Walking robot modelling aspects

Marcin Piątek* and Dariusz Marchewka*

*AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Department of Automatics
Mickiewicza 30 av. 30-059 Kraków, Poland
e-mail: marcin.piatek@agh.edu.pl, dariusz.marchewka@agh.edu.pl

***Abstract*—The paper describes a walking robot leg modelling process. A six-legged walking robot — the so-called hexapod is concerned. Each leg consists of three links driven by Hitec HS-475HB servo motors. The proposed model is used for testing leg kinematics and dynamics of the robot gait. One can examine a number of walking algorithms. The model includes: the identified description of the servo motors, a full state observer (position and velocity), forward kinematics of the position and the velocity, inverse kinematics of the position and the leg movement visualisation. The model structure is explained and depicted. The simulation results are shown in a graphical form. The examples of model applications are applied and described. The advantages and disadvantages of the model are listed. Eventual experiments and applications are announced.**

***Index Terms*—Walking robot, modelling, simulation, identification.**

I. Introduction

Scientist and engineers have been interested in the walking robots since a long time. There are benefits of walking legs in a contrary to rolling on wheels or moving on tracks. Walking legs can move on almost every surface like rock debris, ocean floor, surface of the other planets and such a simple obstacle like the stairs. However, one can find walking legs drawbacks: a slow motion, high power and a lot of identical elements to be used.

As far as motion is concerned the walking robots resemble animals or more precisely insects.

One of the simplest construction is hexapod — a robot equipped with the six legs, usually three on each side. Thanks to the leg numbers we can easily design walking algorithms in this way that the whole construction will be stable in every walk phase (in the contrary to the four-legged and the two-legged robots).

The six-legged walking robot application discussed in this paper was designed and built to model the walking patterns observed in the six-legged insects (Figure 1). As an example the *Blatta Orientalis* was chosen, it is also known as a cockroach. This species are very well examined by biologists and there is a lot of paper devoted to insect walking patterns [8].

The insect legs motion is controlled at a low level of the neural structure. It means that a high neural structure level is free from signals devoted to walking algorithms. These algorithms have a very simple form. Usually correspond to synchronously repetitive actions.

Figure 1. The hexapod — six-legged walking robot.

In the case of the six-legged insects we consider three characteristic ways of walking:

- the insect moves only one leg in a time instance, the other five legs support the insect body
- the insect moves two legs in a time instance and the other four legs support the insect body
- the insect moves three legs in a time instance, the other three legs support the insect body

These three ways of moving legs will be considered as the three different modes of the insect walk and apply also to the six-legged walking robots. Their analysis will help to implement efficient algorithms of walk for a hexapod.

II. The algorithms of the walk

The plane surface without any imperfections is considered. In Figure 2 we can see the scheme of the robot leg. Every leg consists of the three links driven by the servo motors. The numbered legs are shown in Figure 3. Figure 4 shows the way how the legs are moved in the each walking mode. The full cycle consist of six stages in the mode a), three stages in the mode b) and only two stages in the mode c). The analysis of the insect motion helps to develop walking algorithms for the hexapod. One can conclude:

- at the same velocity of the single leg move the robot can achieve three different velocities by changing the walk mode
- the robot can move forward or backward due to the applied sign of the control signal

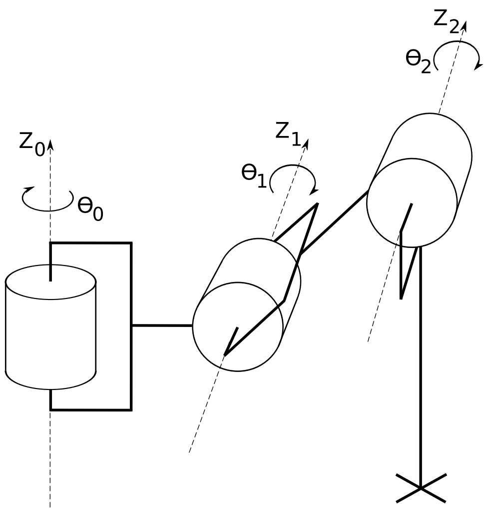

Figure 2. The scheme of the robots leg connections.

