Online ISSN: 2367-5357
DOI: 10.7546/itc-2019-0020

Modeling of a Humanoid Biped Robot through Different Approaches

Sh. Sherif, J. Kralev, Ts. Slavov

Key Words: *Humanoid robot; modelling; biped robot; nonlinear model; walking algorithm; multivariable identification.*

Abstract. *In this article, the authors present two different approaches to modeling a biped humanoid robot. The first approach to modeling is based entirely on geometric and physical parameters, using blocks from SimMechanicsTM library in Simulink®, as well as personally constructed blocks by the authors. The other modeling approach consists of several stages: first a basic walking algorithm is implemented on a biped humanoid robot through the programming language C with purpose to record a set of experimental data necessary for the identification procedure. An initial identification with a multi-variable ARX model is then made, and final model is in state-space and is obtained by identification the initial one by the method of the predicted error.*

1. Introduction

Biped walking robots (humanoids) have been developed since 1970. The walking algorithm of a humanoid can be either static or dynamic [1, 2]. In static walking the robot stops its walking motion at each step in order to dampen the oscillations from the previous step. Dynamic walking is smoother because each parameter of the walking cycle is continuously varied such that center of gravity (CG) translation speed is close to constant. The realization of a stable biped robot walking is a considerable challenge even in deterministic environment [3]. The key to successful walking requires the projection of CG onto the ground to lie within the supporting polygon of the two feet. In the case of dynamic walking the same condition is required for the zero moment point. Today popular biped robots are Atlas of Boston Dynamics [4], Asimo of Honda [5], Nao of SoftBank Robotics [6] and Actroid of Osaka University [7].

2. Laboratory humanoid biped robot

In this article is used specially developed laboratory model of a humanoid robot. The robot is consists of seventeen degrees of freedom, which are provided with the support of servomotors LD-2015, Arduino Mega board with ATmega2560 microcontroller, two gyroscopes GY-521 with MPU-6050 microprocessor and specially designed distribution board. *Figure 1* shows the dimensions and construction of the developed laboratory model of a humanoid robot. The robot has the following specifications: 365 [mm] height, 215 [mm] width, weight about 1650 [g] and 17 [DoF].

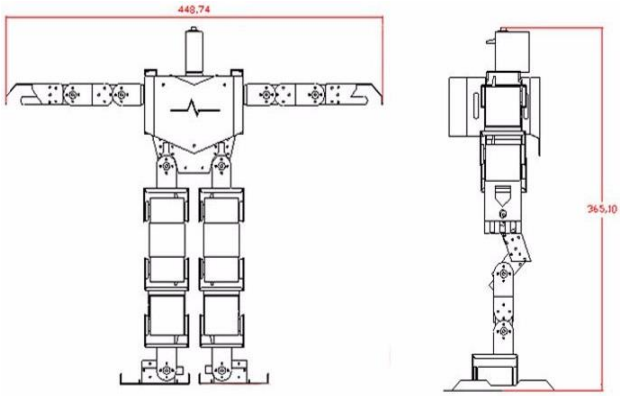


Figure 1. Dimensions of the humanoid

2.1 Motion subsystem

Motion of humanoid is realized with servomotors LD-2015, which is shown in *Figure 2*. These servomotors are chosen because they have high stall torque, high precision and dimensions that are compatible with the construction of the robot. Servomotors have the following specifications:

- Stall torque: 15 [kg/cm] at 6 [V]; 17 [kg/cm] at 7.4 [V];
- Empty load current: 100 [mA];

- Power on: 0.5~2.5 [ms] high level pulse, 0~180 degree change;
- Power off: could rotate 360 degree in manually twist.

Figure 2. Servomotor

2.2 Information system

The Arduino Mega 2560 is board, which based on microcontroller ATmega2560. It has 54 digital input-output pins (15 of which can be used as PWM pins), 16 analog inputs, 4 UARTs (hardware serial ports), 16 [MHz] crystal oscillator, USB connection, power jack and pushbutton reset. *Figure 3* shows the Arduino Mega 2560 board, which is used for the information subsystem of the developed laboratory model of a humanoid robot and more precisely for the control of the motion part, and to obtain information from the gyroscopes. The Arduino Mega 2560 has technical specifications:

- Operating Voltage: 5 [V];
- Input Voltage (recommended): 7-12 [V];
- DC Current per I/O Pin: 40 [mA];
- DC Current for 3.3[V] Pin: 50 [mA];
- Flash Memory: 256 [KB] of which 8 [KB] used by bootloader;
- SRAM / EEPROM: 8 [KB] / 4 [KB].

Figure 3. Arduino Mega 2560

Figure 4 shows a gyroscope that is model GY-521 and is used as a feedback sensor in a humanoid robot. The GY-521 gyroscope has the following technical specifications:

- Embedded chip: MPU-6050;
- Power supply: 3-5 [V] (internal low dropout regulator);
- Gyroscope range: ± 250 | 500 | 1000 | 2000 [°/s];
- Acceleration range: ± 2 | ± 4 | ± 8 | ± 16 [g].

Figure 4. Gyroscope

2.3 Electrical subsystem

A distribution board has been specially designed as addition the Arduino Mega 2560 controller. The board has pins for seventeen servomotors, two gyroscopes pins and one magnetometer pin (not currently used), as well as pin for grounding and external power supply. The dimension of board allows mounting her the following controllers: Arduino Mega 2560, Arduino DUE and Arduino Mega ADK. From *Figure 5* can be seen, that the board arranges pins correctly, which allows easy connection of all hardware to the microcontroller. On *Figure 6* given the assembly scheme showing the distribution of the electrical connections as physical between the two layers of the board and the transitions between layers.

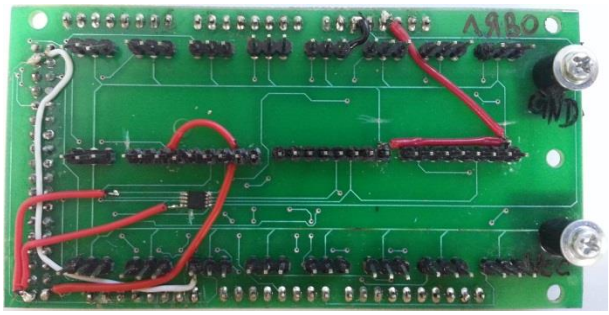


Figure 5. Distribution board

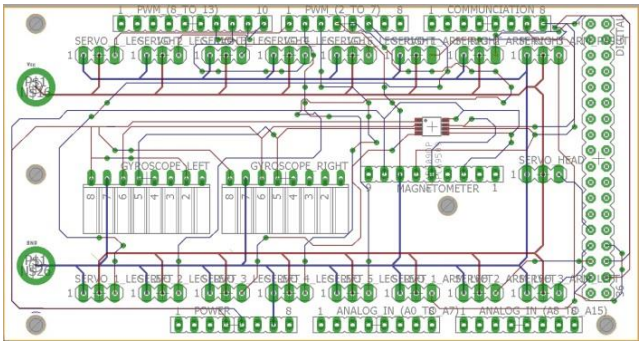


Figure 6. Assembly scheme of distribution board

The distribution board has the following technical specifications:

- Input voltage: up to 7.4 [V];
- Maximum current: 10 [A];
- Dimensions: 53.3 [mm] height, 101.5 [mm] length;
- Weight: 30 [g].

Figure 7 shows location of pins on the integrated circuit (IC). The element enclosed by dashed square in the *figure 8* is an integrated circuit of the PCA9505 type to which the two gyroscopes are connected. *Figure 9* shows the electrical scheme of the board from where can be seen all electrical connections of individual components. From *figure 9* it can be seen that internal structure of the PCA9509 integrated circuit consists of electrical buffers, which are connected in parallel two by two, and individually in opposite direction and by an internal current source capable of gives to 1 [mA].

