Design and Construction of "ArtRo" Omnidirectional Walking Bipedal Robot using Hierarchical Reactive Behaviors Control Architecture

Giovanni Sutanto, Kusprasapta Mutijarsa, and Hilwadi Hindersah

Institut Teknologi Bandung, Bandung, Indonesia
[1]giovanni.sutanto@gmail.com, [2]soni@stei.itb.ac.id, [3]hilwadi@lskk.ee.itb.ac.id

Abstract: Bipedal robot locomotion has been a complicated and challenging matter over the time. This paper describes our work on building omnidirectional walking bipedal robot, included mechanical and electrical design aspect, and its behavior control as well. We adopted a technique introduced by Sven Behnke that enables omnidirectional walking on bipedal robot through the use hierarchical reactive behaviors control architecture.
We called the robot "ArtRo". Currently, ArtRo has successfully implemented a body consisting of two legs, lower abdomen, and a dummy weight representing the upper body load, with a total of 12 degrees-of-freedom (DOF). The mechanical system is designed optimally to facilitate the manufacturing process and the computation of robot parameters.
The electrical system of ArtRo consists of main controller and actuator system. The main controller is an Atmel's ATmega128-based microcontroller system. The actuator system implements a special method involving Time-Division Multiplexing (TDM) to control all ArtRo's servo motors.
The Sven Behnke's technique has a low computational complexity. It is proven by its successful implementation on ArtRo's main controller. Adaptation of the technique has been successfully implemented, ArtRo is capable to do omnidirectional walking, whether it is forward walking, side walking, or turning on the spot behavior.

Keywords: bipedal robot, design and construction, hierarchical reactive behaviors, omnidirectional walking

1. Introduction

The main purpose of this research is for development of low-cost bipedal robot. For achieving this purpose, a bipedal robot with 26 degrees-of-freedom (DOF) is designed. Yet, for bipedal walking technique implementation and testing stage, the manufactured robot body parts are only both of the legs and lower abdomen, while the upper body load are represented by a dummy weight. The manufactured robot body has 12 DOF in total. This robot is called ArtRo.
There are many bipedal walking techniques that have been invented. But, the most suitable one must be determined. The most common approach is trajectory tracking method based on Zero-Moment Point (ZMP) stability criterion. The ZMP is defined as the point on the ground about which the sum of the moments of all the active forces equals zero. In order to achieve a dynamically stable gait, the ZMP should be within the support polygon at every time instance [1]. However, trajectory tracking based on ZMP criterion requires dynamic model of robot's motion, which at most of the time causing the method to have a high computational complexity. In several researches, inverted pendulum is used as a simplified model of the robot's motion dynamic. For example, in [2], robot body is modelled as two-mass linear inverted pendulum, one of the mass is the swing leg and another is the overall mass of the robot's body except of the swing leg. State-space description is used to represent the robot's body dynamics, and then the ZMP desired trajectory is calculated from it.

Different approach, which is quite popular among bipedal robot researchers, is so-called

Received: February 22nd, 2011. Accepted: March 25th, 2011

passive-dynamic walking. Given only a downhill slope as a source of energy, a human-like pair of legs will settle into a natural gait generated by passive interaction of gravity and inertia; No muscular input is required [3]. In the development, powered passive-dynamic walking robots are built, which are using simple actuation to substitute the role of gravitational power [4]. The advantage of adopting passive-dynamic walking method is that the system will be very energy-efficient. However, a quite complicated mechanical system will be required for the robot to implement this technique.

Sven Behnke introduced another approach in [5], which enables omnidirectional walking for bipedal robot. In contrast to the previously-mentioned approaches, which may require a stop in walking when changing direction or turning, this approach generates smoother robot trajectories. The omnidirectional gait was not optimized for speed, but for stability and parameterizability. Moreover, this approach has a low computational complexity.

Our team would like to use minimum cost possible for the system, one way is by using low-cost but reliable microcontroller system, which will require a low computational complexity bipedal walking technique. Therefore, Sven Behnke's omnidirectional bipedal walking technique seems to be the best to be implemented on our robot.

In the next section, ArtRo's mechanical system design will be described. Section 3 describes ArtRo's electrical system design. Section 4 describes ArtRo's behavior control to implement omnidirectional walking. Section 5 shows experimental results obtained from implementation of this technique. Finally we conclude in Section 6.

2. Mechanical System

The mechanical design of ArtRo is made based on observation of human body's anatomy. The first complete mechanical design of ArtRo is depicted on Figure 1.

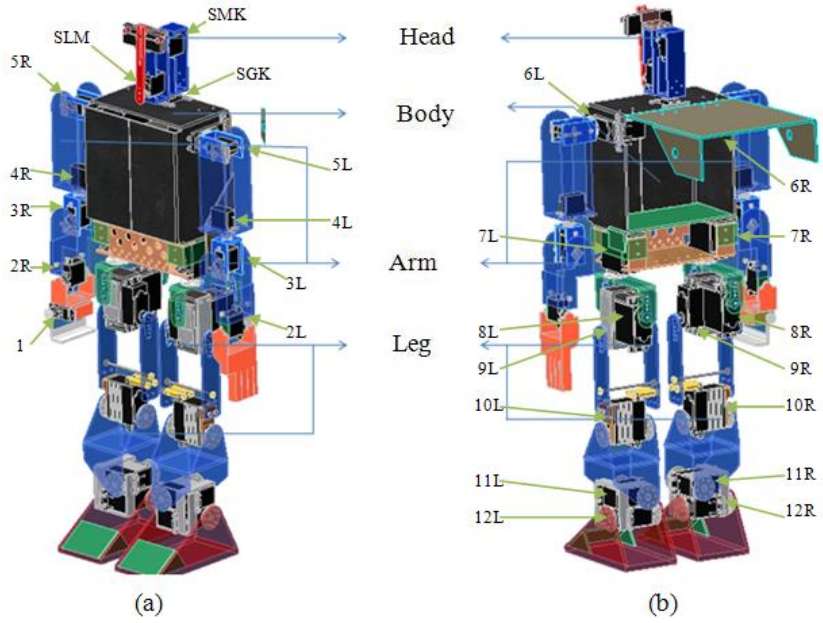


Figure 1. The first complete mechanical design of ArtRo: (a) Front-lateral view; (b) Rear-lateral view.

In Figure 1, joint's numeric code denotes its specific position on the body, while suffix 'R' or 'L' denotes whether the joint is on the right ('R') side or on the left ('L') side of the body. The complete design consists of 26 joints (26 DOF).

Servo motors are used as the joints. Before deciding which servo motor type to be used on a specific joint, the required capacity of torque must be calculated. For this calculation, there are four parameters of the concerned joint that must be determined:

- Total load weight estimate (*W*),
- Length/distance estimate from the joint axle to the load center of mass (*l*),
- Movement angle of coverage of the joint axle ($\alpha_{min} \leq \alpha \leq \alpha_{max}$), and
- Safety factor percentage (*sf*).

Please note that 'load' mentioned in the above parameters is joint-specific. Safety factor is included to represent unpredictable additional load on the joint operation. The required torque capacity of the concerned joint can then be calculated as follows:

$$\tau = (1 + sf) \cdot W \cdot l \cdot \max(\sin \alpha) \quad (1)$$

The function $\max(\sin \alpha)$ in Equation (1) returns the maximum value of $(\sin \alpha)$ within range $\alpha_{min} \leq \alpha \leq \alpha_{max}$.

The above calculation is done firstly by hand and later by the help of mechanical simulation software, to get a more accurate result. Beside of torque capacity requirement, servo motor dimension and weight also become two considered factors in servo motor type selection.

