theta, -\cos\theta, 0]^\top$, than we receive a kinematic model of control of the single-wheel robot as follows (Dulęba, 2001)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_2 \tag{2.8}$$

The single-wheel robot is controlled according to presented kinematic model (2.8) by the influence on forward velocity u1 and the orientation angle speed u_2. By the assumption of the base in the null space of the matrix $\mathbf{A}(\boldsymbol{q})$ (2.4), we transform our description from the space of directions created by the vector $[\sin\theta, -\cos\theta, 0]^\top$ (in which motion was not possible) to the space that permits the motion (Dulęba, 2001). The characteristic feature of the nonholonomic constraints is controllability of the system despite the number of controls is significantly less than the space state dimension. As it was proven for our system, there are two controls u_1, u_2 and three dimensional state space, however the single-wheel robot is able to reach any position on a surface with any orientation. Another feature of the nonholonomic system is that if we are trying to make a linearisation of the kinematic model of control based on nonholonomic constraints, it is always uncontrollable in the equilibrium point. It is connected with the deficiency of the control dimensions referenced to the state space dimensions. The linearisation of kinematic model of control (2.8) of the single-wheel for the equilibrium point, where the controls are $[u_1, u_2]^\top = [0, 0]^\top$, can be written in the following form (Dulęba, 2001)

$$\begin{bmatrix} \delta\dot{x} \\ \delta\dot{y} \\ \delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \delta u_2 \end{bmatrix} = \mathbf{B} \begin{bmatrix} \delta u_1 \\ \delta u_2 \end{bmatrix} \tag{2.9}$$

Linear approximation (2.9) is not controllable, because the rank of $\mathbf{B}$ is equal 2 and it means that this value is less than our three dimensional Euclidean space. As a result of mathematical description of the single-wheel robot presented above, there was a need to design a balancing system responsible for self-stabilisation of the robot. The model of the system in the lateral plane is presented in Fig. 2.

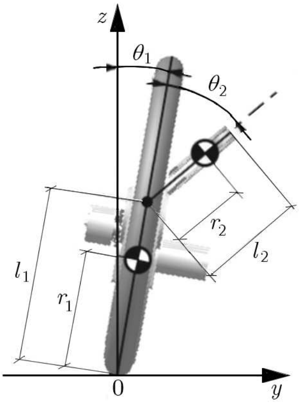

Fig. 2. The model of the system in the lateral plane

The model of the single-wheel robot in the lateral plane is equivalent to a double inverted pendulum whose first joint is underactuated as it is shown in Fig. 2. The idea of a double inverted pendulum is based on motion phenomena where the equilibrium of the system and the orientation changing is dependent on the balancing lever, it means that the lever and wheel balancing dependently and are connected. The equation of motion of the model for the assumptions: $\boldsymbol{x}_l = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]^\top$, $\boldsymbol{\theta}_l = [\theta_1, \theta_2]^\top$, $\boldsymbol{\tau}_l = [0, \tau_2]^\top$ in the lateral plane, can be written as follows (Fujimoto and Uchida, 2007)

$$\dot{\boldsymbol{x}}_l = \begin{bmatrix} \dot{\boldsymbol{\theta}}_l \\ \mathbf{M}_l[\boldsymbol{\theta}_l)^{-1}(\boldsymbol{\tau}_l - \boldsymbol{h}_l(\boldsymbol{\theta}_l, \dot{\boldsymbol{\theta}}_l) - \mathbf{G}_l(\boldsymbol{\theta}_l)] \end{bmatrix} \tag{2.10}$$

where

$$\mathbf{M}_l(\boldsymbol{\theta}_l) = \begin{bmatrix} I_1 + I_2 + m_2(l_1^2 + 2l_1 r_2 \cos\theta_2) & I_2 + m_2 l_1 r_1 \cos\theta_2 \\ I_2 + m_2 l_1 r_2 \cos\theta_2 & I_2 \end{bmatrix}$$

$$\boldsymbol{h}_l(\boldsymbol{\theta}_l, \dot{\boldsymbol{\theta}}_l) = \begin{bmatrix} -m_2 l_1 r_2 \sin\theta_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) + D_1\dot{\theta}_1 \\ m_2 l_1 r_2 \dot{\theta}_1^2 \sin\theta_2 + D_2\dot{\theta}_2 \end{bmatrix}$$

$$
\begin{aligned}
\mathsf{G}_l(\boldsymbol{\theta}_l) = -g & \begin{bmatrix} (m_1r_1+m_2l_1)\sin\theta_1 + m_2r_2\sin(\theta_1+\theta_2) \\ m_2r_2\sin(\theta_1+\theta_2) \end{bmatrix} \\
+g_2 & \begin{bmatrix} (m_1r_1+m_2l_1)\cos\theta_1 + m_2r_2\cos(\theta_1+\theta_2) \\ m_2r_2\cos(\theta_1+\theta_2) \end{bmatrix}
\end{aligned} \tag{2.11}
$$