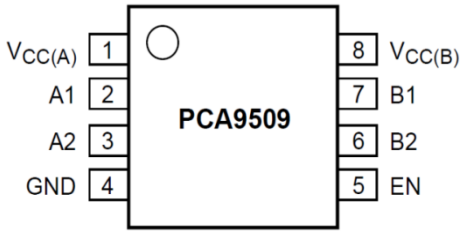


Figure 7. Pins of the IC-PCA9509

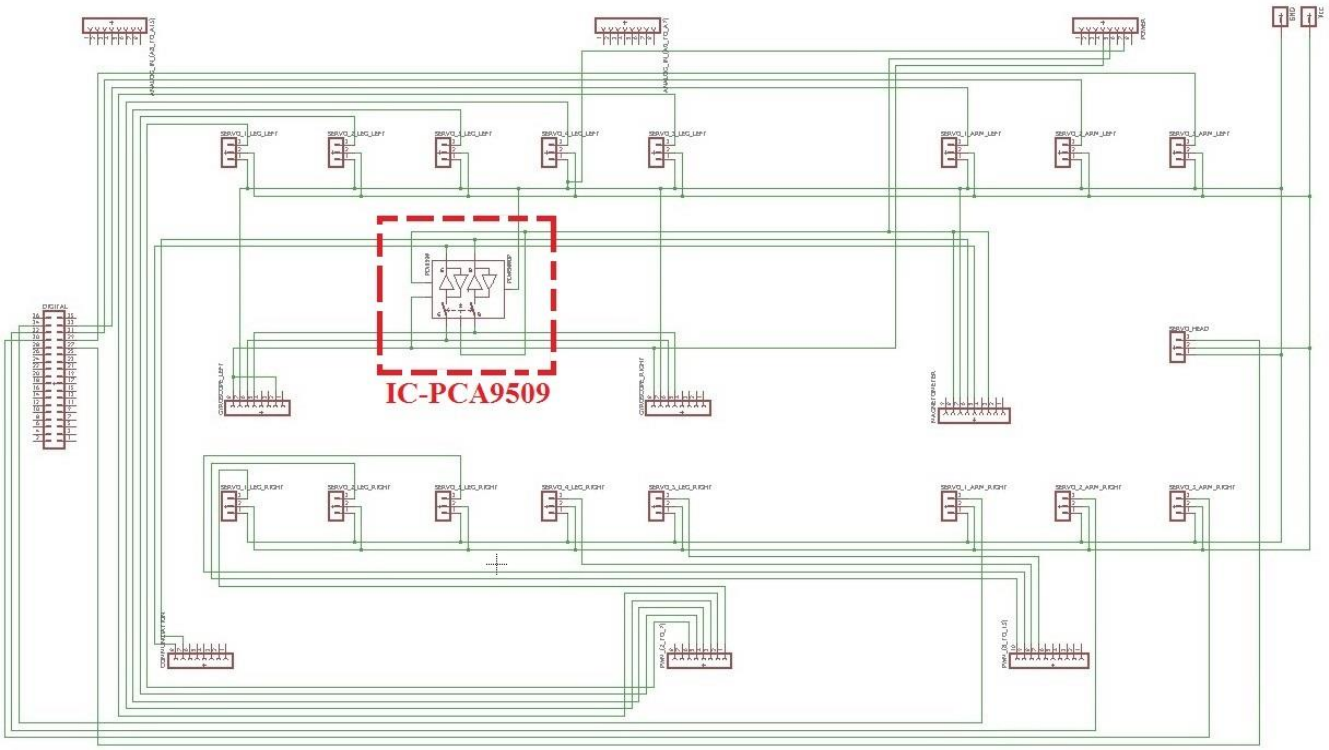


Figure 8. Assembly scheme of distribution board

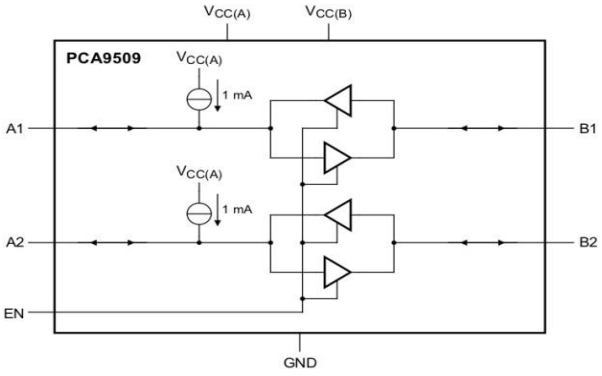


Figure 9. Functional block scheme of IC-PCA9509

The first pin is powered with 3.3 [V] and eighth with 5 [V]. The pins marked with "A1" and "A2" are connected respectively to data line (SDA) and clock line (SCL) on gyroscope, and the sixth and seventh pins on the microcontroller side. The fifth pin is an enable input and is active at a high level, i.e., 5 [V]. This integrated circuit is used to keeping the standard for operation on 3.3 [V] in pins of the two gyroscopes that are for synchronous serial communication (SCL and SDA), because pins for this communication on the Arduino board are 5 [V]. The PCA9509 has a clock generator operating at frequencies from 0 to 400 [kHz].

3. Nonlinear model of the humanoid robot

The nonlinear model was created using SimMechanicsTM (in the new versions of Matlab® is Simscape MultibodyTM) library in Simulink®. The individual block from which the mechanical model is built and their functions [8, 9].

The Machine Environment block (*figure 10*) defines the mechanical simulation environment for the machine to which the block is connected: gravity, dimensionality, analysis mode, constraint solver type, tolerances, linearization and visualization.

A Ground block (*figure 11*) represents an immobile ground point at rest in the absolute inertial World reference frame. Connecting it to a Joint prevents one side of that Joint from moving. You can also connect a Ground block to a Machine Environment block.

Figure 10. Block for machine environment

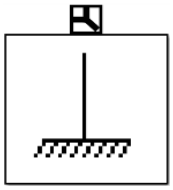


Figure 11. Block for ground

The Six-DoF block (*figure 12*) represents a composite joint with three translational degrees of freedom (DoFs) as three prismatic primitives and three rotational DoFs as one spherical primitives. Represents most general motion of the follower (F) with respect to the base (B) body. Sensor and actuator ports can be added.

The Body block (*figure 13*) represents a user-defined rigid body. Body defined by mass m, inertia tensor I, and coordinate origins and axes for center of gravity (CG) and other user-specified body coordinate systems. The inertia tensor is introduced as a matrix I, which has the following form:

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (1)$$

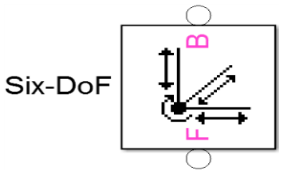


Figure 12. Block for joint with six degrees of freedom

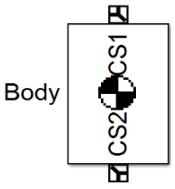


Figure 13. Block for rigid body

The main diagonal, as can be seen from the formula contains the inertial tensors along the axes, and outside the diagonal elements are zero, because the inertial tensor is calculated relative to the center of gravity. *Table 1* shows the formulas for the inertia tensor of a rectangular parallelepiped along the three axes, with respect to its mass and sides.

Table 1. Inertia tensors in rectangular parallelepiped

I_x	I_y	I_z
$\frac{m}{12}(a^2+b^2)$	$\frac{m}{12}(a^2+c^2)$	$\frac{m}{12}(b^2+c^2)$

To calculate the inertia of the servomotor, it will be considered as the equivalent of a parallelepiped shown in *figure 14*.

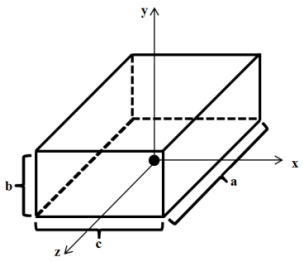


Figure 14. Servomotor equivalent

Figure 15 shows a specially created block used in the mechanical model as a block for earth reaction, and *figure 16* shows its internal structure. Block for Body Sensor measures motion of the body coordinate system to which the sensor is connected. Sensor measures any combination of translational position, velocity, and acceleration; and rotational orientation, angular velocity, and angular acceleration.

The Body Actuator block actuates a Rigid Body block with a generalized force signal, representing a force/torque applied to the body: force for translational motion and torque for rotational motion.

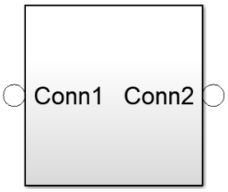


Figure 15. Block for Earth reaction

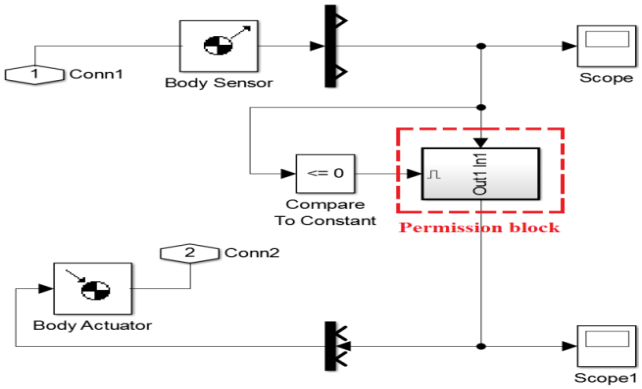


Figure 16. Internal structure of block for earth reaction

As can be seen from the internal structure of the permission block (*figure 17*), it consists of a P-regulator. This subsystem is executed only when we have a unit of the Enable block, i.e., when foots of humanoid have contact with the ground.

Block for Revolute Joint (*figure 18*) represents one rotational degree of freedom. The follower (F) body rotates relative to the base (B) body about a single rotational axis going through collocated body coordinate system origins.

Figure 19 shows block for servomotor used in the mechanical model of the humanoid. The set point for the servomotor is set via the constant block. *Figure 20* shows a model of a servomotor that was created in the Simulink®. The model, like the real object, has two contours, one in speed and one in position, which is regulated by P-regulators. Block for Joint Actuator (in *figure 20*) – actuates a joint primitive with generalized force/torque or linear/angular position, velocity, and acceleration motion signals. Base-follower sequence and joint axis determines sign of forward motion.

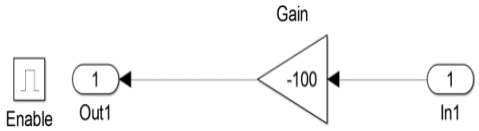


Figure 17. Internal structure of the permission block

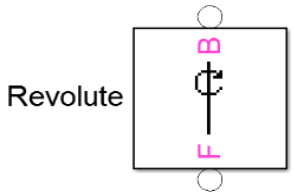


Figure 18. Block for revolute joint

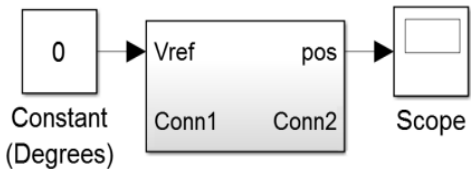


Figure 19. Block for servomotor

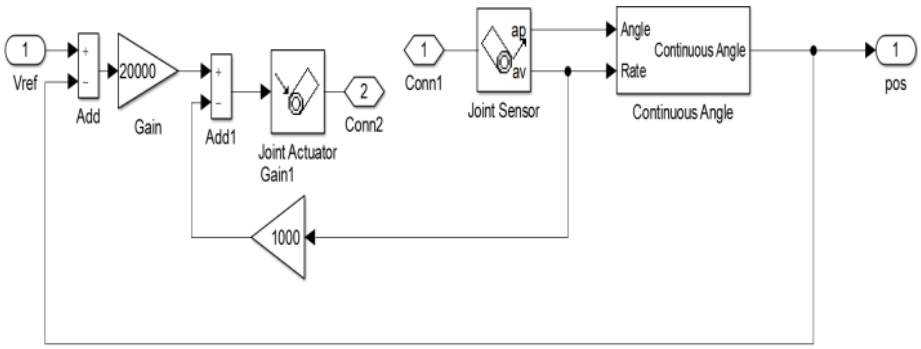


Figure 20. Assembly scheme of distribution board

Joint Sensor block measures linear/angular position, velocity, acceleration, computed force/torque and/or reaction force/torque of a joint primitive.

Figure 21 shows the mechanical model created through the SimMechanicsTM library in Simulink® of laboratory model of a humanoid robot. The individual blocks from which the mechanical model is created and their functions are given in *figures 10* to *20*.