Even though the design in Figure 1 seems to be good, some mechanical components were rather difficult to be manufactured. Therefore, to facilitate the manufacturing process and the computation of parameters in the bipedal walking technique implementation, the design must be modified. There are several considerations for this modification, as follows:

A. Aluminum Sheet Metal Components Usage

Most mechanical components of the new design are made of aluminum sheet metal which is cut and folded.

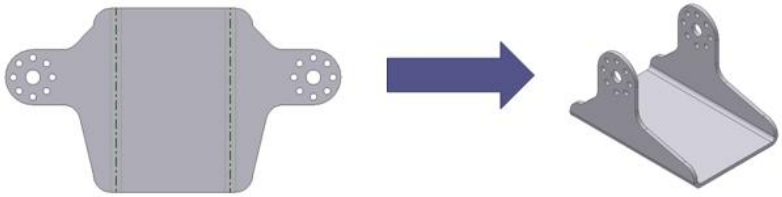


Figure 2. Example of cut-and-fold aluminum sheet metal component.

Aluminum is chosen over other metals, because it is cheap, easy-to-find, and having a good ratio of strength versus weight. Moreover, the components are easy to be manufactured using available mechanical tools in our workshop.

B. Coupled Joints Design

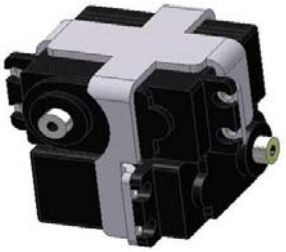


Figure 3. Example of coupled joints

.

At some positions, two joints are coupled into one, so that both of its axis are crossing-over each other perpendicularly, as close as possible. This applies at hip and at ankle, between each of its pitch-direction joint and roll-direction joint. This design consideration is made to mimic human's anatomy, at which between hip-pitch and hip-roll and between ankle-pitch and ankle-roll, the movement axis are colliding perpendicularly each other. Since on ArtRo we are using one-axis servo motor as joint, this condition is impossible to be reached. However, we try to reach the closest condition to human's, by placing both joints as close as possible.

C. Servo Bottom Support Axle Addition

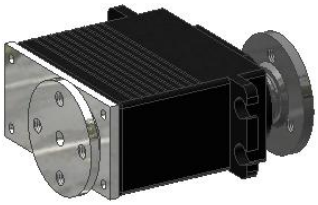


Figure 4. Example of servo motor with additional bottom support axle.

Most types of small-size servo motor only have one-side (active) axle, that is on top of its package. However, to handle a heavy load sturdily, an additional axle is needed. Therefore, an additional pair of mounting and support (passive) axle are added on the bottom side of the servo motor's package. The example of additional pair of mounting and support axle is depicted as parts drawn in silver in Figure 4.

D. One-Line Axis Design

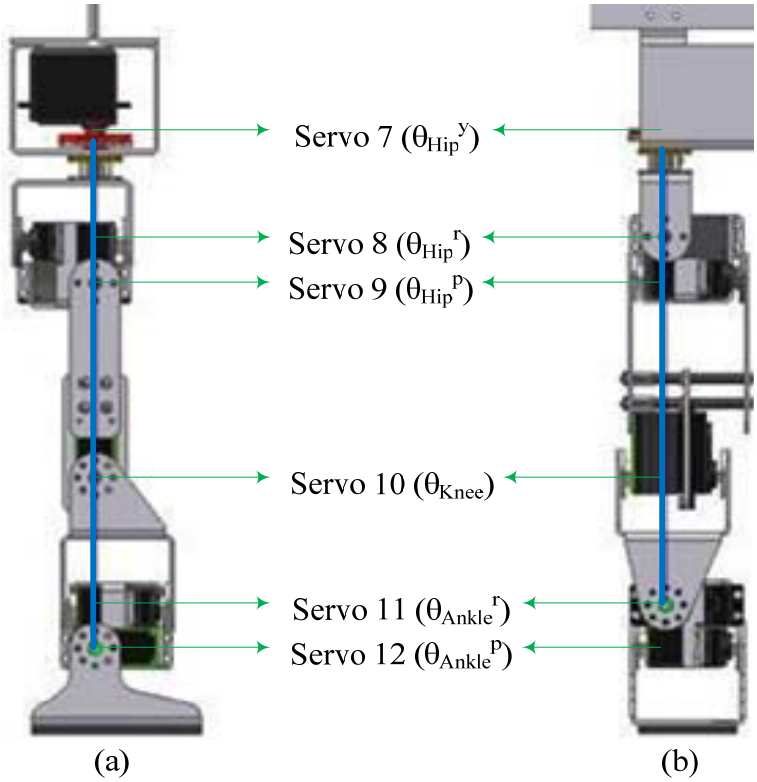


Figure 5. One-line axis design: (a) Seen from lateral-side; (b) Seen from front-side.

Figure 5 illustrates one-line axis design concept. Blue lines in Figure 5 (a) and (b) indicate hip-yaw (θ_{Hip}^{y}) joint axis, seen from lateral-side and from front-side, respectively. When every leg joints are at 0° reference angle (occurs when ArtRo is standing upright), the hip-yaw (θ_{Hip}^{y}) joint axis will collide hip-pitch (θ_{Hip}^{p}), knee (θ_{Knee}), and ankle-pitch (θ_{Ankle}^{p}) joint axis perpendicularly, all are located at the coronal plane [6], as illustrated in Figure 5 (a); At the same time, the hip-yaw (θ_{Hip}^{y}) joint axis will also collide hip-roll (θ_{Hip}^{r}) and ankle-roll (θ_{Ankle}^{r}) joint axis perpendicularly, all are located at a plane parallel to sagittal plane [6], as illustrated in Figure 5 (b). This design concept will simplify the inverse kinematics computation.

For bipedal walking technique implementation and testing stage, the manufactured parts were only the two legs and lower abdomen, while the upper body load is represented by a dummy weight, making a total of 12 DOF bipedal robot. The manufactured design is depicted in Figure 6.

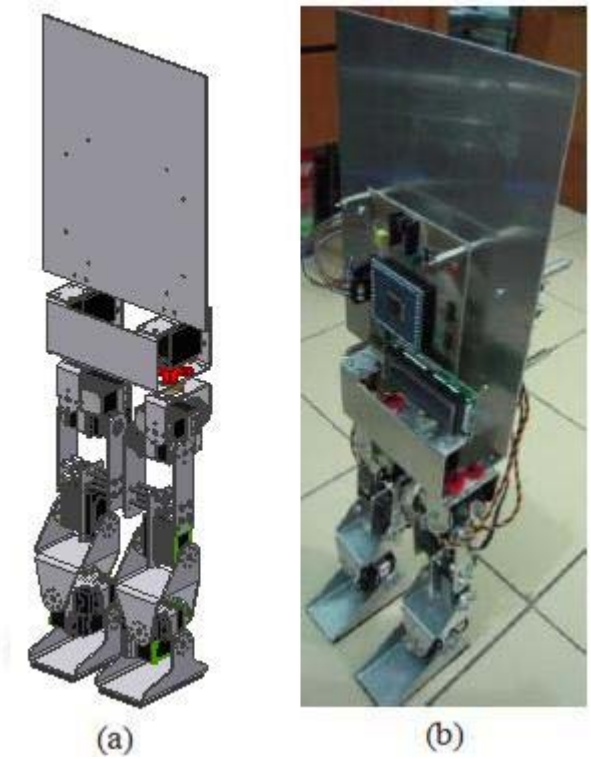


Figure 6. The manufactured design: (a) Mechanical design simulation; (b) Realization, along with electrical system.

3. Electrical System

The electrical system consists of main controller and actuator system. There is currently no sensor needed for the implementation of this bipedal walking technique. But for future development, some pressure sensors, accelerometers, and gyroscopes may be used as balance sensors, to implement a more robust bipedal walking technique on ArtRo.

A. Main Controller

The main controller is using Atmel's ATmega128-based microcontroller system. The following are the microcontroller features that are used for the bipedal walking technique implementation.