The construction data necessary to calculate equation of motion (2.10) consist of: l_1 – distance from the ground to the rotational joint of the lever, l_2 – length of the lever [m], m_1, m_2 – mass of first and second element [kg], θ_1 – angle of the wheel inclination [rad], θ_2 – angle of the lever rotation [rad], τ_2 – input torque to the rotational joint [Nm], r_1, r_2 – position of the center of mass of the wheel and the lever [m], I_1, I_2 – inertia of the wheel and the lever [kgm^2], D_1, D_2 – damping factor of the wheel inclination and the lever rotation [Nms/rad], g – gravity acceleration [m/s^2], g_2 – centrifugal acceleration [m/s^2].

After linearisation of nonlinear equation (2.10) around the equilibrium point, where $\theta_1 = \tan^{-1}(g_2/g)$, $\dot{\theta}_1 = \theta_2 = \dot{\theta}_2 = 0$, the kinematic model of control for the lateral plane can be presented as follows (Fujimoto and Uchida, 2007)

$$
\dot{\boldsymbol{x}}_l = \mathsf{A}_l\boldsymbol{x}_l + \boldsymbol{B}_l\tau_2 \qquad \boldsymbol{y}_l = \mathsf{C}_l\boldsymbol{x}_l \tag{2.12}
$$

where

$$
\begin{aligned}
\mathsf{A}_l &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{S_l}\sqrt{g^2+g_2^2}a_1 & -\frac{D_1I_2}{S_l} & -\frac{1}{S_l}\sqrt{g^2+g_2^2}a_5 & \frac{D_2a_6}{S_l} \\ 0 & 0 & 0 & 1 \\ \frac{1}{S_l}\sqrt{g^2+g_2^2}a_2 & \frac{D_1a_6}{S_l} & \frac{1}{S_l}\sqrt{g^2+g_2^2}a_4 & \frac{-D_2a_3}{S_l} \end{bmatrix} \\
\boldsymbol{B}_l &= \begin{bmatrix} 0 \\ -\frac{a_6}{S_l} \\ 0 \\ \frac{a_3}{S_l} \end{bmatrix} \qquad \mathsf{C}_l = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}
\end{aligned} \tag{2.13}
$$

and

$$
\begin{aligned}
& a_1 = I_2(m_1r_1+m_2l_1) - m_2^2l_1r_2^2 \\
& a_2 = m_2r_2[I_1 - l_1(m_1r_1 - m_2r_2)] - I_2(m_1r_1+m_2l_1) \\
& a_3 = I_1 + I_2 + m_2l_1^2 + 2m_2l_1r_2 \qquad a_4 = m_2r_2(I_1 + m_2l_1^2 + m_2l_1r_2) \\
& a_5 = m_2^2l_1r_2^2 \qquad a_6 = I_2 + m_2l_1r_2 \qquad S_l = I_2(I_1+m_2l_1^2) - m_2^2l_1^2r_2^2
\end{aligned} \tag{2.14}
$$

The kinematic model of the robot enables one to describe self-stabilisation phenomena, based on the double inverted pendulum and robot motion. The order of the state variable for the robot model is defined as $\boldsymbol{x}_l = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]$.

3. Mechanical structure

The mechanical structure of the robot described in this paper consists of several major components, presented in Fig. 3, 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 a 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.