Figure 22 is given the geometric model of the humanoid, which is visualized by its mechanical model, and

through the block for machine environment are set the parameters of the visualization. There is accordance between laboratory model of the humanoid and geometric model in terms of geometric parameters, which are dimensions along the three axes, and the physical parameter, the center of gravity.

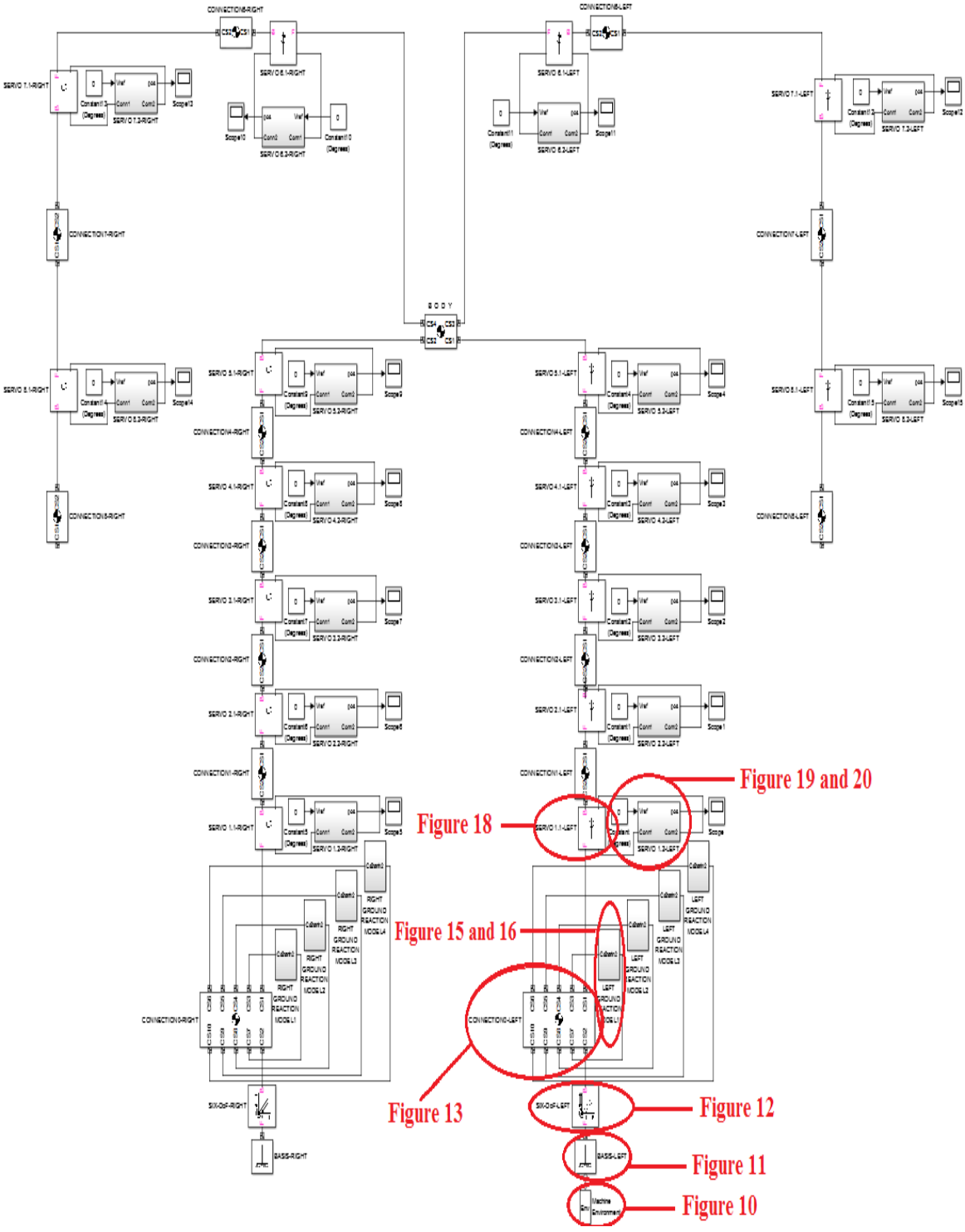


Figure 21. Mechanical model of the humanoid

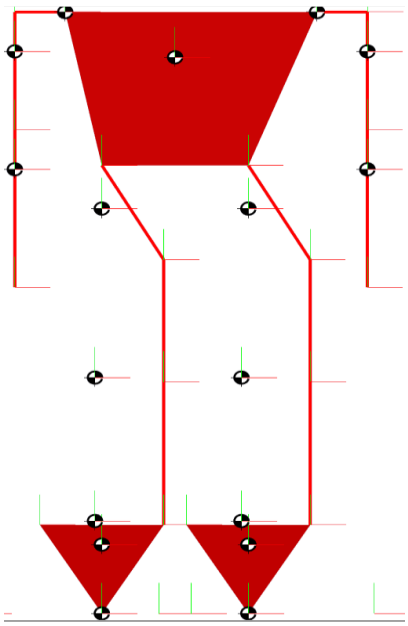


Figure 22. Geometric model of the humanoid

4. Simulation results from the nonlinear model of the humanoid robot

The two graphs (*figures 23 and 24*) show that there is a process of interaction with the earth's surface. The earth's surface is modeled as a system of springs and a logical element, due to which there are ups and downs in the characteristics. When the footsteps on the earth's surface of the graphs we have an increase and the logical element is one, and when it separates from the earth's surface these rises stop and a decrease in the characteristics begins and accordingly the logical element is zero.

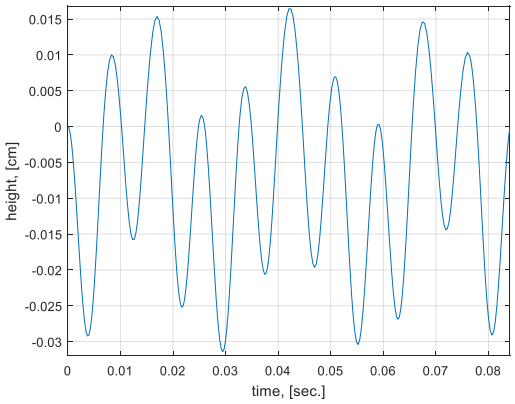


Figure 23. Vertical position of the leg

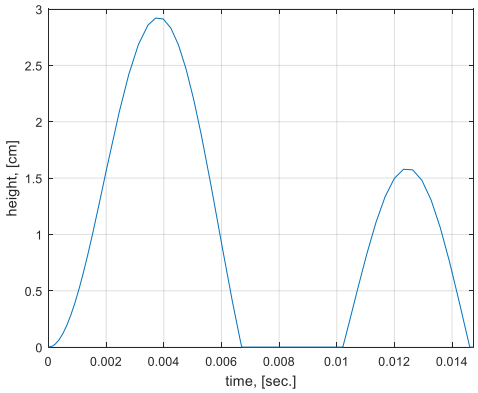


Figure 24. Reaction force on the foothold

As can be seen from *figure 25* step responses between the servomotor and its created model overlap, which is proof that the model is reliable.

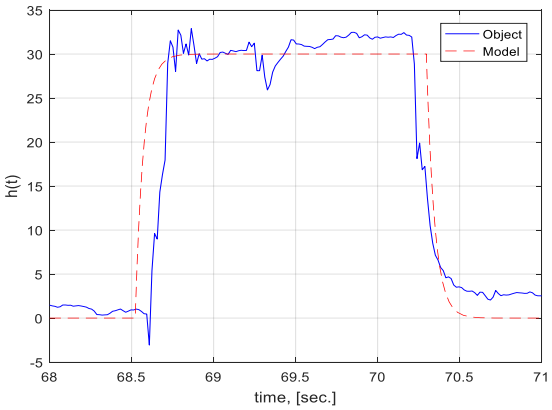


Figure 25. Step responses of the servomotor

The *figures 26* and *27* show simulation results made on the foot of the humanoid, and the *figures 28* and *29* – on the hand. From the step responses, which are from the model of servomotor given in *figure 30*, it can be seen that they reach settling time in about 200 milliseconds. From settling time, it can be concluded that we have little rise time. Also step responses have no overshoot and ringing.

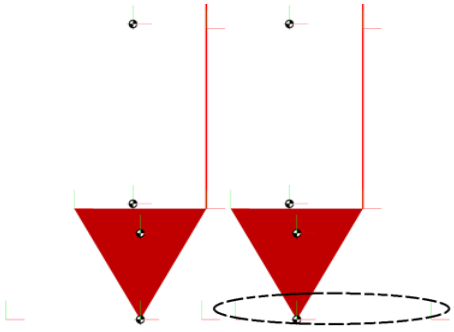


Figure 26. Step responses of the servomotor

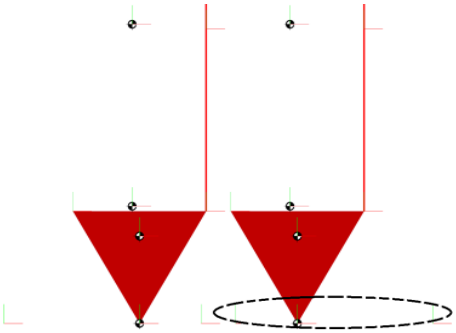


Figure 27. Step responses of the servomotor

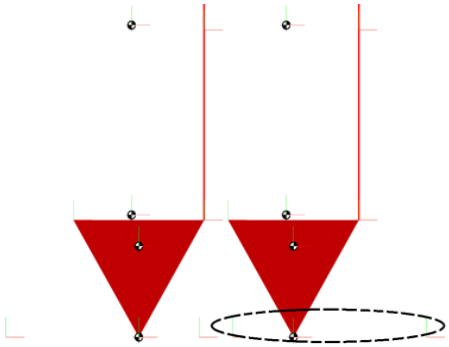


Figure 28. Step responses of the servomotor

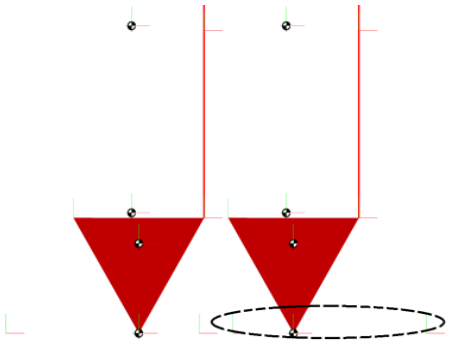


Figure 29. Step responses of the servomotor

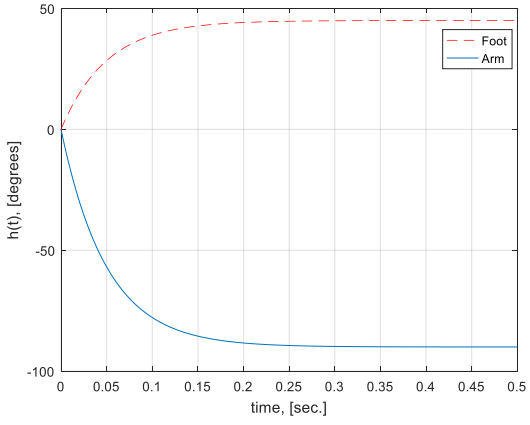


Figure 30. Output signals from the servomotor model

5. External disturbances

The aim of the open-loop control for biped walking robots is to attenuate external disturbances acting on the mechanical system. In order to achieve such control the external disturbances have to be from deterministic origin. For the present laboratory model of humanoid robot the key external disturbances are: gravitational force, friction, inertial forces and center of gravity motion.