- *Input/Output (I/O) Ports*
  I/O ports are used to generate control signals for Liquid Crystal Display (LCD) and to produce demultiplexing synchronization signals for actuator driver circuit (which will be explained later). LCD is used to display parameters being observed, for debugging purpose.
- *Timer/Counter*
  There are two 16-bit timer/counter units and two 8-bit timer/counter units in ATmega128 microcontroller.

The 16-bit timer/counter units are used to generate Pulse Width Modulation (PWM) signals. Each of the 16-bit timer/counter unit has three independent output compare units, so that by using both 16-bit timer/counter units, a total of six independent 16-bit PWM signals can be generated. These PWM signals are sent to actuator driver circuit, and together with the control signals from I/O ports, all actuators (servo motors) of ArtRo are controlled.

One of the 8-bit timer/counter units has a different role on ArtRo. It is used as the walking clock in the bipedal walking technique implementation.

- *Universal Asynchronous Receiver/Transmitter (UART)*
  UART manages communication between microcontroller system and Personal Computer (PC) through serial port, when a human-robot interface is needed. The human-robot interface software has been developed to do testing on robot software parameterization by commands sent from PC keyboard.
- *Interrupt*
  Interrupts are used in the timer/counter units operation, to gain a good-precision timing.

Beside of the above mentioned features, ATmega128 microcontroller also has 8 channels of 10-bit Analog-to-Digital Converter (ADC) that are capable of receiving analog input signals from sensors and convert it into digital representations for further processings, if there is any development on ArtRo in the future.

B. Actuator System

As mentioned before, ArtRo is using servo motors as actuators. Servo motors that are used on ArtRo are Direct-Current (DC) motors with positional control. The control signal required for controlling this type of servo motor is PWM signal with 20 milliseconds period (T = 20 ms) and pulse length varies between 0.6 to 2.4 milliseconds (0.6 ms $\leq t_H \leq$ 2.4 ms), as depicted in Figure 7 (a). Angular position of the servo motor axle depends proportionally linear to the pulse length (t_H) parameter of the PWM signal fed to it.

Since the final complete design shall have 26 DOF, it means that 26 independent PWM signals will be needed to control 26 servo motors independently each other on ArtRo. However, the microcontroller system only has six channels of 16-bit PWM signal.

Therefore, a special method involving Time-Division Multiplexing (TDM) is used on ArtRo. The PWM signal for controlling servo motor has maximum pulse length (t_H) as long as 2.4 ms, while the rest of it until the end of one signal cycle, as long as (20 – 2.4) ms = 17.6 ms, is actually 'blank' and meaningless, as depicted in Figure 7 (a). The 'significant' part of the PWM signal for controlling servo motor is the part containing the pulse length, as long as 2.4 ms (or 2.5 ms to be safe), beside of the signal period that has to be 20 ms. The method exploits the 'blank' spot of one PWM signal, by putting the 'significant' parts of several other PWM signals on it. The combined signal is called Time-Division Multiplexed PWM (TDM-PWM) signal. Using this method, a maximum of (20 ms 4 2.5 ms) = 8 PWM signals can be put together into 1 TDM-PWM signal. For implementing this method, an additional device is needed to demultiplex the resulted TDM-PWM signal, back into the original PWM signals.

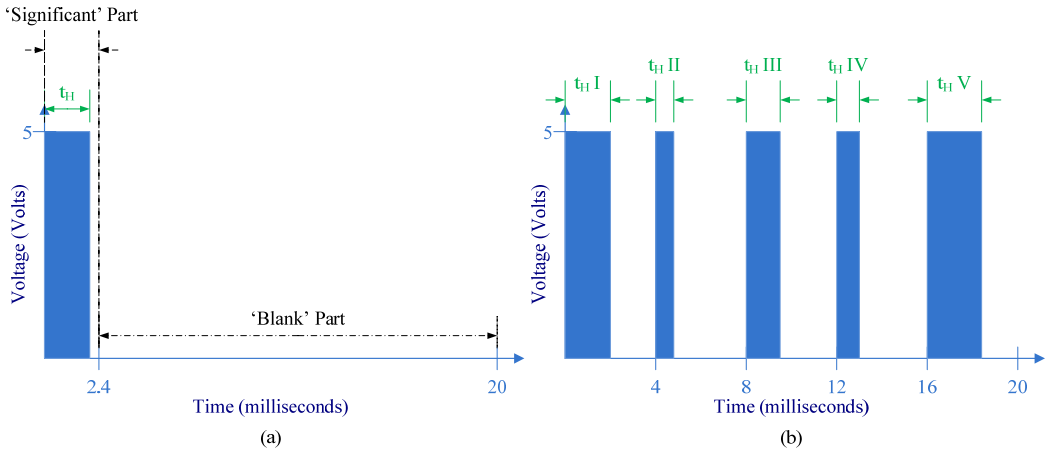


Figure 7. One-cycle signal visualizations: (a) PWM signal for controlling servo motor; (b) TDM-PWM signal resulted from combination of 5 PWM signals.

On ArtRo, each channels of 16-bit PWM signal are programmed to transmit TDM-PWM signal which is a combination of 5 PWM signals, as depicted in Figure 7 (b). In Figure 7 (b), t_H I, t_H II, t_H III, t_H IV, and t_H V represent the pulse length of 5 PWM signals that are combined into the TDM-PWM signal.

The TDM-PWM signal is transmitted to the actuator driver circuit to be demultiplexed based on demultiplexing synchronization signals generated by microcontroller's I/O ports. With six 16-bit PWM signal channels available on the microcontroller and the role of the actuator driver circuit as demultiplexer, it is possible to generate 30 independent PWM signals, which are more than enough to control all servo motors of ArtRo.

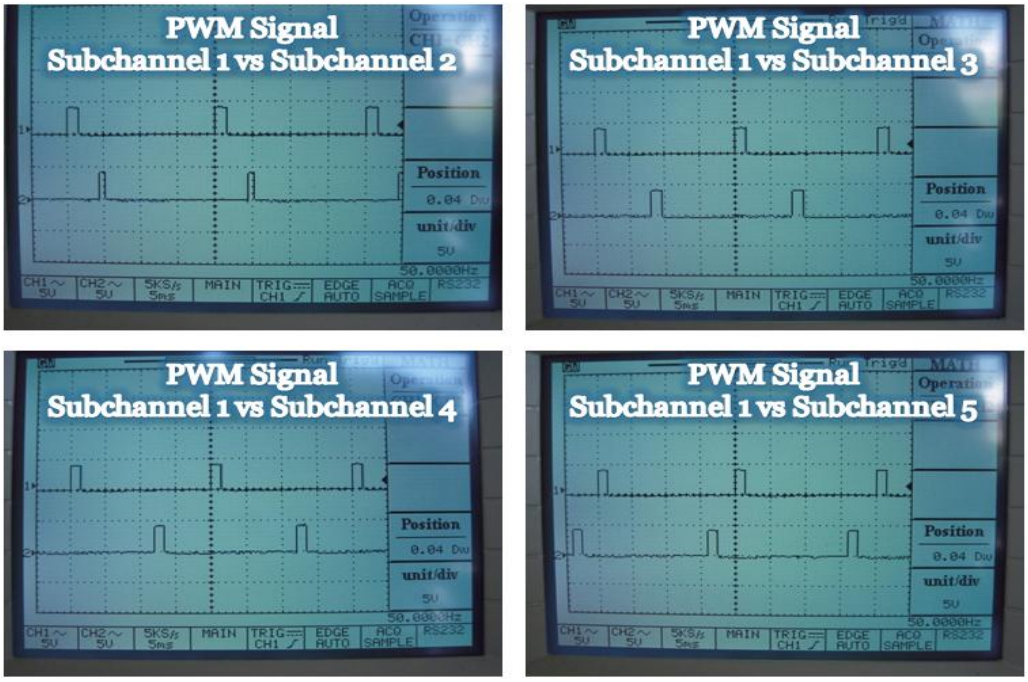


Figure 8. Comparison between PWM signals which are generated from the demultiplexation of one-channel TDM-PWM signal.