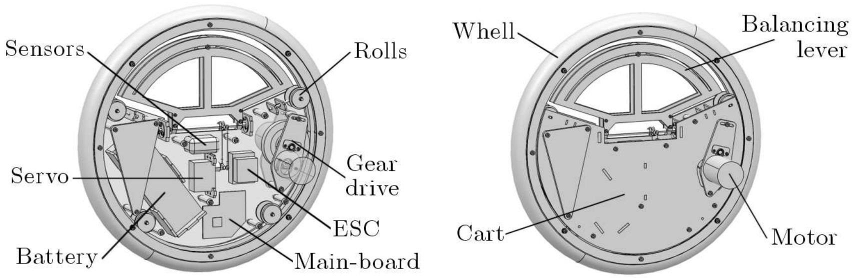

Fig. 3. The mechanical construction of the single-wheel robot

The balancing mechanism consists of an arc shaped lever with a slot for mounting the 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 torque 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 correct distances between the shaft axes. 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. 4, it is tethered to a personal computer for control and monitoring purposes.

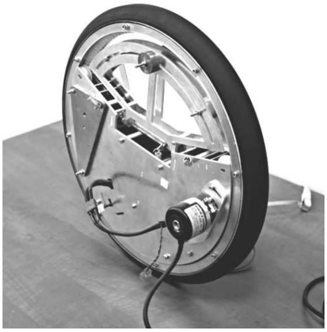

Fig. 4. The prototype of the robot during the phase of self-stabilization

4. Electronic structure

The electronic structure consists mainly of elements taken from radio controlled models, these are: a digital, high torque, high speed servomechanism, a brushless DC motor coupled with an electronic speed controller (sensored design), a Lithium-Polymer, high discharge rate battery and a 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 addition of a few simple electronic parts, it is also able to 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-stabilization of the robot is shown in Fig. 5.

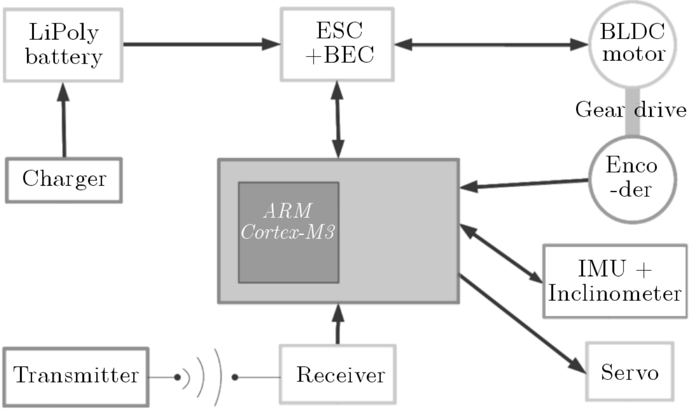

Fig. 5. The scheme of the electro-mechanical system

A very important ability of the presented structure is to allow wireless communication with a personal computer or a human operator (Braunl, 2008; Fahimi, 2009). Everything is powered by the robot battery.

5. Simulation

A multi-body simulation of the design was done to check whether it is correct, prior to building a prototype. The simulation process involved using two connected computational environments – MD Adams and Matlab/Simulink. The idea of the simulation workflow is shown in Fig. 6.

The first stage was creating a mechanical model in Adams View and determining the state variables – model inputs and outputs. 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 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 angular position and the motor rotating speed. The outputs of the model are its tilt in the longitudinal and lateral planes in reference to the ground plane

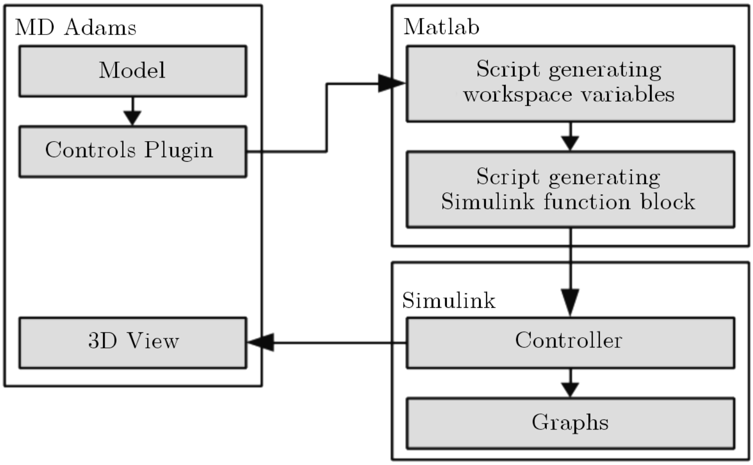
MD Adams
Model
Controls Plugin
3D View
Matlab
Script generating workspace variables
Script generating Simulink function block
Simulink
Controller
Graphs

Fig. 6. The scheme of the simulation procedure

and the torque acting on the servomechanism and the motor. 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 connects 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 rotation 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. 7, there is a presentation of the Simulink control model and the result of its work as a 3D visualisation of motion.