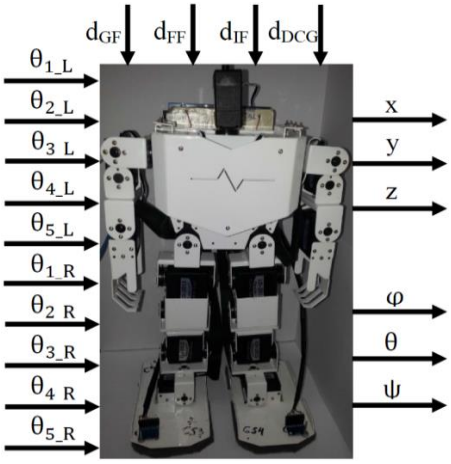


Figure 31. Scheme for implementation of the control algorithm

The control actions are $\boldsymbol{Q_{i,L}}$ and $\boldsymbol{Q_{i,R}}$ are the angular offset in the left and right key joints, with $i = 1 \div 5$. The output signals $\boldsymbol{x, y, z}$ represent the position for the robot CG and Euler angles $\boldsymbol{\varphi, \theta, \psi}$ represent the robot orientation

- **Gravitational force - $\mathbf{d_{GF}}$**

The gravitation is disturbing the stability on the robot when the incidence angle between the body and the ground surface is increased. If that angle goes beyond certain limit the robot cannot be stabilized with the control action from the servo motors alone.

- **Friction - $\mathbf{d_{FF}}$**

Successful walking requires a certain amount of contact force between the ground surface and robot feet which prevents foot slipping. However when a foot is translated over the surface developed friction force can disturb the orientation of the robot and consequently its trajectory.

- **Inertial forces - $\mathbf{d_{IF}}$**

These forces are fictitious forces representing the effect of kinematic variables upon the orientation of the body. If the frequency of the step cycle is increased beyond certain limit the abundant energy from uncompensated inertial forces cannot be dampen fast enough and leads to instability.

- **Center of gravity motion - $\mathbf{d_{DCG}}$**

Generally if the center of gravity is not projected upon the feet base then the mechanical construction becomes less stable and small uncompensated disturbance force would cause the robot to fall down.

The location of the robot CG is the most important factor which prevents the robot from falling during the execution of the walking step cycle. Location of the CG can be expressed with (2) by accounting for the mass action of the various components of the robot:

$$\vec{X}_{CG} = \frac{m_B}{M}\vec{X}_{CG,B} + \frac{m_{HL}}{M}\vec{X}_{CG,HL} + \frac{m_{HR}}{M}\vec{X}_{CG,HR} + \frac{m_{LL}}{M}\vec{X}_{CG,LL} + \frac{m_{LR}}{M}\vec{X}_{CG,LR}, \quad (2)$$

where $\vec{X}_{CG}$ is the location vector of the CG with components

$$\vec{X}_{CG} = \begin{pmatrix} x_{CG} \\ y_{CG} \\ z_{CG} \end{pmatrix} \quad (3)$$

and $\vec{X}_{CG,i}$ is the location of CG of respective body element $i \in \{B, HL, HR, LL, LR\}$, M is the total mass of the robot, m_B is the mass of the trunk, m_{HL} is the left hand mass, m_{HR} is the right hand mass, m_{LL} is the left leg mass and m_{LR} is the right leg mass such that:

$$M = m_B + m_{HL} + m_{HR} + m_{LL} + m_{LR}. \quad (4)$$

The CG of each component of the robot at time instant k is shifted with respect to CG location at time instant $k-1$

$$\vec{X}_{CG,i}(k) = \vec{X}_{CG,i}(k-1) + \vec{\Delta}_i \quad (5)$$

with $\vec{\Delta}_i$ representing the CG displacement caused by joint reconfiguration during walking. Then the displacement of the robot CG $\vec{\Delta}_{CG}$ will be a weighted sum over the displacements $\vec{\Delta}_i$ of CGs of its components with weights m_i/M. Hence by rotating a joint in particular direction the robot CG is translated in this direction. In order to keep the upper stance the projected location of the robot CG onto the ground surface should be kept within the supporting foot base.

6. Walking algorithm

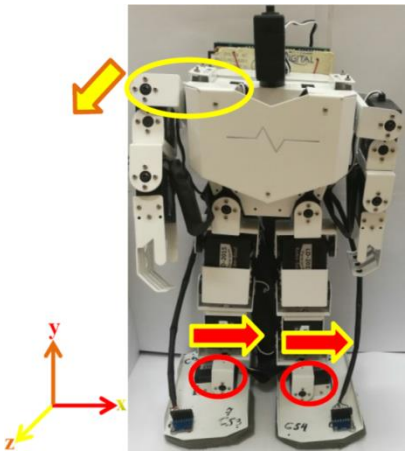
Walking algorithm on humanoid is implemented around the operating points of the ten servomotors used in the joint coordination sequence scheme [10]. At initial position the robot joint angles are set to their trim values. They are experimentally tuned to guarantee its upper standing position and are summarized in *table 2*.

Table 2. Absolute values of servomotors

Joint	Value [degrees]
Left ankle roll	97
Left ankle pitch	74
Left knee pitch	6
Left hip pitch	85
Left shoulder pitch	155
Right ankle roll	79
Right ankle pitch	82
Right knee pitch	142
Right hip pitch	70
Right shoulder pitch	87

The joint coordination sequence consists of thirteen steps in order for the humanoid to move forward without falling. The starting position of the walking robot is denoted as "A" when its center of gravity is distributed on both legs as in *figure 31*.

In stage "B" as *figure 32* shows the center of gravity projection is shifted to the left leg by applying ankle roll and right shoulder pitch.



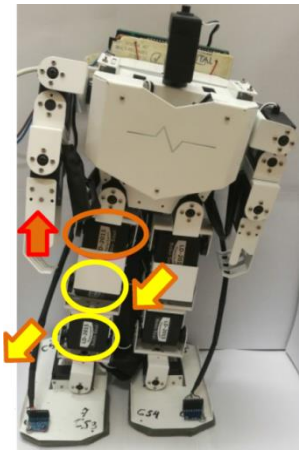

```
if (Step_State > 0.83 && Step_State < 5)
{
 Offset_1_L = -7;
 Offset_1_R = -7;
 Offset_5_R = +30;
}
```

Figure 32. Stage "B"

In stage "C" (*figure 33*) the right leg is moved forward but the center of gravity projection remains on the left foot and keeps the stability of the robot. In this stage the right leg pitch is performed in ankle, knee and hip joints.

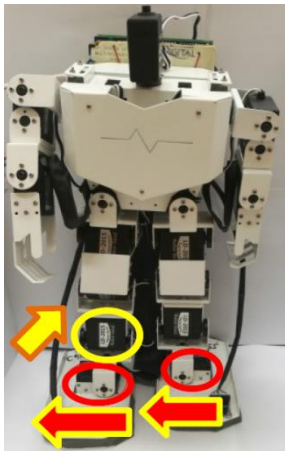
In stage "D" given in *figure 34* it can be noticed that the right leg is exerted a little further from the left, which is due to the changes in the previous stage (stage "C" – *figure 33*). Also in stage "D" center of gravity projection is again distributed on both legs. To accomplish this step, three

changes are made to the current values of the servomotors – roll of both ankles and pitch of the right ankle.



```
else if (Step_State > 7.08 && Step_State <
19.56)
{
 Offset_2_R = -10;
 Offset_3_R = -20;
 Offset_4_R = -20;
}
```

Figure 33. Stage "C"

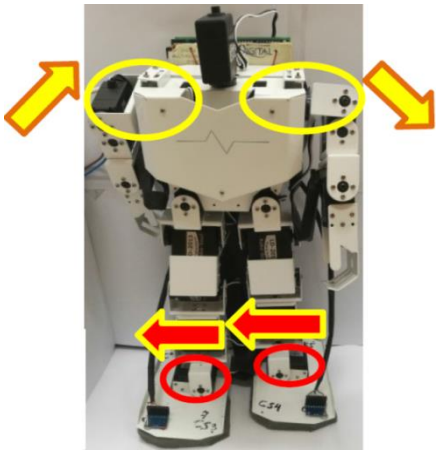


```
else if (Step_State > 21.67 && Step_State
< 25.83)
{
 Offset_1_L = 0;
 Offset_1_R = 0;
 Offset_2_R = 0;
}
```

Figure 34. Stage "D"

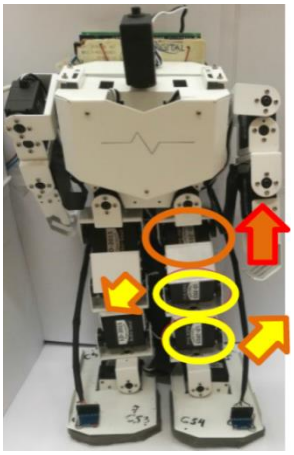
In the next stage "E" (*figure 35*) the center of gravity projection is shifted to the right foot by rotating four of the joints – ankle's roll and shoulder's pitch.