Figure 8 shows demultiplexation results of one-channel TDM-PWM signal, seen on oscilloscope. It can be seen that between one PWM signal to the others are not started at the same time (i.e. not synchronous), due to the TDM method implementation. But this is not a problem, since the PWM signal period is still 20 ms and the pulse length is still in the range of 0.6 ms $\leq t_H \leq$ 2.4 ms. The resulted PWM signals can be directly used to control the servo motors.

To give summary on the electrical system of ArtRo, the schematics is depicted in Figure 9.

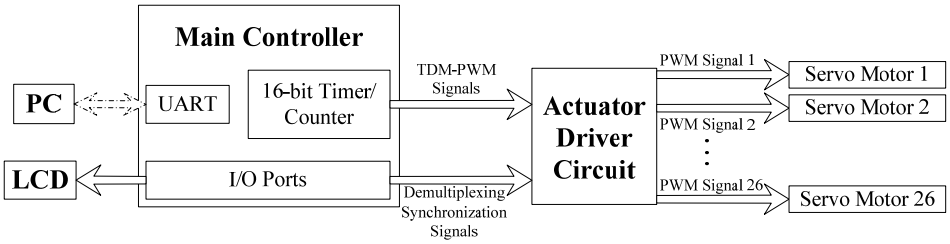
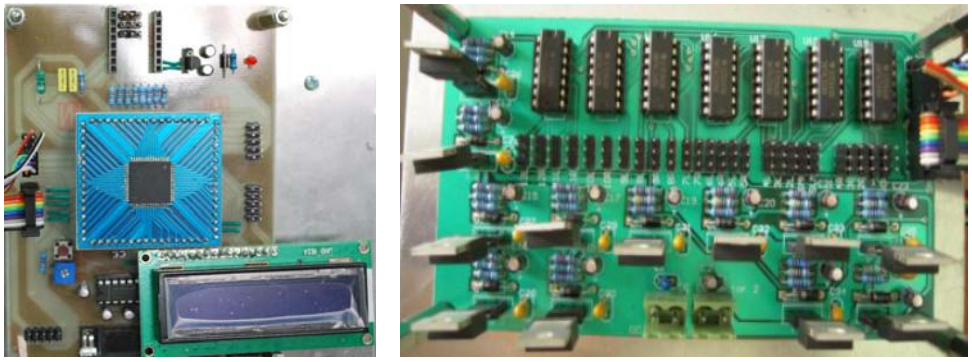


Figure 9. Electrical system schematics.



(a) (b)

Figure 10. Electrical system components: (a) Main controller; (b) Actuator driver circuit.

4. Behavior Control Architecture

To implement omnidirectional walking on ArtRo, an architecture that supports a hierarchy of reactive behaviors [7] is used in the actuator side of ArtRo. The architecture requires definition of multiple layers of different behavior complexity, as depicted in Figure 11.

In Figure 11, the top layer is motion vector of ArtRo, followed by walking phase, leg, and individual joint angle at the bottom layer. The individual joint angle (or angular position of each servo motor axle) at the bottom layer is concrete actuator, while higher layers (leg, walking phase, and motion vector) are abstract actuators.

Interfaces are made between abstract actuators at higher layers and its lower layers, to give higher-level behaviors the possibility to configure the lower layers. By changing parameters in higher layer abstract actuators (e.g. γ in leg), the lower layers (e.g. individual joints as the concrete actuators) are manipulated (or actuated) in a coordinated way. The next subsections will mainly explain about interfaces between layers, since it defines relationship between layers, which is the core of control scheme of omnidirectional walking on ArtRo.

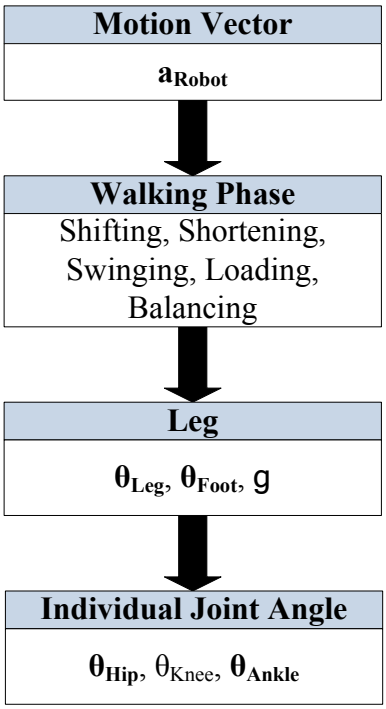


Figure 11. Actuator hierarchy of reactive behaviors.

A. Leg Interface

The entire leg can be positioned relative to the trunk using three parameters: leg angle, foot angle, and leg extension. The leg angle and foot angle parameters can be visualised using Figure 12.

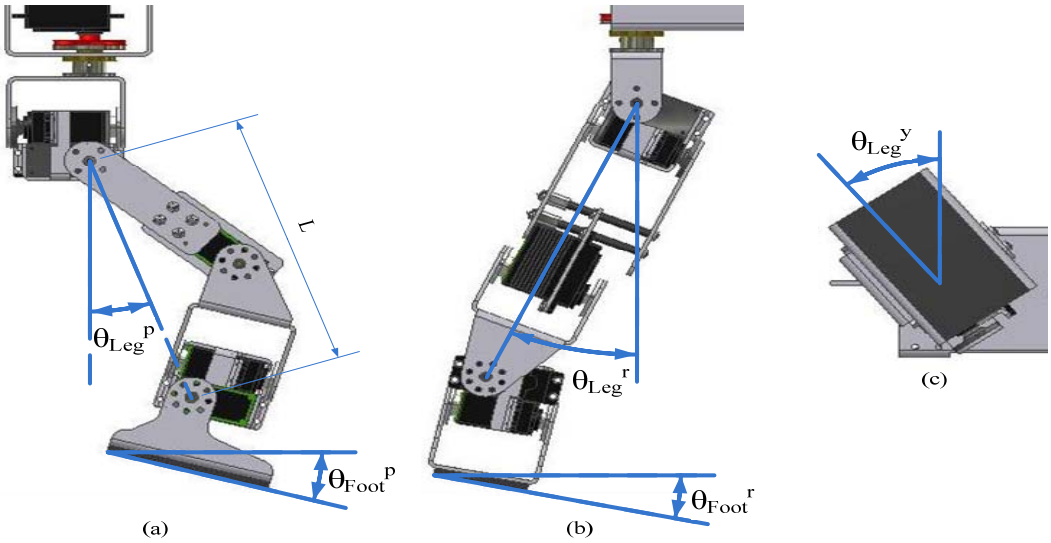


Figure 12. The leg angle ($\boldsymbol{\theta_{Leg}}$) and foot angle ($\boldsymbol{\theta_{Foot}}$) parameters: (a) Leg-pitch angle (θ^{p}_{Leg}) and foot-pitch angle (θ^{p}_{Foot}); (b) Leg-roll angle (θ^{r}_{Leg}) and foot-roll angle (θ^{r}_{Foot}); (c) Leg-yaw angle (θ^{y}_{Leg}), seen from under transverse plane [6].