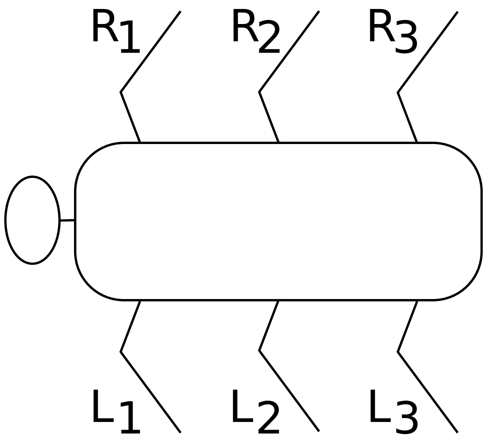

Figure 3. The schematic structure of the six-legged insects.

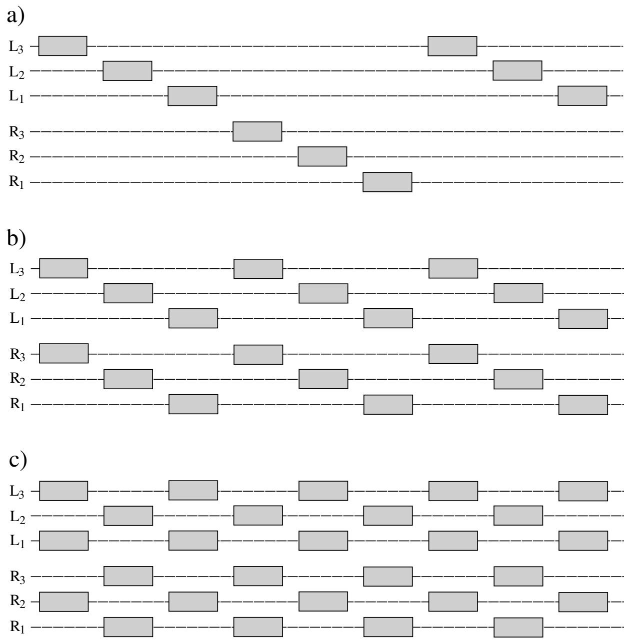

Figure 4. The scheme presenting three basic modes of the six-legged insects walk.

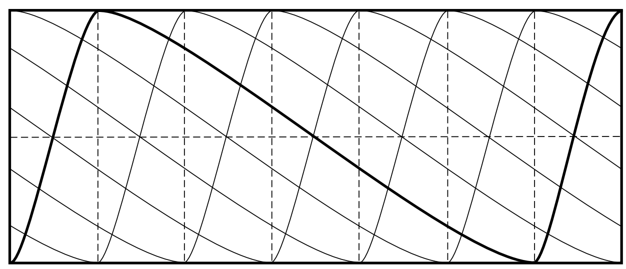

Figure 5. The first DOF control signals in the a) mode of the walk.

- the higher number of legs raised, the faster the robot can move but the legs on which the robot is leaning are more burdened
- when a leg (or legs) are moving forward, the supporting legs have to move backward with the velocity given by the formula

$$v_b = v_f \frac{n}{6-n} \quad (1)$$

  where v_f is the velocity of the legs moving forward and n is the number of the legs moving forward.

The considered velocities relate to velocity of the robot leg tip. The last observation corresponds to the diagrams presented in Figures 5, 6 and 7. They show positions of the first degree of freedom (DOF) of the robot legs vs. time in each mode of the walk.

Figure 5 presents the legs motion in the mode a). It shows that the legs move forward one by one. The position of the one leg is drawn by the bold line to make the diagram more clear. In this mode every leg moves separately and we can easily distinguish two phases: the fast phase — when the leg moves forward and the slow phase — when the leg moves backward. According to equation 1 the leg velocity when it is moving backward is five times smaller then the velocity when it is moving forward.

Figure 6 presents the legs motion in the mode b). In this mode the legs move being coupled in such a way that in the same time instance two of them move forward and the other four supports the robot. The coupled legs move synchronously. It is possible to see only one leg for each couple. The combination presented in Figure 4 is not the only one possible. The legs could be arranged in couples in few different ways. The only requirement is that construction has to be stable in every phase of the walk. The velocity of the legs when moving backward is two times smaller then in the case when moving forward.