During the sixth stage "F" shown in *figure 36* the left foot of the robot is moved forward by performing ankle, knee and hip pitch. The center of gravity projection then lies on a right leg which is static to ensure the stability of the robot. The difference between this stage compared to the stage "C" is in supporting leg where the center of gravity projection is located.



```
else if (Step_State > 27.92 && Step_State
< 32.08)
{
 Offset_1_L = +7;
 Offset_5_L = -30;
 Offset_1_R = +7;
 Offset_5_R = -30;
}
```

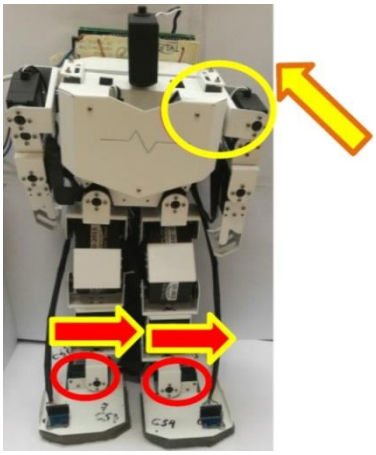
Figure 35. Stage "E"



```
else if (Step_State > 34.17 && Step_State
< 46.67)
{
 Offset_2_L = -10;
 Offset_3_L = +40;
 Offset_4_L = +40;
}
```

Figure 36. Stage "F"

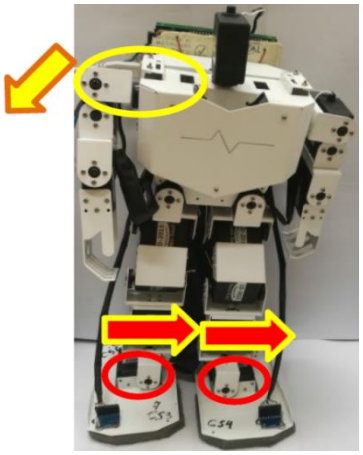
In stage "G" (*figure 37*) the weight of the robot is distributed over the both legs, but it is not in the starting position because the right knee joint, the right shoulder joint and some of the left leg joints are still displaced. To achieve this stage roll of both ankles is applied and also a pitch of the left shoulder.



```
else if (Step_State > 48.75 && Step_State
< 52.92)
{
 Offset_1_L = 0;
 Offset_5_L = 0;
 Offset_1_R = 0;
}
```

Figure 37. Stage "G"

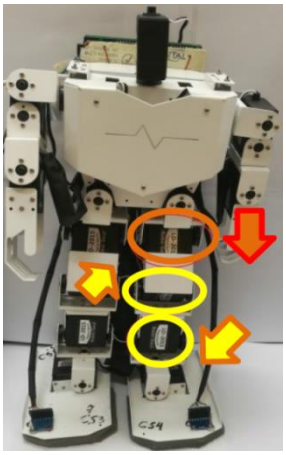
The stage "H" from *figure 38* shifts the center of gravity projection to the left leg. The difference here with respect to stage "B" is that the two servos in the knee of the right leg are not in their initial position and also the servo in the shoulder joint of the right hand. The other difference is in the left leg and in particular in the servomotor for ankle pitch and the two servomotors in the knee, which in the current stage ("H") are not in their initial position.



```
else if (Step_State > 55 && Step_State <
59.17)
{
 Offset_1_L = -7;
 Offset_1_R = -7;
 Offset_5_R = +30;
}
```

Figure 38. Stage "H"

In stage "J" (*figure 39*) the left foot of the humanoid begins its return to its starting position. To prevent the robot from falling this is done in two stages ("J" and "K"). The need for two steps is due to the fact that the current angle of the left knee joint is twice as large as that of the right knee. The difference in this stage with respect to "F" stage is the foot on which supports the center of gravity projection. In order to accomplish this step, three changes were made to the current values of the joints. The ankle and knee servos are still not returned to their starting position. Their current angels are set to an intermediate values with respect to stages "F" and "K".

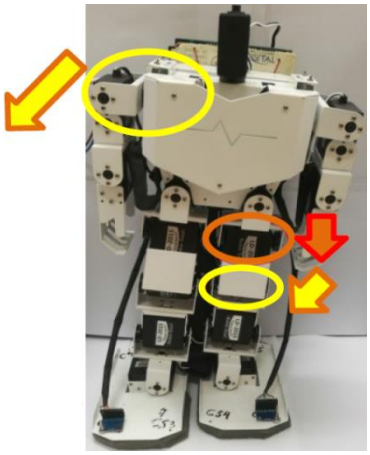


```
else if (Step_State > 61.25 && Step_State
< 65.42)
{
 Offset_2_L = 0;
 Offset_3_L = +20;
 Offset_4_L = +20;
}
```

Figure 39. Stage "J"

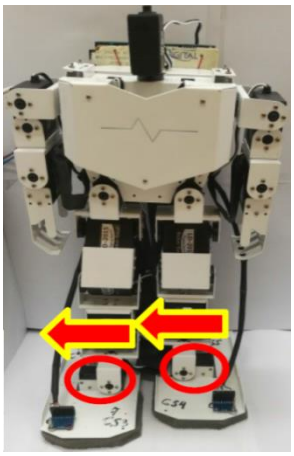
In the tenth stage "K" (*figure 40*) the process of returning of the ankle and knee joints of the left leg to initial position is completed. Also, at this stage, the right shoulder joint is set to its trim value.

In stega "L" shown in *figure 41* the center of gravity projection lies on both robot's legs, but it is still not in the starting position because the right foot is exerted slightly further than the left. The right leg is ahead of the left due to its ankle and knee joints. In order to realize this stage, there are changes in the current values of two servomotors, which are located in the feet joints.



```
else if (Step_State > 67.5 && Step_State <
71.92)
{
 Offset_3_L = 0;
 Offset_4_L = 0;
 Offset_5_R = 0;
}
```

Figure 40. Stage "K"

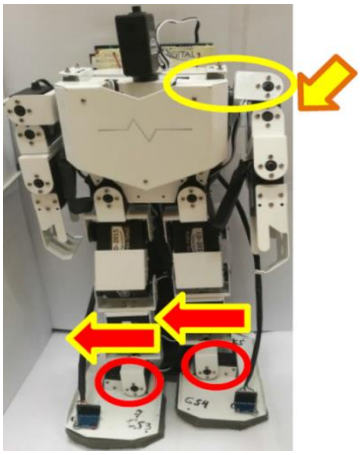


```
else if (Step_State > 73.75 && Step_State
< 77.92)
{
 Offset_1_L = 0;
 Offset_1_R = 0;
}
```

Figure 41. Stage "L"

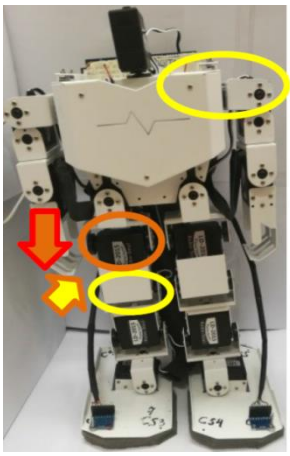
In stage "M" (*figure 42*) the center of gravity projection is shifted to the right leg. In order to accomplish this step, changes are made to the current values of three servomotors pointed in the figure.

In next stage "N" (*figure 43*) the robot's right foot returns to the starting position. For the completion of this stage right hip and knee joint are rotated. Also the left shoulder joint is rotated. After the rotations the servomotors in the knee of the right leg and the servomotor in the left shoulder are set to their trim values.



```
else if (Step_State > 80 && Step_State <
84.17)
{
 Offset_1_L = +7;
 Offset_5_L = -30;
 Offset_1_R = +7;
}
```

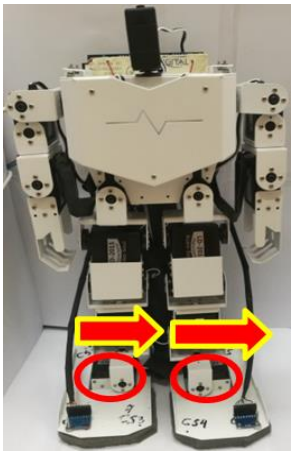
Figure 42. Stage "M"



```
else if (Step_State > 86.25 && Step_State
< 90.42)
{
 Offset_5_L = 0;
 Offset_3_R = 0;
 Offset_4_R = 0;
}
```

Figure 43. Stage "N"

A sub-stage before returning to stage "A" after the last stage "N" – stage "A = 0" (*figure 44*) is executed in order to return all joints to their trim values. In this sub-step, the weight of the robot is distributed over both legs. Then the robot completes the joint coordination sequence and its location is one stage ahead.



```
else if (Step_State > 92.92 && Step_State
< 97.08)
{
 Offset_1_L = 0;
 Offset_1_R = 0;
}
```

Figure 44. Stage "A = 0"

7. System identification with parametric models

The identification process is divided in two stages. In the first stage an initial model is estimated and in the second stage the initial model is improved through iterative optimization procedure in order to obtain a final model of the robot.

The offsets around the equilibrium for the joints are obtained with the following expression:

$$\Delta\theta_{ij} = \left(\theta_{C,ij} - \theta_{TRIM,ij}\right) \tag{6}$$

where $\Delta\boldsymbol{\theta_{ij}}$ represents the offset of the angle in the servomotor; $\boldsymbol{\theta_{C,ij}}$ is the current angle and $\boldsymbol{\theta_{TRIM,ij}}$ contain the absolute value of the angle. The absolute values of the angles of the servomotors are given in *table 2*. Depending on the indices ***i*** and ***j*** respective servomotor is addressed. The index ***i*** shows in which part of the robot body the observed motor is located (in the hands or in the feet), and ***j*** shows the side (left of right).

The test signals used for system identification are shown in *figure 45* and *figure 46*. The data contains 1000 points, and the start conditions are selected to be zeros.