- *Leg Angle* $\left(\boldsymbol{\theta_{Leg}} = \left(\theta^r_{Leg}, \theta^p_{Leg}, \theta^y_{Leg}\right)\right)$

The leg angle is the angle between the trunk and the line from hip to ankle (leg line), with convention:

- for leg-roll angle:
  $\theta^r_{Leg} > 0$ if the leg is moved outwards against the sagittal plane.
  $\theta^r_{Leg} = 0$ if the leg is parallel to the sagittal plane.
  $\theta^r_{Leg} < 0$ if otherwise.
  Figure 12 (b) shows an example of positive-value leg-roll angle ($\theta^r_{Leg} > 0$).
- for leg-pitch angle:
  $\theta^p_{Leg} > 0$ if the leg is moved frontwards against the coronal plane.
  $\theta^p_{Leg} = 0$ if the leg is located at the coronal plane.
  $\theta^p_{Leg} < 0$ if otherwise.
  Figure 12 (a) shows an example of positive-value leg-pitch angle ($\theta^p_{Leg} > 0$).
- for leg-yaw angle:
  $\theta^y_{Leg} > 0$ if the leg is twisted, such that the foot points outwards.
  $\theta^y_{Leg} = 0$ if the foot points in forward direction.
  $\theta^y_{Leg} < 0$ if otherwise.
  Figure 12 (c) shows an example of positive-value leg-yaw angle ($\theta^y_{Leg} > 0$).

- *Foot Angle* $\left(\boldsymbol{\theta_{Foot}} = \left(\theta^r_{Foot}, \theta^p_{Foot}\right)\right)$

The foot angle is the angle between pelvis plate and foot plate, with convention:

- for foot-roll angle:
  $\theta^r_{Foot} > 0$ if the inner side of the foot plate points outwards against the pelvis plate.
  $\theta^r_{Foot} = 0$ if the foot plate is parallel to the pelvis plate.
  $\theta^r_{Foot} < 0$ if otherwise.
  Figure 12 (b) shows an example of positive-value foot-roll angle ($\theta^r_{Foot} > 0$).
- for foot-pitch angle:
  $\theta^p_{Foot} > 0$ if the frontal side of the foot plate points inwards against the pelvis plate.
  $\theta^p_{Foot} = 0$ if the foot plate is parallel to the pelvis plate.
  $\theta^p_{Foot} < 0$ if otherwise.
  Figure 12 (a) shows an example of negative-value foot-pitch angle ($\theta^p_{Foot} < 0$).

- *Leg Extension* (γ)

The value range of leg extension is $-1.0 \leq \gamma \leq 0$, with convention:
$\gamma = 0$ if the leg is fully extended ($\eta_{max} = 1.0$).
$\gamma = -1.0$ if the leg is shortened to $\eta_{min} = 0.875$.

η is the proportion of the leg length against its maximum value, or shortly can be said as relative leg length. The leg length itself is defined as the absolute distance between hip-pitch joint axis and ankle-pitch joint axis. Target relative leg length η can be computed from leg extension γ as:

$$\eta = 1.0 + \left((1.0 - \eta_{min}) \cdot \gamma\right) \quad (2)$$

Knee angle θ_{Knee} is manipulated for leg shortening process, but it would also change the leg angle $\boldsymbol{\theta_{Leg}}$ and foot angle $\boldsymbol{\theta_{Foot}}$. To compensate this effect, hip-pitch angle θ^p_{Hip} and ankle-pitch angle θ^p_{Ankle} need to be manipulated as well. The manipulations of θ_{Knee}, θ^p_{Hip}, and θ^p_{Ankle} for leg shortening process are represented by α_1, α_2, and α_3, respectively, as depicted in Figure 13.

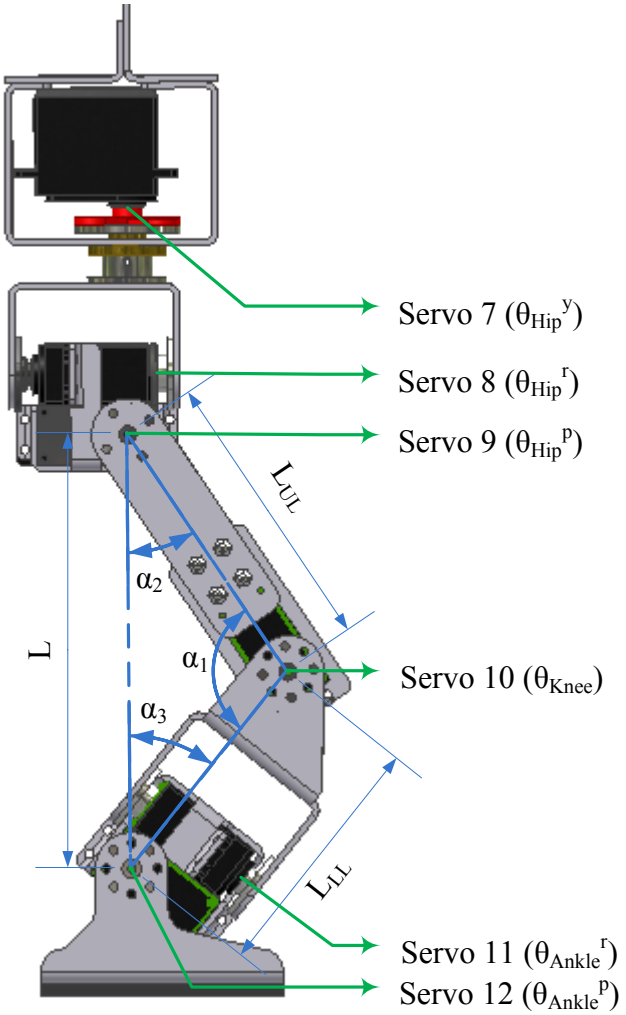


Figure 13. Manipulations for leg shortening process.

In Figure 13, L is the variable leg length, while L_{UL} is the constant upper leg length and L_{LL} is the constant lower leg length. Clearly, the maximum value of L is:

$$L_{max} = L_{UL} + L_{LL}$$

By definition:

$$\eta = \frac{L}{L_{max}}$$

Then L can be expressed as:

$$L = \eta \cdot L_{max} = \eta \cdot (L_{UL} + L_{LL}) \quad (3)$$

For defining relationship between leg extension γ and individual joint angles, α_1, α_2, and α_3 in Figure 13 need to be computed. Therefore, a simple inverse kinematics computation based on triangular and trigonometric properties is used.

Based on triangular property, relationship between L and α_1 in Figure 13 is:

$$L^2 = {L_{UL}}^2 + {L_{LL}}^2 - (2 \cdot L_{UL} \cdot L_{LL} \cdot \cos \alpha_1)$$

Then α_1 can be computed from L as:

$$\alpha_1 = \cos^{-1}\left(\frac{{L_{UL}}^2+{L_{LL}}^2-L^2}{2 \cdot L_{UL} \cdot L_{LL}}\right) \quad (4)$$

Based on trigonometric property, relationship between α_1 and α_2 in Figure 13 is:

$$\frac{\sin \alpha_1}{L} = \frac{\sin \alpha_2}{L_{LL}}$$

Then α_2 can be computed from α_1 as:

$$\alpha_2 = \sin^{-1}\left(\frac{L_{LL}}{L} \cdot \sin \alpha_1\right) \quad (5)$$

Also based on triangular property, relationship between α_1, α_2, and α_3 in Figure 13 is:

$$\alpha_1 + \alpha_2 + \alpha_3 = \pi$$

Then α_3 can be computed from α_1 and α_2 as:

$$\alpha_3 = \pi - (\alpha_1 + \alpha_2) \quad (6)$$

Since α_1, α_2, and α_3 have been translated from leg extension γ by Equation (2) through (6), with leg angle $\boldsymbol{\theta_{Leg}}$ and foot angle $\boldsymbol{\theta_{Foot}}$ as additional information, the individual joint angles can be computed as:

- *Hip Angle* $\left(\boldsymbol{\theta_{Hip}} = \left(\theta^r_{Hip}, \theta^p_{Hip}, \theta^y_{Hip}\right)\right)$

$$\boldsymbol{\theta_{Hip}} = \begin{pmatrix}\theta^r_{Hip}\\ \theta^p_{Hip}\\ \theta^y_{Hip}\end{pmatrix} = \begin{pmatrix}\theta^r_{Leg}\\ \theta^p_{Leg}\\ \theta^y_{Leg}\end{pmatrix} + \begin{pmatrix}0\\ \alpha_2\\ 0\end{pmatrix} \quad (7)$$

- *Knee Angle* (θ_{Knee})

$$\theta_{Knee} = \alpha_1 - \pi \quad (8)$$

- *Ankle Angle* $\left(\boldsymbol{\theta_{Ankle}} = \left(\theta^r_{Ankle}, \theta^p_{Ankle}\right)\right)$

$$\boldsymbol{\theta_{Ankle}} = \begin{pmatrix}\theta^r_{Ankle}\\ \theta^p_{Ankle}\end{pmatrix} = \begin{pmatrix}0\\ \alpha_3\end{pmatrix} + \left(\begin{pmatrix}\theta^r_{Foot}\\ \theta^p_{Foot}\end{pmatrix} - \begin{pmatrix}\theta^r_{Leg}\\ \theta^p_{Leg}\end{pmatrix}\right) \quad (9)$$

Compared to hip angle $\boldsymbol{\theta_{Hip}}$ and ankle angle $\boldsymbol{\theta_{Ankle}}$ computations which are described in [5], the same-purpose of computations which are described in Equation (7) and (9) in this paper, respectively, are simpler because no matrix multiplication is involved. This computation complexity difference is caused by robot's mechanical structure difference in the placement of leg-yaw concrete actuator.

Humanoid robot Jupp, which is introduced in [5], has thigh joint as leg-yaw concrete actuator, placed between hip and knee joints, so that it can rotate the knee joint, lower leg, ankle joints, and foot, against the hip. As consequence, the computation of $\boldsymbol{\theta_{Hip}}$ and $\boldsymbol{\theta_{Ankle}}$ in bent knee $(\theta_{Knee} < 0)$ and twisted leg $\left(\theta_{Thigh} = \theta^y_{Leg} \neq 0\right)$ condition involves multiplication with some rotation matrix.

In the other hand, ArtRo has hip-yaw joint as leg-yaw concrete actuator, placed above hip-roll and hip-pitch joints, so that it can rotate the whole leg (including hip-roll, hip-pitch, knee, ankle-roll, and ankle-pitch joints) against the trunk. Thus, no matrix multiplication is required in $\boldsymbol{\theta_{Hip}}$ and $\boldsymbol{\theta_{Ankle}}$ computations.

Equation (2) through (9) gives complete definition of leg interface, that is the interface between leg parameters and individual joint angles. In the next subsection, the interfaces between higher layers (leg, walking phase, and motion vector) will be defined.

B. Omnidirectional Walking

To be able to walk properly, ArtRo requires a timing guide which is called walking clock ϕ_{Trunk}. This walking clock is generated by one of main controller's 8-bit timer/counter unit. Its value is maintained within range $-\pi \leq \phi_{Trunk} < \pi$.

Both legs of ArtRo have their own gait phase ϕ_{Leg}^{L} (for left leg) and ϕ_{Leg}^{R} (for right leg), which are:

$$\phi_{Leg}^{L} = \phi_{Trunk} + \frac{\pi}{2} \tag{10}$$

$$\phi_{Leg}^{R} = \phi_{Trunk} - \frac{\pi}{2} \tag{11}$$

The value of ϕ_{Leg}^{L} and ϕ_{Leg}^{R} are also maintained within range $-\pi \leq \phi_{Leg} < \pi$ by addition of $\pm 2\pi$ to the result of Equation (10) and (11), when necessary. Based on its gait phase, leg extension γ, leg angle $\boldsymbol{\theta_{Leg}}$, and foot angle $\theta_{\mathbf{Foot}}$ trajectories are generated for each leg.

In omnidirectional walking, the target walking direction, magnitude, and rotation are specified by motion vector $\mathbf{a\_Robot} = (a\_Robot\hat{}r, a\_Robot\hat{}p, a\_Robot\hat{}y)$, where a^{p}_{Robot} denotes the walk in forward (sagittal) direction, a^{r}_{Robot} denotes the walk in lateral direction, and a^{y}_{Robot} denotes ArtRo's rotation around the vertical axis.

The motion vector $\mathbf{a_{Robot}}$ is then translated into the lower layer, that is into the walking phase parameters. According to [5], with some parameter adjustments based on several tests, the definitions of walking phase parameters are:

- *Shifting*

The first walking phase is the lateral shifting of ArtRo's center of mass:

$$\theta_{Shift} = a_{Shift} \cdot \cos\left(\phi_{Leg} + \frac{\pi}{2}\right) \tag{12}$$

where:

$$a_{Shift} = 0.12 + 0.08 \cdot \left\|\left(a^{r}_{Robot}, a^{p}_{Robot}\right)\right\| + 0.7 \cdot \left|a^{r}_{Robot}\right| \tag{13}$$

a_{Shift} is the shifting amplitude, which increases with the gait speed as well as with the lateral speed. The leg-roll and foot-roll joints are used to side-shift ArtRo:

$$\theta_{LegShift} = \theta_{Shift} \tag{14}$$

$$\theta_{FootShift} = -0.5 \cdot \theta_{Shift} \tag{15}$$

This keeps the upper body upright in the lateral plane.

- *Shortening*

As ArtRo shifts to a side, the opposite leg is shortened, because it is not needed to support the weight. The time course of the shortening is determined by the shortening phase:

$$\phi_{Short} = v_{Short} \cdot \left(\phi_{Leg} + \frac{\pi}{2} + o_{Short}\right) \tag{16}$$

where:

$$v_{Short} = 3.0 \tag{17}$$

$$o_{Short} = -0.05 \tag{18}$$

v_{Short} determines the duration of the shortening and o_{Short} determines the phase shift of the shortening, relative to lateral weight shifting. The leg shortening is defined as:

$$\gamma_{Short} = \begin{cases} -0.5 \cdot a_{Short} \cdot (\cos(\phi_{Short}) + 1) & \text{if } -\pi \leq \phi_{Short} < \pi \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where:

$$a_{Short} = 0.2 + 2.0 \cdot \left\|\left(a^{r}_{Robot}, a^{p}_{Robot}\right)\right\| \tag{20}$$

a_{Short} is the shortening amplitude, which increases with the gait speed. To lift the foot at the end pointing into walking direction, a similar term is used:

$$\theta_{FootShort} = \begin{cases} -0.125 \cdot a_{Robot}^{p} \cdot (\cos(\phi_{Short}) + 1.0) & \text{if } -\pi \leq \phi_{Short} < \pi \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

This avoids accidental contact between the leading part of the foot and the ground during swing.