Figure 7 presents the legs motion in the mode c). In this mode legs are arranged by three. When three legs move forward the other three move backward. This is the fastest possible way of the walk of the hexapod. To make the construction stable in every phase of the walk there is only one possible way to arrange legs in three. The middle leg from one side has to move synchronously to the front and the back leg from the other side. The velocity of the moving backward legs is the

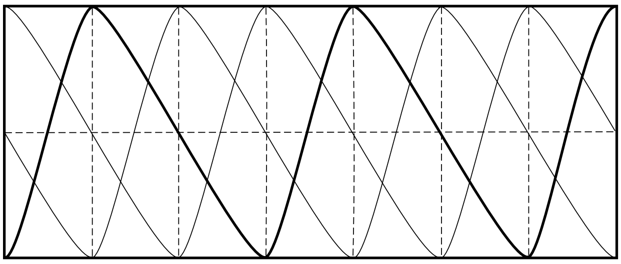

Figure 6. The first DOF control signals in the b) mode of the walk.

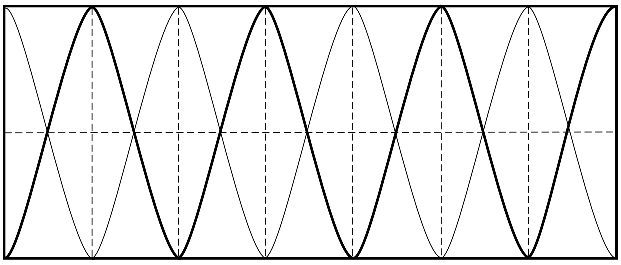

Figure 7. The first DOF control signals in the c) mode of the walk.

same as the legs that are moving forward.
It is important to notice that absolute value of velocity of the robot is equal to the absolute value of velocity of the moving backwards legs.

III. The one leg move

The previous diagram shows only the position vs. time of the first DOF of the legs. To simulate and eventually build the whole walking robot based on the presented algorithm we have to consider motion of the two remaining DOF. It is possible to find it out from the position of the leg tip by the inverse kinematics procedure. Figure 8 presents the trajectory of the robot leg tip during the walk. The leg tip moves only in the two dimensions: the robot front–rear and up–down. The trajectory has a shape of the ellipse where its one diameter is the step length and the other diameter is the step height. In the upper part of the trajectory the leg moves with the velocity equal to v_f and in its lower part the velocity is equal to v_b.

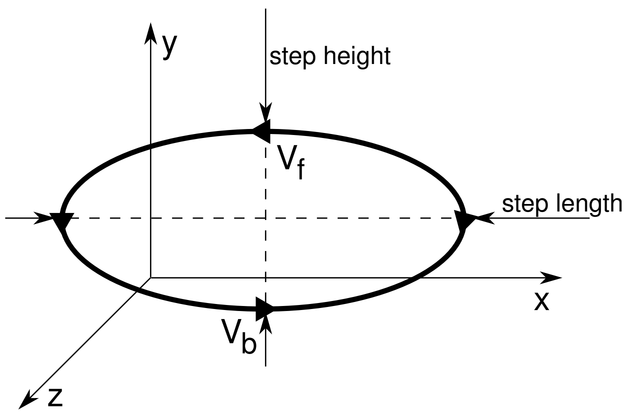

Figure 8. The diagram with the trajectory of the robot leg tip

The conclusion from the diagram is that the position of the second and the third DOF of the robots leg is directly dependent on the position and sign of the velocity of the first DOF. To find the relationship we have to derive the inverse kinematics of the robot leg thus the model of this system was created in the Simulink environment [7] [3].

IV. The robot leg model

The model contains:

- the identified dynamics of the Hitec HS-475HB servo motor
- the state observer to calculate the servo motor velocity and to filter the measured position signal
- the forward kinematics of the robot leg
- the inverse kinematics of the robot leg.

Through the Simulink modelling tool it was possible to identify, simulate and verify the correctness of the created model.
The Hitec HS-475HB servo motor is built with the DC motor, the mechanical gear, the potentiometer (to measure position signal) and the proportional controller in a feedback loop. The model structure is presented on Figure 9. Table I presents names and values of the all parameters. The values were obtained during the identification process [5] [4]. The initial values of the servo motors positions equals $90°$ for the first DOF, $30°$ for the second and $120°$ for the third one. The maximal values for the servo motors positions are equal $\pm 60°$ with respect to the initial positions. The High Gain Observer algorithm was used to estimate the servo motor velocity and to filter the position signal [6]. Figure 10 presents the observer model. Figure 11 presents the architecture of the leg model. In the Figure we can distinguish separate components like: the three servo motors models with the observers, the forward kinematics and the inverse kinematics. The forward kinematics problem of the position and the velocity was solved according to rules described in [7]. The coordinate axes has been designated according to Denavit-Hartenberg convention. Figure 12 presents the leg diagram with the assigned axes. During the modelling procedure of this part some problems were already identified for the future implementation. The most important one is computational complexity of the derived formula. It uses the trigonometric functions (sine and cosine) which are real challenge for the small processors usually embedded in the mobile robots. To simplify the equation entry