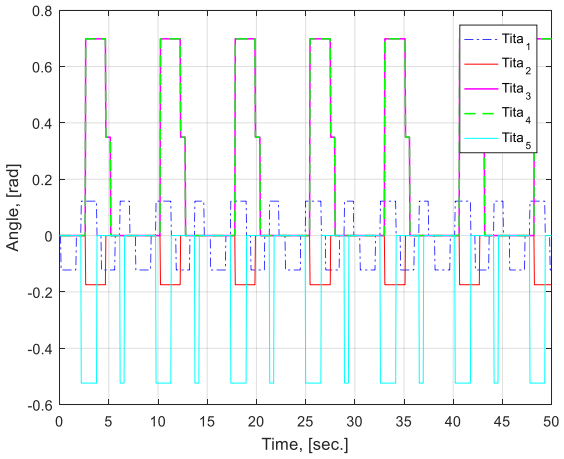


Figure 45. Offsets of the servomotors in the left side

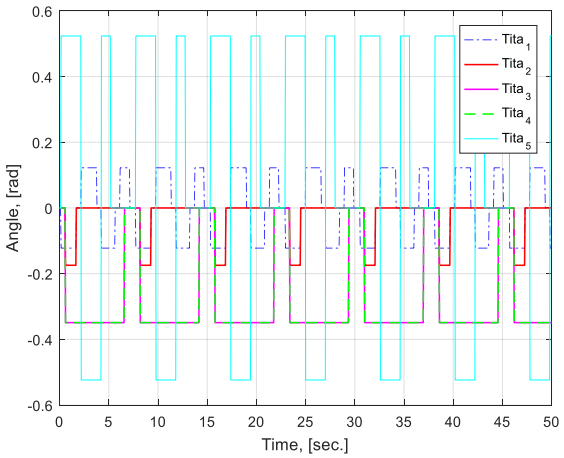


Figure 46. Offsets of the servomotors in the right side

Expression (7)

$$\varepsilon_i = \varepsilon_i^*[LSB] = \frac{\varepsilon_i^*}{131}[°/sec] = \frac{\varepsilon_i^*}{131}\frac{\pi}{180}[rad/sec] \tag{7}$$

defines the scaling method used for data preprocessing where $\boldsymbol{\varepsilon_i}$ the measured angular velocities are and $\boldsymbol{\varepsilon_i^*}$ are the scaling factors. These variables store the raw information measured from the gyroscopes. The scaling factor for converting the data from raw data to degrees per second equals to 131. The index ***i*** could be replaced with ***x, y, z*** in relation to the coordinates. *Figure 47* and *figure 48* show the measured angular velocities. It can be observed that they are periodic and synchronized with the movement of the robot represented by the input offsets. The maximal and minimum amplitudes in the measured angular velocities are related to the oscillations around the equilibrium state of the robot when it is moving forward.

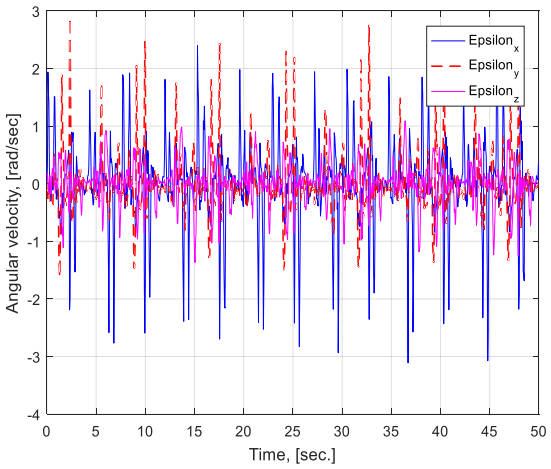


Figure 47. Angular velocities measured with the left gyroscope sensor

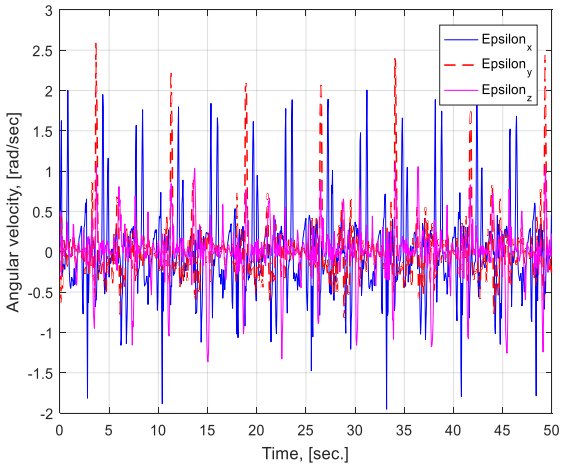


Figure 48. Angular velocities measured with the right gyroscope sensor

The data from the gyroscopes are obtained at the same time, because they have a different address – the one in the left leg has the address *0x68* (default address) and in the other leg the address of the sensor is *0x69*. The change of the gyroscope address of the right leg is done by a jumper between the address pin and power pin.

A more detailed view on the identification data set is presented on *figure 49*. There the first row shows the angular velocities measured by the gyroscopes and the second row contains the amplitudes of the input signals (the offsets around the equilibrium state for each servomotor). The form of the input signals is designed to guarantee stable walking of the robot. They correspond to stages of the step cycle which are tuned to guarantee smooth walking motion.

7.1 Initial model

The initial model is a multiple input multiple output (MIMO) autoregressive exogenous (ARX) model of 24th order which is estimated with the standard least squares method [11, 12]. The following expression defines the structure of the initial ARX model:

$$y_{MxN}(z) = W_{yy}(z)y_{MxN}(z) + W_{yu}(z)u_{MxN} + W_{ye}(z)e_{MxN} \quad (8)$$

where $\boldsymbol{y_{MxN}(z)}$ contains the measured output values. These output signals are the angular velocities vectors of the left and right feet. The vector $\boldsymbol{u_{MxN}(z)}$ represent the current input values, which are the offsets around the equilibrium trim state of the joints. We only use ten servo motors for the forward movement of the robot. The signals $\boldsymbol{e_{MxN}(z)}$ represent the residuals in the model. These error terms are the absolute difference between the measured angular velocity from the humanoid and the angular velocity produced by the model calculation. $\boldsymbol{W_{yy}(z)}$ is symmetric transfer matrix with zeros on the main diagonal which defines the mutual influence between the outputs. Each term is a fraction with denominator autoregressive polynomial of the respective output and the numerator is polynomial describing the cross-correlation. $\boldsymbol{W_{yu}(z)}$ contains information about the dynamic influence of inputs upon the outputs of the system. The elements in this matrix are fractions with denominator respective output autoregressive polynomial and an input-output polynomial. The matrix $\boldsymbol{W_{ye}(z)}$ is diagonal which elements are the typical ARX noise filters.

7.2 Final model

The final model is a state-space model (9), estimated by the prediction error method from the initial MIMO ARX models from the 24th order [13]. The estimation is achieved with the *ssest* command in MATLAB®:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ge(k) \\ y(k) &= Cx(k) + e(k) \end{aligned} \quad (9)$$

where $\boldsymbol{x(k+1)}$ is the state vector at the instant **k+1**, $\boldsymbol{y(k)}$ is the measured output value in the current moment, $\boldsymbol{u(k)}$ is the input vector signal and $\boldsymbol{e(k)}$ is the residual error. Matrix $\boldsymbol{A}$ shows the relationship between states at the current moment and the states at the next moment. Matrix $\boldsymbol{B}$ gives the links between the input signals at the current moment and the states at the next moment. Matrix $\boldsymbol{C}$ shows the relationship between the states at the current moment and the measurable output signals, which are the angular velocities vector of the feet. The system is linearized by the identification procedure in the neighborhood of the joint angles equilibrium.

In the upper part of *figure 49* is the roll, pitch and yaw rates of the left foot followed by the rates of the right foot. At the bottom of the figure are the joint actuator control signals. First five correspond to the left leg and the last five to the right leg. Each leg has two ankles DOF (roll and pitch), one knee DOF (pitch) and two hips DOF (roll and pitch).

First three subplots on *figure 50* correspond to the roll, pitch and yaw angular rates of the left foot and the bottom three subplots correspond to the right foot. All amplitudes are measured in degrees.