- *Swinging*

After the leg has been shortened, it is moved quickly into the walking direction. This swing is reversed slowly during the rest of the gait cycle. The time course of the swing is described by the swinging phase:

$$\phi_{Swing} = v_{Swing} \cdot \left(\phi_{Leg} + \frac{\pi}{2} + o_{Swing}\right) \quad (22)$$

where:

$$v_{Swing} = 2.0 \quad (23)$$

$$o_{Swing} = -0.15 \quad (24)$$

v_{Swing} determines the duration of the swinging and o_{Swing} determines the phase shift of the swinging, relative to lateral weight shifting. While the swinging is sinusoidal, the reverse motion is linear:

$$\theta_{Swing} = \begin{cases} \sin(\phi_{Swing}) & \text{if } -\frac{\pi}{2} \leq \phi_{Swing} < \frac{\pi}{2} \\ \left(b \cdot \left(\phi_{Swing} - \frac{\pi}{2}\right)\right) + 1 & \text{if } \phi_{Swing} \geq \frac{\pi}{2} \\ \left(b \cdot \left(\phi_{Swing} + \frac{\pi}{2}\right)\right) - 1 & \text{otherwise} \end{cases} \quad (25)$$

The speed of the reverse motion is:

$$b = -\frac{2}{(2 \cdot \pi \cdot v_{Swing}) - \pi} \quad (26)$$

The swing is done with the leg angle and balanced partially with the foot angle:

$$\theta_{LegSwing}^{r} = Is \cdot a_{Robot}^{r} \cdot \theta_{Swing} \quad (27)$$

$$\theta_{LegSwing}^{p} = a_{Robot}^{p} \cdot \theta_{Swing} \quad (28)$$

$$\theta_{LegSwing}^{y} = Is \cdot a_{Robot}^{y} \cdot \theta_{Swing} \quad (29)$$

$$\theta_{FootSwing}^{r} = 0.25 \cdot a_{Robot}^{r} \cdot \theta_{Swing} \quad (30)$$

$$\theta_{FootSwing}^{p} = Is \cdot 0.25 \cdot a_{Robot}^{p} \cdot \theta_{Swing} \quad (31)$$

where:

$$Is = \begin{cases} -1 & \text{for left leg} \\ 1 & \text{for right leg} \end{cases} \quad (32)$$

- *Loading*

Immediately after the leg is fully extended and the heel landed, it is shortened a second time, in order to facilitate the loading of this leg. The loading phase is:

$$\phi_{Load} = \left(v_{Load} \cdot \text{piCut}\left(\phi_{Leg} + \frac{\pi}{2} - \frac{\pi}{v_{Short}} + o_{Short}\right)\right) - \pi \quad (33)$$

where:

$$v_{Load} = 3.0 \quad (34)$$

v_{Load} determines the duration of the loading. The function piCut(*) maps its argument to the range $[-\pi,\pi)$ by adding multiples of $\pm 2\pi$. The leg shortening in loading phase is defined as:

$$\gamma_{Load} = \begin{cases} -0.5 \cdot a_{Load} \cdot (\cos(\phi_{Load}) + 1) & \text{if } -\pi \leq \phi_{Load} < \pi \\ 0 & \text{otherwise} \end{cases} \quad (35)$$

where:

$$a_{Load}=0.025+\left(0.5\cdot\left(1-\cos\left(\left|a^{p}_{Robot}\right|\right)\right)\right) \quad (36)$$

a_{Load} is the loading amplitude, which depends on the sagittal amplitude.

- *Balancing*

ArtRo is balanced by tilting it every step in the sagittal and coronal/lateral plane, and by adding offsets to the leg and foot angles:

$$\theta^{r}_{FootBal}=0.5\cdot Is\cdot a^{r}_{Robot}\cdot\cos\left(\phi_{Leg}+0.35\right) \quad (37)$$

$$\theta^{p}_{FootBal}=0.02+\left(0.12\cdot a^{p}_{Robot}\right)-\left(0.04\cdot a^{p}_{Robot}\cdot\cos\left(2\cdot\phi_{Leg}+0.7\right)\right) \quad (38)$$

$$\theta^{r}_{LegBal}=0.05+\left(Is\cdot a^{r}_{Robot}\right)+\left|a^{r}_{Robot}\right|+\left(0.1\cdot a^{y}_{Robot}\right) \quad (39)$$

The leg-roll angle makes sure that the legs do not collide while walking to the side or while rotating.

- *Output*

The individual components of walking phase parameters are then combined as follows:

$$\theta^{r}_{Leg}=\theta^{r}_{LegSwing}+\theta_{LegShift}+\theta^{r}_{LegBal} \quad (40)$$

$$\theta^{p}_{Leg}=\theta^{p}_{LegSwing} \quad (41)$$

$$\theta^{y}_{Leg}=\theta^{y}_{LegSwing} \quad (42)$$

$$\theta^{r}_{Foot}=\theta^{r}_{FootSwing}+\theta_{FootShift}+\theta^{r}_{FootBal} \quad (43)$$

$$\theta^{p}_{Foot}=\theta^{p}_{FootSwing}+\theta_{FootShort}+\theta^{p}_{FootBal} \quad (44)$$

$$\gamma=\gamma_{Short}+\gamma_{Load} \quad (45)$$

The interface between motion vector (top layer) and walking phase parameters has been completely defined by Equation (12) through(39), while the interface between walking phase and leg parameters has been completely defined by Equation(40) through (45). Both definitions complete the actuator hierarchy of reactive behaviors of ArtRo depicted in Figure 11.

5. Experimental Results

The actuator hierarchy of reactive behaviors containing omnidirectional walking described in Section 4 has been successfully implemented on ArtRo, with some modifications in parameter values.

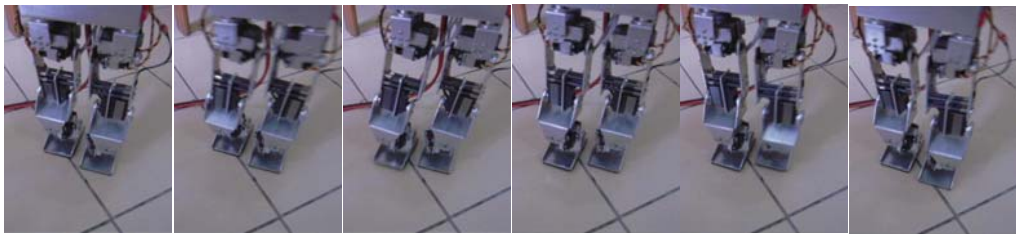


Figure 14. Image sequence extracted from video of ArtRo walking forward (every sixth frame).

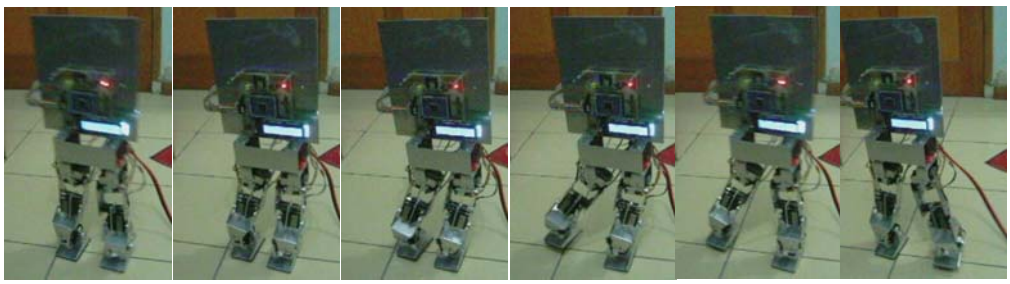


Figure 15. Image sequence extracted from video of ArtRo walking to the side (every sixth frame).

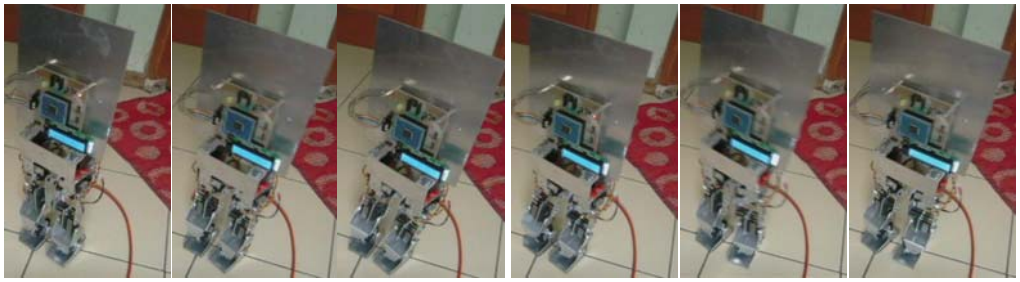


Figure 16. Image sequence extracted from video of ArtRo turning on the spot (every sixth frame).