Parameter Name	Symbol	Value
DC motor inertia	T_i	0.0181
proportional controller coefficient	k	20.8279
upper control signal constraint	$\overline{S}$	220.9851
lower control signal constraint	$\underline{S}$	-220.854

Table I
The servo motor model parameters

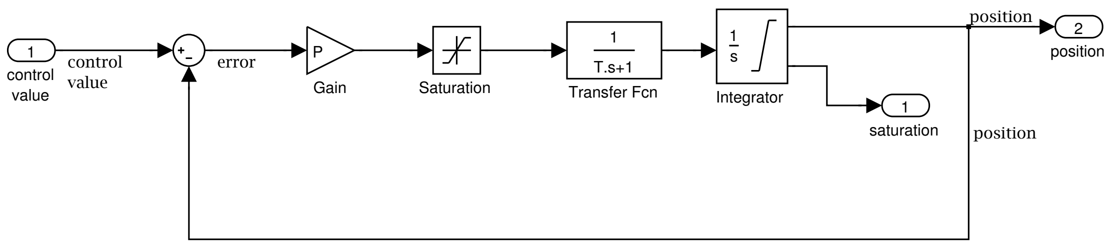

Figure 9. The model of the servo motor

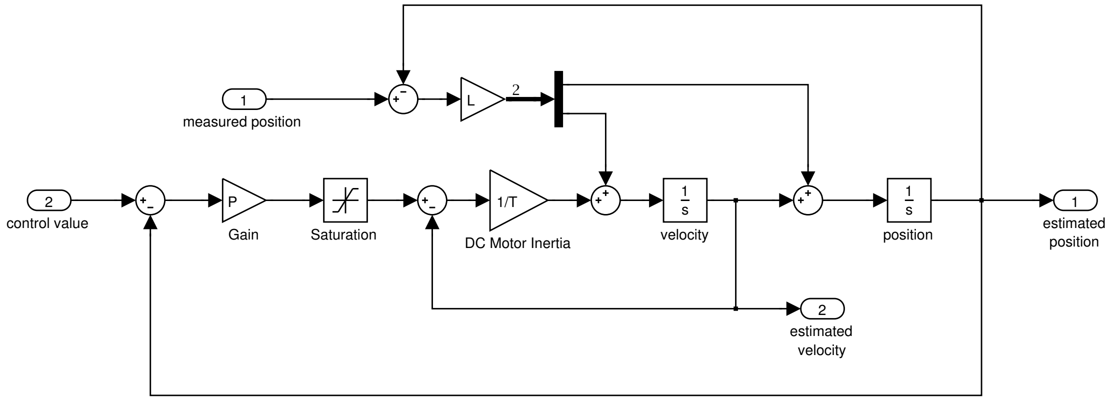

Figure 10. The model of the observer

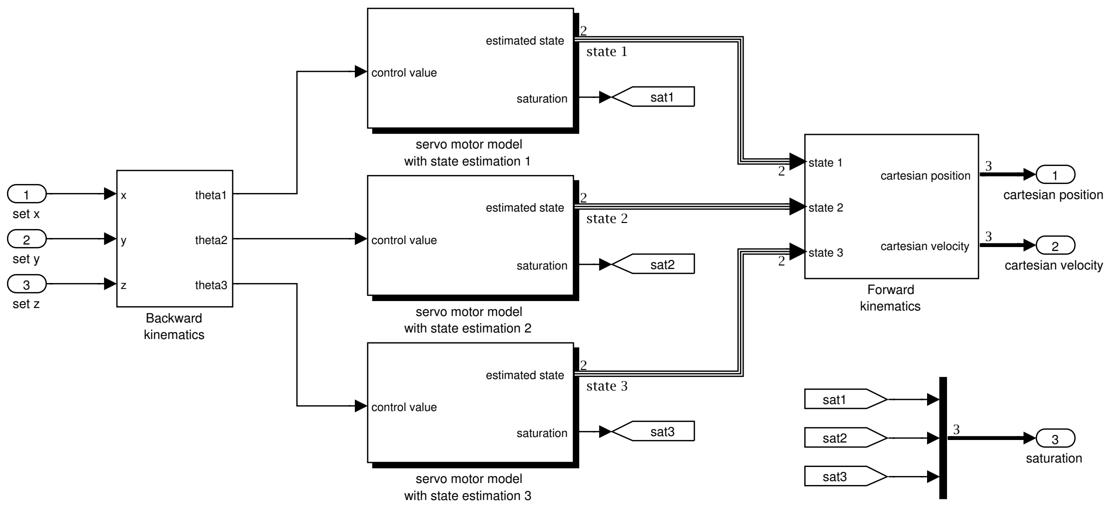

Figure 11. The structure of the leg model

some definitions were made:

$$
\begin{aligned}
s_1 &= \sin(\theta_1)\\
s_2 &= \sin(\theta_2)\\
s_3 &= \sin(\theta_3)\\
&\\
c_1 &= \cos(\theta_1)\\
c_2 &= \cos(\theta_2)\\
c_3 &= \cos(\theta_3)
\end{aligned}
\tag{2}
$$

The equation 3 solves the forward kinematics of the position.

$$
\begin{aligned}
x &= l_3c_1c_2c_3 + l_3c_1s_2s_3 + l_2c_1c_2 + l_1c_1\\
y &= l_3s_1c_2c_3 + l_3s_1s_2s_3 + l_2s_1c_2 + l_1s_1\\
z &= l_3s_2c_3 - l_3c_2s_3 + a + l_2s_2
\end{aligned}
\tag{3}
$$