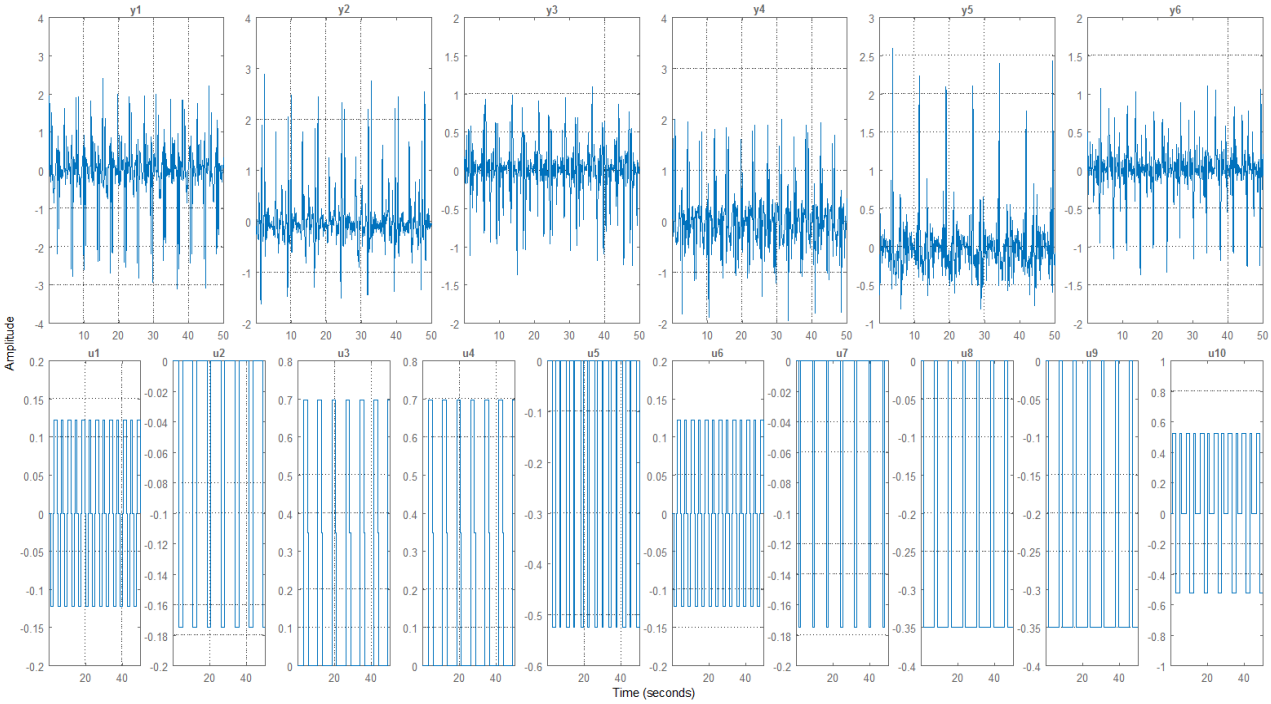


Figure 49. Identification dataset for humanoid robot

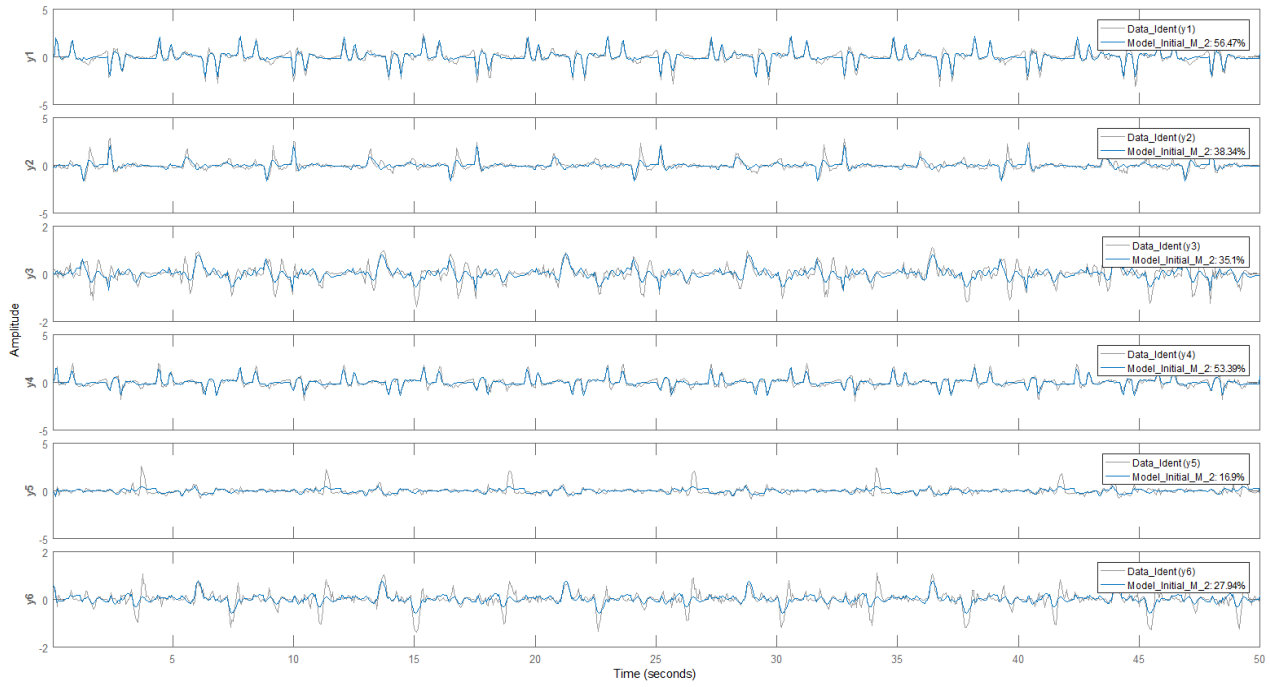


Figure 50. Comparison between output signals from the initial model with the experimental data

8. Evaluation of the models

8.1 Initial model

Comparison between the measured angular rates $\boldsymbol{y}$ and the simulated outputs $\boldsymbol{y_{sim}}$ of the model is shown in *figure 50*. The fit for the roll rate of the left foot is 56 [%] and of the right foot is 53 [%]. The fit for the pitch rate of the left foot is 38 [%] and of the right foot is 17 [%]. Finally the fit for the yaw rate of the left foot is 35 [%] and of the right foot is 28 [%]. These levels are relatively low despite of the obvious similarities between signals due to the length of the data-set. Also interesting to observe is that the fit for the left foot is close in value to the foot of the right foot. The level of fit between them is calculated as:

$$(10) \quad FIT = \frac{\|y - y_{sim}\|_2}{\|y\|_2} \times 100$$

Figure 51 presents the Henkel singular values corresponding to the relative importance of the states determined from the controllability and observability Gramians. This figure indicates that the order of the identified model cannot be reduced considerably. The relative importance of the increasing order decays exponentially which testifies for the infinite dimension of the system.

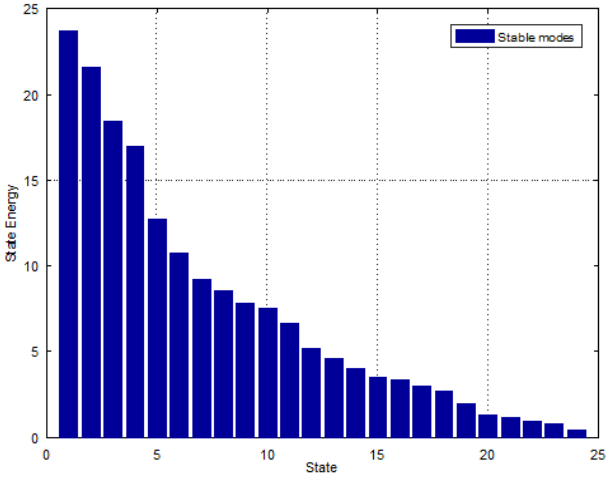


Figure 51. Henkel singular values of the initial model

Figure 52 shows frequency response of the singular values of the initial model. They are calculated from the model transfer matrix. There can be observed that the bandwidth of the identified model is 54.8 [rad/s] and the maximal gain of the system is around 30.1 [dB]. The step response of the open loop will be close to aperiodic with some oscillations around the steady state. Since the number of step responses is 60 they are not included in the present paper but their examination agrees with the observed transients of the robot during walking.

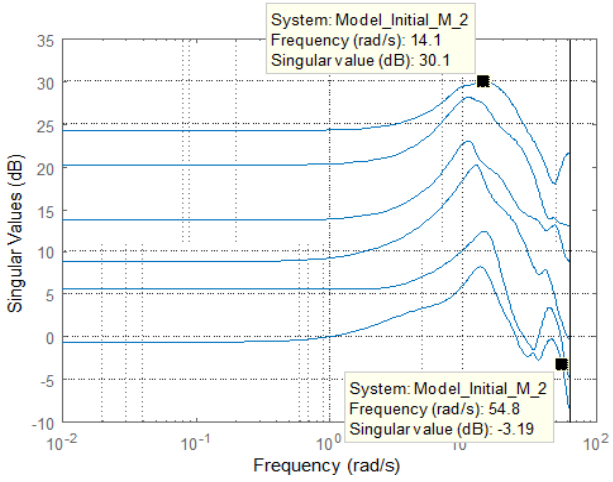


Figure 52. Henkel singular values of the initial model

The initial model is also validated with the residual test which calculates the six autocorrelation functions between the residuals in the model:

$$(11) \quad R_{xx}(\tau) = E[e_i(t)e_i(t-\tau)]$$

The twenty-seven cross-correlation functions between the residual corresponding to different out-puts are also calculated according to:

$$(12) \quad C_{xx}(\tau) = E\left[e_i(t)e_j(t-\tau)\right]$$

These correlation measures should be zero if all information from the data is properly captured in the model (unbiased estimates).

8.2 Final model

The comparison between measured and simulated outputs of the final model, which is updated version of the initial MIMO ARX model, are shown on *figure 53*. The fit for the roll rate of the left foot is 61 [%] and of the right foot is 59 [%]. The fit for the pitch rate of the left foot is 48 [%] and of the right foot is 42 [%]. Finally the fit for the yaw rate of the left foot is 49 [%] and of the right foot is 55 [%]. All levels of fit are increased in relation to the initial model. The largest increase is for the pitch rate of the right foot. This proves that the selected structure from 24th order is rich enough to capture the walking motion of the robot and could be further improved by prediction error optimization. Due to the nonlinearity of the optimization criteria an initial model guess was needed.