Although the full model of the vehicle was built, we mainly focused on the problem of maintaining it 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 the 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. That is why the controller is only connected to the servomechanism angle input and the tilt in the lateral plane output. There are a few important elements in the Simulink block diagram which simulate the parameters of real electronics used in the design. These are Rate Transition blocks which change the sampling frequency of output signals from the model so that the controller works with a speed possible to achieve using the chosen microcontroller and the Transfer Function block simulating the positioning speed

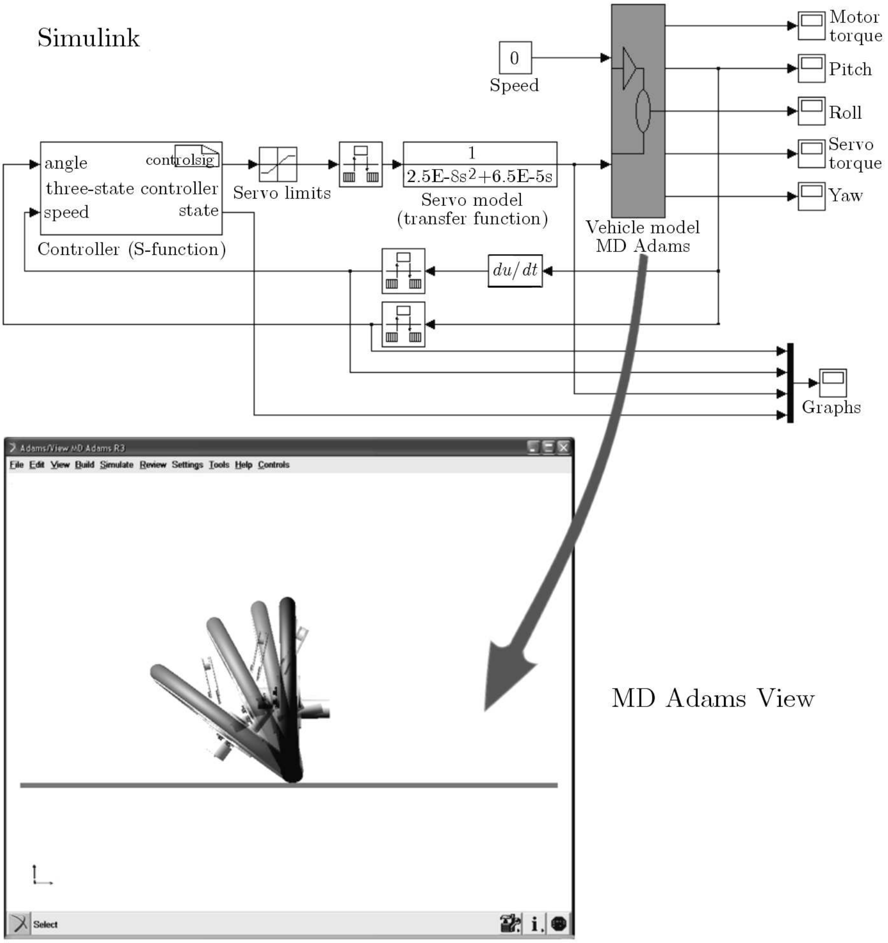

Fig. 7. The coupled Matlab/Simulink and MD Adams simulation of the robot dynamics

of the servomechanism. Different controller types were tested, and it turned out they all have problems in stabilising the robot. The first that we used was a proportional controller, then we added integral and derivative parts, but it did not work either. The second attempt was to use a fuzzy logic controller but it was hard to determine correct rules. Finally, we decided to build a custom controller using a 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 value, the tilt angle and the angular speed multiplied by the experimentally received factors. What is significant is that it takes prior control values into account, not only the control deviation. It is presented as a block diagram in Fig. 8.