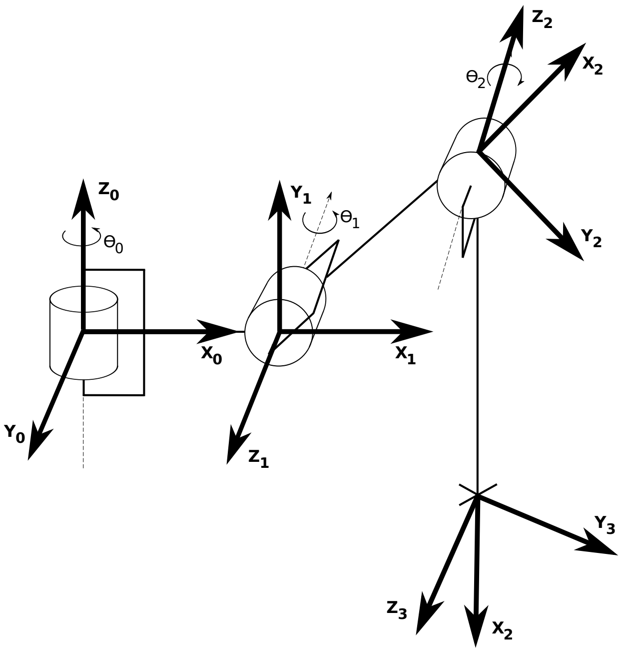

Figure 12. The leg structure with coordinate axes designated in accordance with Denavit-Hartenberg convention

The equation 4 solves the forward kinematics of the velocity.

$$
\begin{aligned}
\dot{x} &= (-l_3 s_1 c_2 c_3 - l_3 s_1 s_2 s_3 - l_2 s_1 c_2 - l_1 s_1) \cdot \dot{\phi}_1 \\
&\quad -c_1 \cdot (l_3 s_2 c_3 - l_3 c_2 s_3 + l_2 s2) \cdot \dot{\phi}_2 \\
&\quad c_1 \cdot (l_3 s_2 c_3 - l_3 c_2 s_3) \cdot \dot{\phi}_3 \\
\dot{y} &= (l_3 c_1 c_2 c_3 + l_3 c_1 s_2 s_3 + l_2 c_1 c_2 + l_1 c_1) \cdot \dot{\phi}_1 \\
&\quad -s_1 \cdot (l_3 s_2 c_3 - l_3 c_2 s_3 + l_2 s_2) \cdot \dot{\phi}_2 \\
&\quad s_1 \cdot (l_3 s_2 c_3 - l_3 c_2 s_3) \cdot \dot{\phi}_3 \\
\dot{z} &= (s_1 * (l_3 s_1 c_2 c_3 + l_3 s_1 s_2 s_3 + l_2 s_1 c_2) \\
&\quad +c_1 * (l_3 c_1 c_2 c_3 + l_3 c_1 s_2 s_3 + l_2 c_1 c_2)) \cdot \dot{\phi}_2 \\
&\quad (-s_1 \cdot (l_3 s_1 c_2 c_3 + l_3 s_1 s_2 s_3) \\
&\quad -c_1 * (l_3 c_1 c_2 c_3 + l_3 c_1 s_2 s_3)) \cdot \dot{\phi}_3
\end{aligned} \tag{4}
$$