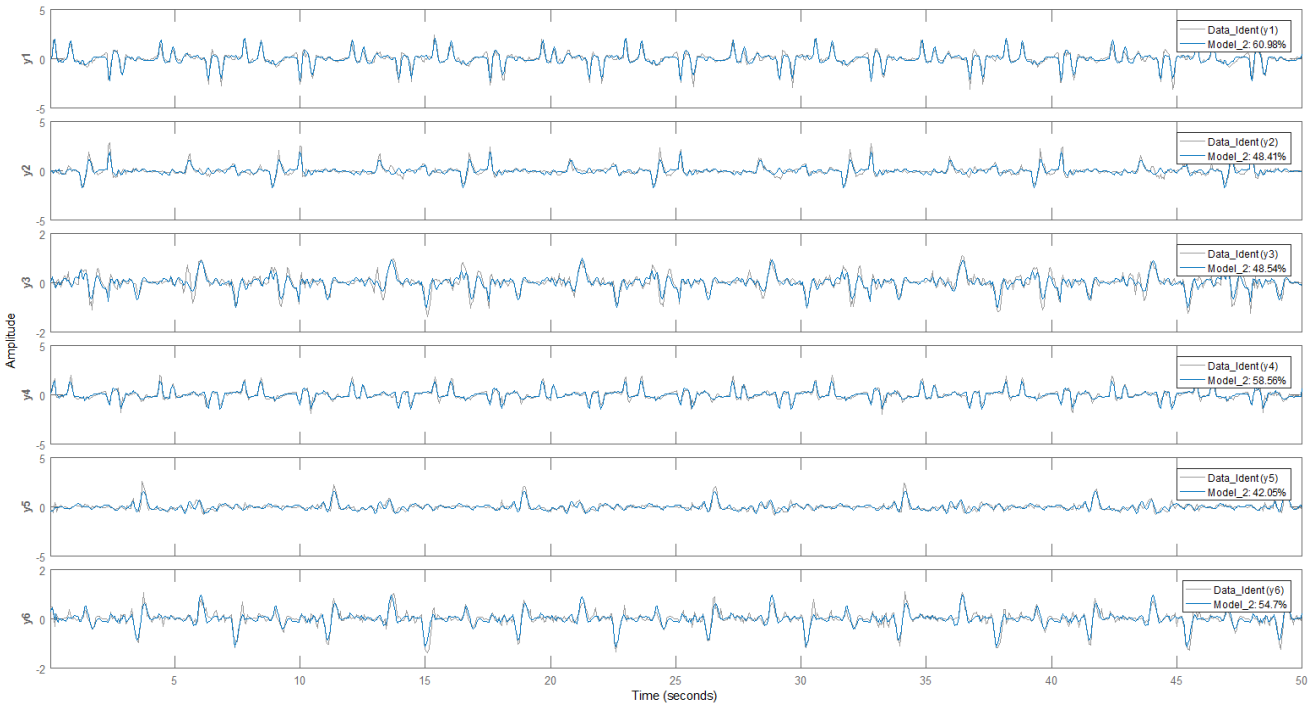


Figure 53. Comparison between output signals from the final model with the experimental data

Henkel singular values of the final model are shown on *figure 54*. It is easy to be seen that the plant is well described with 24th order model which cannot be reduced without degradation of the information content in the model.

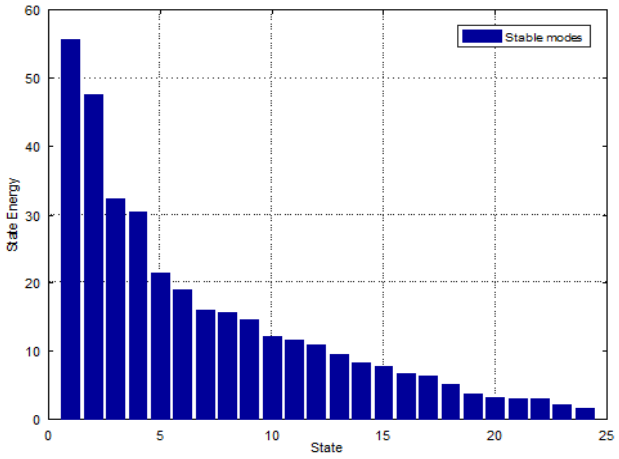


Figure 54. Henkel singular values of the final model

Figure 55 shows the singular values of the model transfer matrix which indicates that the bandwidth is around 20 [rad/s] and the largest magnitude gain is 38 [dB]. In comparison to the initial model the bandwidth of the final model is increased together with the maximal gain of the magnitude response.

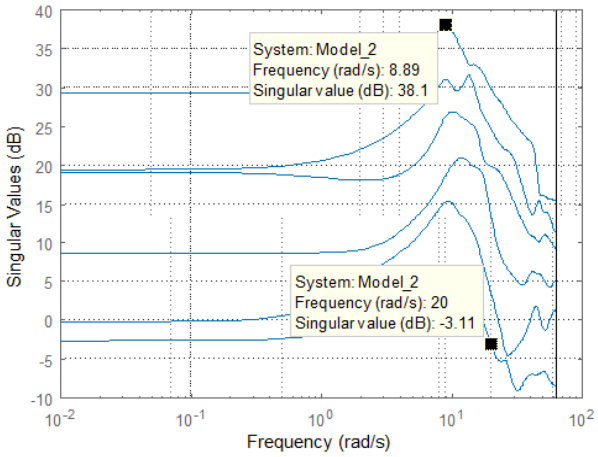


Figure 55. Final model singular values

The step responses of the final model are similar to the ones produced by the initial model. The covariance of the estimated model parameters is small enough and leads to small confidence regions for the model characteristics. This can be attributed to the large enough signal to noise ration of the identification data-set and the achieved fit by the model. Also the residual test of the final model with an independent validation data-set asserts that the identified model is unbiased. Therefore the identified final model is appropriate for the design of feedback control algorithms. *Table 3* summarizes the predicted and least square errors achieved by

the initial and the final model. As can be seen the both error measures are reduced after the optimization of the initial model.

Table 3. Performance of the estimated models

Initial Model		Final Model	
Predicted Error	Least Square Error	Predicted Error	Least Square Error
9.484×10^{-7}	6.509×10^{-1}	4.268×10^{-7}	6.056×10^{-1}

9. Conclusion

In this article research a specially developed laboratory model of a biped humanoid robot, which consists of seventeen degrees of freedom. Detailed information is given for the mechanical, motion, information and electrical subsystem. The authors are designed distribution board intended specifically for this humanoid robot. A nonlinear model was created through SimMechanicsTM library in Simulink®, which requires only geometric and physical parameters, which are entered through the standard blocks from this library, as well as through the blocks construct by the authors. The simulation results obtained from the nonlinear model are also shown. The four external disturbances that have the greatest effect on open loop control are described, and advice is given on how they can be overcome. The walking algorithm on robot has been developed, so that it provides stability when walking. The algorithm is implemented around operating points of the servomotors, which are used in joint coordination sequence scheme, which consists from thirteen steps. All stages of the scheme are described in detail and shown through schematic photos. The identification is performed in two main stages. The first step is to create an MIMO autoregressive exogenous ARX model using the standard least squares method. The initial model is estimated by the prediction error method to obtain the final model, which is in state-space. Finally, the two models are evaluated according to quantitative and qualitative criteria to prove the need for an initial and final model. As can be seen from the identification results, the final model has much better parameters than initial.

Acknowledgements

The present research was received seed funding by the Research Sector of Technical University – Sofia under contract 202PD0013-08.

References

1. Plestan, F., Grizzle, J. W., Westervelt, E. R., et al. Stable Walking of a 7-DOF Biped Robot. – *IEEE Transactions on Robotics and Automation*, 19, 2003, No. 4, doi: 10.1109/TRA.2003.814514.
2. Gienger, M., Loffler, K., Pfeiffer, F. Towards the Design of a Biped Jogging Robot. *IEEE International Conference on Robotics and Automation*, 2001 Seoul, South Korea, doi: 10.1109/ROBOT.2001.933265.
3. Pratihar, D. K., Jain, L. C. Intelligent Autonomous Systems: Foundations and Applications. Springer, 2010.
4. Atmeh, G. M., Ranatunga, I., Popa, D. O., et al. Implementation of an Adaptive, Model Free, Learning Controller on Atlas Robot. *American Control Conference*, 2014 Portland, OR, USA, doi: 10.1109/ACC.2014.6859431.
5. Chestnutt, J., Lau, M., Cheung, G., et al. Footstep Planning for the Honda ASIMO Humanoid. *International Conference on Robotics and Automation*, 2005 Barcelona, Spain, doi: 10.1109/ROBOT.2005.1570188.
6. Gardecki, A., Podpora, M., Kawala-Janik, A. Implementation of an External Laser Scanner into Control System of the NAO Robot. – *IFAC-PapersOnLine*, 51, 2018, No. 6, doi: 10.1016/j.ifacol.2018.07.159.
7. Yoshikawa, M., Matsumoto, Y., Sumitani, M., et al. Development of an Android Robot for Psychological Support in Medical and Welfare Fields. *IEEE International Conference on Robotics and Biomimetics*, 2011, Karon Beach, Phuket, Thailand, doi: 10.1109/ROBIO.2011.6181654.
8. Haug, E. J., Computer-aided Kinematics and Dynamics of Mechanical Systems, Volume 1: Basic Methods. Boston: Allyn & Bacon, 1989.
9. Featherstone, R. Robot Dynamics Algorithms. Boston: Kluwer Academic Publishers, 1987.
10. Kim, J.-Y., Park, I.-W., Oh, J.-H. Walking Control Algorithm of Biped Humanoid Robot on Uneven and Inclined Floor. – *Journal of Intelligent and Robotic Systems*, 2007, doi: 10.1007/s10846-006-9107-8.
11. Ljung, L. System Identification: Theory for the User. 2nd Edition, Linkiping University, Sweden, 1997.
12. Isermann, R., Münchhof, M. Identification of Dynamic Systems, Springer, 2011.
13. Zhu, Y. Multivariable System Identification for Process Control. 1st Edition, 2001.

Manuscript received on 04.03.2019

Sherif Sherif *was born in Targovishte in 1994. He received a Master's degree (2019) in Automation from the Technical University of Sofia. He is currently a Ph.D. student at the Department of Systems and Control, Technical University – Sofia and work as production automation engineer at Syscont Ltd. His scientific interests include embedded systems, robotics, artificial intelligence and automation in petroleum industry.*

Contacts:
Systems and Control Department
Technical University – Sofia
e-mail: mr.shark94@abv.bg

Jordan Kralev *has M.Sc. degree in Automatics, Information and Control Technologies. Since March 2013 he is a Ph.D. student in the Department of Systems and Control, Technical University –Sofia. His work is in the field of implementation of complex control structures with programmable platforms.*

Contacts:
Systems and Control Department
Technical University – Sofia
e-mail: jkralev@yahoo.com

Tsonyo Slavov *was born in Sofia, Bulgaria in 1978. He graduated from the Department of Systems and Control of the Technical University of Sofia in 2002. He received Ph.D. degree in 2007. Presently he is an Associated Professor in the Department of Systems and Control at the Technical University – Sofia. He is co-authoring of the book Design of Embedded Robust Control Systems with Matlab/Simulink (IET Control, Robotics and Sensors Series, London, 2018).*

Contacts:
Systems and Control Department
Technical University – Sofia
e-mail: ts_slavov@tu-sofia.bg