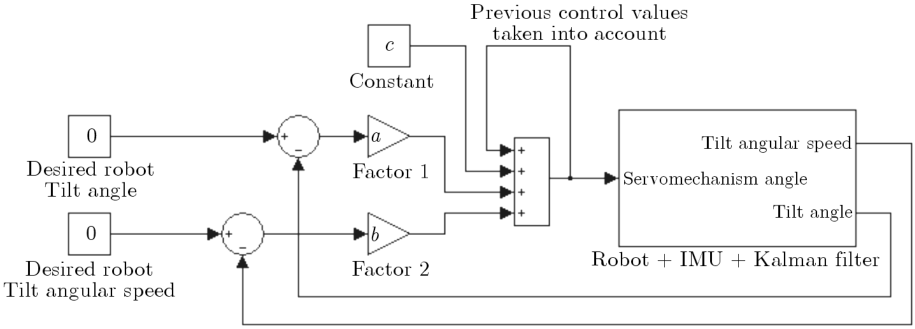

Fig. 8. Block diagram of the robot controller formula - stabilisation in the lateral plane

Using the developed custom controller, we were able to achieve upright position for an unlimited period of time.

6. Control system implementation

A prototype was built when the simulations showed that the designed structure works properly and a control algorithm was then implemented on the ARM CortexM3 micro-controller (Braunl, 2008). 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 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 (Mohinder and Angus, 2008). 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. The typical Kalman filter algorithm (Iagnemma and Dubowsky, 2004; Mohinder and Angus, 2008) is presented in Fig. 9.

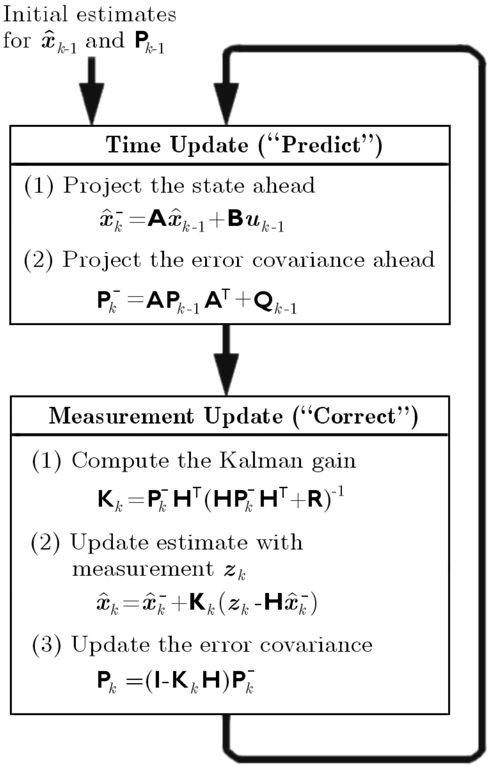
Initial estimates for $\hat{\boldsymbol{x}}_{k-1}$ and $\mathbf{P}_{k-1}$
Time Update ("Predict")
(1) Project the state ahead
$\hat{\boldsymbol{x}}_k^- = \mathbf{A}\hat{\boldsymbol{x}}_{k-1} + \mathbf{B}\boldsymbol{u}_{k-1}$
(2) Project the error covariance ahead
$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}_{k-1}$
Measurement Update ("Correct")
(1) Compute the Kalman gain
$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T(\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1}$
(2) Update estimate with measurement $\boldsymbol{z}_k$
$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \mathbf{K}_k(\boldsymbol{z}_k - \mathbf{H}\hat{\boldsymbol{x}}_k^-)$
(3) Update the error covariance
$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-$

Fig. 9. The Kalman filter algorithm

As one can see, it is a two-stage recurrent algorithm. In the first stage we predict new values of the monitored parameters based on last values and the measurements 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 is the influence 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 $(z_k - \mathbf{H}\hat{\boldsymbol{x}}_k^-)$. Finally, we compute corrected values of the covariance matrix. In our implementation of the described algorithm, we decided to monitor three parameters: the tilt, the angular velocity and the gyroscope bias, therefore our state vector is

$$\boldsymbol{x} = [\theta_1, \dot{\theta}_1, g_{bias}]^\top \tag{6.1}$$

The values of the particular matrices are as follows

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 1 & 0 & -dt \\ 0 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} dt \\ 1 \\ 0 \end{bmatrix} \\ \mathbf{H} &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{aligned} \tag{6.2}$$

The variance matrices look like this

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \boldsymbol{q} \tag{6.3}$$

where $\boldsymbol{q}$ is the gyroscope noise.

$R = r$, where r is the accelerometer noise. The covariance matrix is initialized as follows

$$\mathbf{P} = \begin{bmatrix} R & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6.4}$$

and the state vector $\boldsymbol{x}$ 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 the 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 (Biagiotti and Melchiori, 2008). It introduces a slight delay in servomechanism movements, but it effectively protects the drive from overstress. 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.