To evaluate the performance of the system, several tests are conducted, by putting various combinations of values at the top layer behavior, that is at the motion vector $\mathbf{a}_{Robot}$, observe the resulting walking behavior, and measure the walking speed for each combination. For motion vector $\mathbf{a}_{Robot} = (0,0,0)$, the result is walking on the spot behavior (walking speed *0.0* cm/s). While $\mathbf{a}_{Robot} = (0,0.07,0)$, ArtRo is walking forward, as depicted in Figure 14, with estimated measured speed of *3.8* cm/s. While $\mathbf{a}_{Robot} = (0.1,0,0)$, walking to the side behavior is resulted (Figure 15), with estimated measured speed of *3.5* cm/s. While $\mathbf{a}_{Robot} = (0,0,0.2)$, turning on the spot behavior is resulted (Figure 16), with estimated measured turning speed of *12.1*°/s. The video, from which Figure 14, 15, and 16 were extracted, was recorded by using video recorder with speed of 30 frames per second (fps), and all the above walking speed measurement estimations were derived from this video, by stepping frame-by-frame at the timeline, paying attention to the walking phases completion, and estimating the robot's displacement amplitude.

From observations, the omnidirectional walking scheme worked pretty fine, but sometimes it is disturbed by the occurance of motion slip between the base of the foot plates and the floor. Mostly the motion slip occured when the robot was doing turning on the spot walking behavior. This disturbance possibly caused by the lack of upper body weight. When the upper body parts (i.e. the trunk, arms, and head) of the robot has been manufactured and integrated to the lower body parts, this disturbance shall be reduced.

6. Conclusion and Future Research

A. Conclusion

This paper described the design and construction of low-cost but reliable mechanical and electrical system of ArtRo bipedal robot, as well as the implementation and adaptation of the online trajectories generation for omnidirectional bipedal walking technique, which is introduced by Sven Behnke in [5], on it.

At the walking technique implementation and testing stage, ArtRo is a bipedal robot consisting of two legs, lower abdomen, and a dummy weight representing the upper body load, with a total of 12 DOF. The mechanical system design has been optimized to facilitate the manufacturing process and the computation of walking parameters of the robot.

The electrical system consists of main controller and actuator system. The main controller is an Atmel's ATmega128-based microcontroller system. The actuator system consists of Pulse Width Modulation (PWM) signal-controlled servo motors and its driver circuit. Using a special method involving Time-Division Multiplexing (TDM), 30 independent PWM signals are generated from only six 16-bit PWM signal channels available on the main controller, which are more than enough to control all servo motors on ArtRo.

To implement omnidirectional walking, actuator system of ArtRo is coordinated by hierarchical reactive behaviors control architecture. The architecture consists of four layers, from the top layer to the bottom layer, they are motion vector, walking phase, leg parameter, and individual joint angle, respectively. The target walking direction, magnitude, and rotation are specified within the top layer, that is the motion vector, and then translated into the lower layers until the bottom layer, that is the individual joint angle, so that every joints of ArtRo are actuated in a coordinated way, to achieve the target walking behavior.

This technique has been successfully implemented on ArtRo's main controller, which is an 8-bit microcontroller-based system, proving that it has a low computational complexity. Even though there is mechanical design difference between humanoid robot Jupp in [5] and ArtRo, that is in the placement of the leg-yaw concrete actuator, adaptation in the way of leg parameters computation and adjustment of some walking phase parameters have been done successfully in the implementation of the technique on ArtRo. On the tests, in spite of the motion slip disturbance between the base of the foot plates and the floor caused by the lack of upper body weight, ArtRo has been able to do walking on the spot, walking forward, walking to the side, and turning on the spot, depend on the parameter set value in the motion vector. This shows that by the implementation of the technique, ArtRo is capable to do omnidirectional walking.

B. Future Research

A more stable walking behavior can be achieved when ArtRo has upper body parts, that are the trunk, arms, and head. Beside of the role of upper body parts in reducing the motion slip disturbance between the base of the foot plates and the floor, each arm can be programmed to move synchronously with its contralateral leg, to improve walking stability.

Also, there is possibility to combine the bipedal walking technique, which is introduced by Sven Behnke in [5], with Zero-Moment Point (ZMP)-based control scheme. For ZMP-based control scheme, force sensors need to be placed at several points at the base of the foot plates. ZMP is then computed for each legs, based on force measured by each of the force sensors while walking, using Equation (16.12) in [8]. The main purpose of the control scheme is to maintain the ZMP to be at the center of the foot plate for ground-contacting foot/feet at any time. If the ZMP is not at the center of the foot plate, error-correction value is added proportionally, derivatively, and/or integratively (PID) to some parameters in walking phase layer of the actuator hierarchy of reactive behaviors, so that the ZMP can be corrected to the center of the foot plate while ArtRo is walking. This control scheme may improve dynamic balance of the walking behavior.

7. Acknowledgment

The author would like to thank Dody Suhendra for financial support provision, Sapto Adi Nugroho and Nico Prayogo from CentrUMS ITB as mechanical design and manufacturing advisors, Danar Tri Yurindatama and Nur Muhammad Malikul Adil as ArtRo's first mechanical designers, Hao-Che Chen from Tamkang University, Taiwan, as bipedal walking technique advisor, and Sven Behnke, who presented the bipedal walking technique that is

implemented in this work. This work was supported by PT TELKOM Indonesia, PT PLN Indonesia, and Institut Teknologi Bandung (ITB).

8. References

[1] M.H.P. Dekker, "Zero-Moment Point Method for Stable Biped Walking," Internship Report, Eindhoven, pp. 10, July 2009. URL: www.mate.tue.nl/mate/pdfs/10796.pdf.
[2] K. Erbatur, U. Seven, "An Inverted Pendulum Based Approach to Biped Trajectory Generation with Swing Leg Dynamics," *Proceedings of 7th IEEE-RAS International Conference on Humanoid Robots*, Pittsburgh, PA, 2007.
[3] T. McGeer, "Passive Walking with Knees," *Proceedings of IEEE International Conference on Robotics and Automation*, Cincinnati, USA, 1990.
[4] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient Bipedal Robots Based on Passive-Dynamic Walkers," *Science vol. 307*, pp. 1082-1085, 2005.
[5] S. Behnke, "Online Trajectory Generation for Omnidirectional Biped Walking," *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, Florida, 2006.
[6] _____, "Sagittal Plane," accessed at November 20th, 2010. URL: http://en.wikipedia.org/wiki/Sagittal_plane.
[7] S. Behnke and R. Rojas, "A Hierarchy of Reactive Behaviors Handles Complexity," in *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*. Springer, 2001, pp. 125-136.
[8] Siciliano and O. Khatib, *Springer Handbook of Robotics*, Berlin Heidelberg, Germany: Springer-Verlag, 2008, pp. 372-373.

Giovanni Sutanto, He graduated from his Bachelor degree program in Electrical Engineering at Bandung Institute of Technology (ITB), Indonesia in July 2010. He is currently working as Robotics Engineer at Asea Brown Boveri (ABB) Indonesia. His special fields of interest included robotics, control engineering, artificial intelligence, and computer vision.

Kusprasapta Mutijarsa, He graduated his Doctoral program from Institut Teknologi Bandung. His employment as a lecturer and researcher in School of Electrical Engineering and Informatics ITB. He is a head of Autonomous Vehicle Research Group SEEI ITB. His special fields of interest included artificial intelligence, robotics, and autonomous system.

Hilwadi Hindersah, He graduated his Doctoral program from Institut Teknologi Bandung. His employment as a lecturer and researcher in School of Electrical Engineering and Informatics ITB. His fields of interest are vision-based robot control, design interaction, and adaptive control.