The parameters l_1, l_2 and l_3 are the length of the separate leg components and the parameter a is a vertical displacement between the first link and the center of the main coordinates system. The inverse kinematics algorithm is less demanding but the same computational problem will appear in future implementations. The pseudo code listed below presents how the inverse kinematics problem is solved:

$$
\begin{aligned}
&\theta_1 = \arctan\left(\frac{y}{x}\right) \\
&\mathrm{if}(\theta_1 < 0) \\
&\qquad \theta_1 = \theta_1 + \pi \\
&\mathrm{end} \\
&r = \sqrt{x^2 + y^2} \\
&c = (z-a)^2 + (r-l_1)^2 \\
&\theta_3 = \arccos\left(\frac{c - l_2^2 - l_3^2}{2 l_2 l_3}\right) \\
&\alpha = \arccos\left(\frac{l_3^2 - l_2^2 - c}{-2 l_2 \sqrt{c}}\right) \\
&\beta = \arctan\left(\frac{z-a}{r-l_1}\right) \\
&\mathrm{if}((r-l_1) \geq 0) \\
&\qquad \theta_2 = \alpha + \beta \\
&\mathrm{else} \\
&\qquad \theta_2 = \alpha - (\pi - \beta) \\
&\mathrm{end}
\end{aligned}
$$

Presented algorithm was also extended with procedures preventing errors like division by zero and similar.

V. The model usage

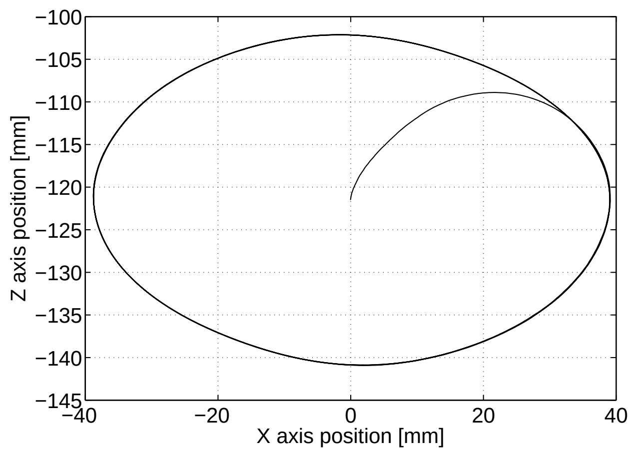

Figure 13. The model of the observer

The model described in the previous section has been and will be used for a few different purposes. One of them is simulation of the observer algorithm, the forward kinematics of the position, the forward kinematics of the velocity and the inverse kinematics of the position. The simulation allows to validate algorithm correctness and verify against different inputs values. The inputs values that can cause the algorithm errors are the most interesting. The errors early identification lets possibly avoid the software reimplementation and even hardware damage in future.