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 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. The main-board programming is done using a JTAG interface working with Eclipse IDE. The realtime 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 this 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 work.

The experimental results presented below show two cases which may occur during the halt period or the movement period of the robot with a velocity lower than 0.3 m/s. In these phases of robot movement, the gyroscopic effect influence is really small and the robot loses technical stability. The first case is connected with a force acting on the robot, whose value exceeds the critical value, resulting in the robot inclinating more than 5 degrees and loosing stability. This situation may occur either as a result of this kind of force applied to the highest point of the robot and perpendicular to the wheel surface or as a result of reaching the maximum inclination of the balancing lever and keeping this position in relation to the surface of the wheel for too long. In Fig. 10 and Fig. 11, the time courses of the basic parameters related to the stability for thes cases, where the acting force exceeds the critical value, are presented.

As you may notice in the graphs shown, the mentioned force action appears in time $t = 41\,\mathrm{s}$, which causes a balancing lever reaction in order to compensate the inclination. As a result of the too high robot inclination from the vertical plane, there occurs a technical stability loss, which is proven by exponentially increasing values of the parameters. The second case shows the robot behaviour during normal work conditions, where the balancing lever rotates in order to effectively create self-stability by reducing the inclination from the vertical plane. The time courses of the parameters connected with the stability for this case are presented in Fig. 12 and Fig. 13. As it can be noticed, the robot is theoretically able to create self-stability for an unlimited period of time, but in practice it is limited by the energy supply necessary to power the actuators and electronic devices.

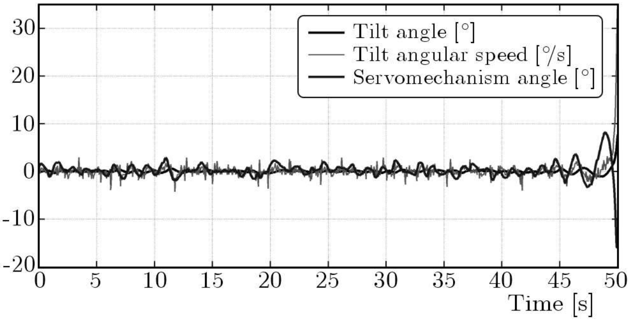

Fig. 10. Time courses of the tilt angle, tilt angular speed and servomechanism angle presented in the common graph for a test rig with a force greater than the critical

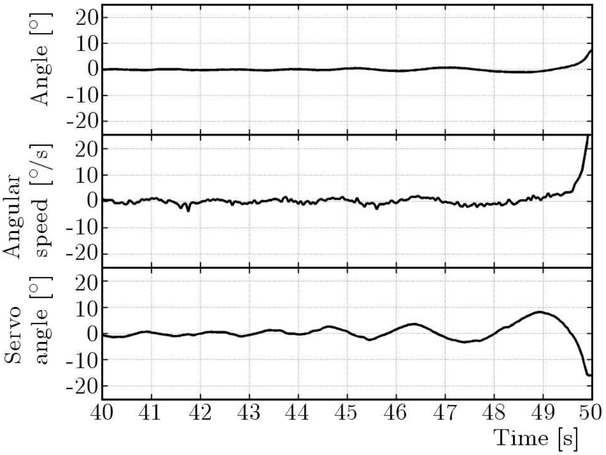

Fig. 11. Time courses of the tilt angle, tilt angular speed and servomechanism angle split into separate graphs

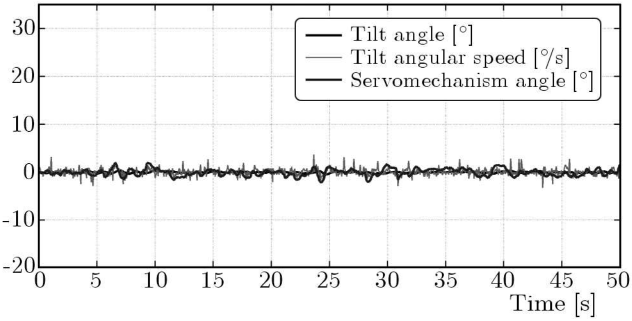

Fig. 12. Time courses of the tilt angle, tilt angular speed and servomechanism angle presented in the common graph for the test rig without disturbances