Figures 13 and 14 present the result of the model simulation

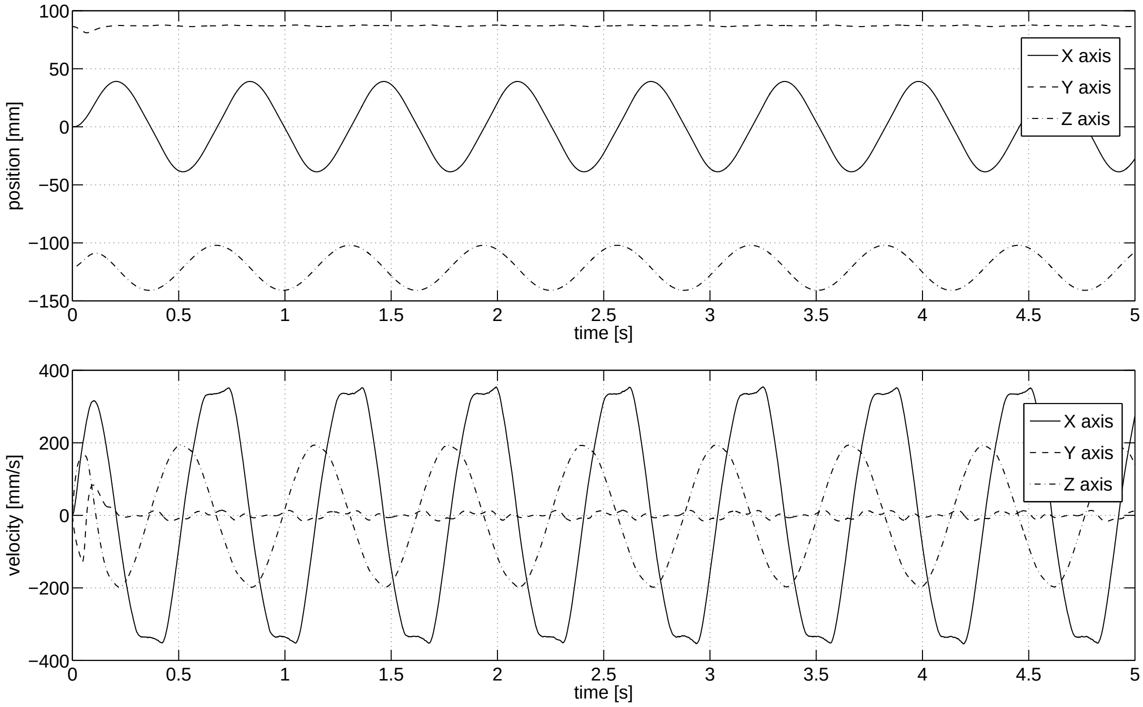

Figure 14. The model of the observer

where the input value to the model was a function generating ellipse trajactory on X and Z axes (like during walking process).

The model can also be used for the code generation, what is one of the Simulink features. The C code could be generated from the prepared model or the parts of the model and then deployed directly to the walking robot control system.

The model is also useful during the leg control algorithm design. As mentioned in the previous sections, the robot leg moves along the ellipse with the given velocity. The model helps to choose the ellipse diameters and the velocities in the different parts of trajectory. It is also useful for testing other control strategies.

As the forward and the inverse kinematics algorithms are computationaly complex it would be almost impossible to implement them on the smaller processors, especially not equipped with the Floating Point Unit (FPU) like ARM7 and ARM9 [1] [2]. As the trajectory of the leg tip is periodic in time, the small processor could be equipped with the Look-Up Table of the servo motor positions over the whole trajectory. The control algorithm could iterate over the Look-Up Table with varying velocity to determine the next servo motors positions. The values retrieved from the array can also be scaled with an appropriate ratio to determine different step sizes. The look-up table can be determined with the designed model during the simulation of the designed control algorithms.

The described applications are going to be implemented in subsequent experiments and projects carried out by the students and the stuff of the AGH-UST.

REFERENCES

[1] Z. Fortuna, B. Macukow, and J. Wąsowski. *Metody numeryczne*. Podręczniki akademickie. Elektronika. Informatyka. Telekomunikacja. Wydawnictwa Naukowo-Techniczne, 7 edition, 2006.

[2] D. Marchewka and M. Piątek. Stosowanie systemów wbudowanych do sterowania robotami mobilnymi. *Półrocznik AGH Automatyka*, 12(2):257–266, 2008.

[3] B. Mrozek and Z. Mrozek. *MATLAB i Simulink. Poradnik użytkownika.* Helion, 2 edition, 2004.

[4] M. Piątek. Identification of the servo motor used in the walking robot. *Półrocznik AGH Automatyka*, 14(1), 2010.

[5] M. Piątek and P. Piątek. Development environment for the walking robot application. In *Computer Methods and Systems 2009*, pages 295–299, Nov. 2009.

[6] M. Piątek and A. Tutaj. Angular velocity estimation in the six-legged walking robot application. In *PPEEm 2009: Fundamental problems of power electronics electromechanics and mechatronics*, Dec. 2009.

[7] M. W. Spong and M. Vidyasagar. *Robot Dynamics and Control*. Wydawnictwa Naukowo-Techniczne, 1997.

[8] T. Zielińska. *Maszyny kroczące*. Wydawnictwa Komunikacji i Łączności, Warszawa, 2005.