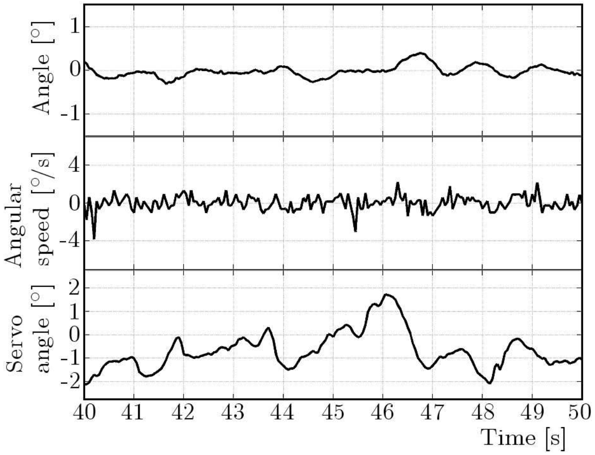

Fig. 13. Time courses of the tilt angle, tilt angular speed and servomechanism angle split into separate graphs

The range of works presented in this article made it possible to achieve technical stability of a complex mechatronic unit which is a single-wheel mobile robot at the most difficult stage of the robot operation. The fact of the complication of this stage is connected to the minimal influence of the effect of sustaining vertical position of the robot wheel resulting from the gyroscopic effect (low angular velocity of the wheel) or its lack.

8. Conclusion

In this paper, a mathematical description of a single-wheel robot was presented. Then a mechanical design of a particular robot was described throughly. The electronic structure was also discussed to complete the robot description. The whole simulation process and its results where 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 technical stability of the robot was proved. The future works on the described design will consist of working out a turning algorithm and increasing the 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. This paper proposes a different solution to design and drive the single-wheel robot in relation to those described in the bibliography.

References

1. Biagiotti L., Melchiorri C., 2008, *Trajectory Planning for Automatic Machines and Robots*, Springer-Verlag Berlin, Heidelberg
2. Braunl T., 2008, *Embeded Robotics – Mobile Robot Design and Application with Embedded Systems*, Springer-Verlag Berlin, Heidelberg
3. Dulęba I., 2001, *Metody i algorytmy planowania ruchu robotów mobilnych i manipulacyjnych*, Akademicka Oficyna Wydawnicza EXIT, Warszawa
4. Fahimi F., 2009, *Autonomous Robots – Modeling, Path Planning and Control*, Springer
5. Fujimoto Y., Uchida S., 2007, Three dimensional posture control of monowheel robot with roll rotatable torso, *Proceedings of International Conference on Mechatronics*, Kumamoto Japan
6. *Honda U3-X: Reinventing the wheel*, http://www.fasterandfaster.net/2009/09/honda-u3-x-reinventing-wheel.html
7. Iagnemma K., Dubowsky S., 2004, Mobile robots in rough terrain, *Springer Tracts in Advanced Robotics*, **12**, Springer-Verlag Berlin, Heidelberg
8. Mohinder S.G., Angus P.A., 2008, *Kalman Filtering: Theory and Practise Using MATLAB*, Wiley
9. Greenley P., Rehkemper J., 2008, *Mono-wheel Vehicle with Tilt Mechanism*, United States Patent 7337862
10. Tchoń K., Mazur A., Dulęba I., Hossa R., Muszyński R., 2000, *Manipulatory i roboty mobilne – modele, planowanie ruchu, sterowanie*, Akademicka Oficyna Wydawnicza PLJ, Warszawa
11. Xu Y., Brown B., *A Single-Wheel, Gyroscopically Stabilized Robot (GYROVER)*, http://www.cs.cmu.edu//afs/cs/project/space/www/gyrover/gyrover.html
12. Xu Y., Brown B., 1997, A single-wheel, gyroscopically stabilized robot, *IEEE Robotics and Automation Magazine*, **4**, 3, 39-44
13. Xu Y., Ou Y., 2002, Balance control of a single wheel robot, *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Switzerland

14. Xu Y., Sun L.W., 2000, Dynamics of a rolling disk and a single wheel robot on an inclined plane, *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Japan

Samostabilizujący się uniwersalny jednokołowy robot

Streszczenie

Praca przedstawia zgłoszoną do opatentowania konstrukcję jednokołowego robota anholonomicznego, wyposażonego w układ sterowania oraz system stabilizacji bazujący na budowie odwróconego wahadła fizycznego.

Manuscript received December 1, 2010; accepted for print May 13, 2011