MOTION TRAJECTORY GENERATION BASED ON POLYNOMIAL METHOD FOR HUMANOID ROBOT

Jurnal

Diajukan Sebagai Salah Satu Persyaratan

Mencapai Derajat Sarjana

Oleh:

Vivy Fauziah
1800022023

PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS AHMAD DAHLAN

YOGYAKARTA

2023

HALAMAN PERSETUJUAN

Jurnal

MOTION TRAJECTORY GENERATION BASED ON POLYNOMIAL METHOD FOR HUMANOID ROBOT

Yang diajukan oleh :

Vivy Fauziah

1800022023

Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Ahmad Dahlan

Telah disetujui untuk submit ke jurnal Prodi oleh :

Pembimbing,

Nuryono Satya Widodo, S.T., M.Eng.   12 Juni 2023

Motion Trajectory Generation Based on Polynomial Method for Humanoid Robot

Vivy Fauziah [1], Nuryono Satya Widodo,. S.T.,M.Eng [2]
[1,2]Teknik Elektro, Universitas Ahmad Dahlan, Jl. Jl. Ringroad Selatan, Daerah Istimewa Yogyakarta

ARTICLE INFO

Article history:

Received
Revised
Published

Keywords:

Humanoid Robot;
Robot Soccer;
Motion Generation;
Trajectory;
kinematics;

ABSTRACT

Humanoid Robot is a robot whose shape and appearance resembles the human body and is equipped with technology so that it can carry out basic functions like humans in general. Robot Soccer Universitas Ahmad Dahlan (R-SCUAD) is one of Robotic Teams that competes on Indonesian Humanoid Robotic Soccer Competition. The focus of the competition is on developing humanoid robots that can play soccer, and each robot is equipped with sensors, actuators, and other supporting technologies to recognize the playing field, identify the ball, perform basic movements such as walking, kicking, standing up, communicating with teammates, and making decisions during the game.

One of the challenges in robot development is generating smooth and continuous movements. The currently used method of manual movement generation by the robot operator is often unsuccessful due to a lack of precision and consistency. Robots often fail to generate movements because the movements that have been recorded are not played coherently.

To address this issue, this research propose a trajectory generation method based on fourth-degree polynomial functions that allows for more precise control over the movement path. The operator can simply provide the robot with a target position and orientation, and the trajectory generates the necessary movement to reach the target. The main focus is on the robot's ability to generate kicking movements using this trajectory generation method. Results show that the robot successfully generated left, right, and side kicks, with an average time of 5.334 seconds for left kicks, 6.099 seconds for right kicks, and 3.88 seconds for side kicks, based on ten trials.

Corresponding Author:

Nuryono Satya Wiodo, Teknik Elektro, Universitas Ahmad Dahlan, Jl. Ringroad Selatan, D.I Yogyakarta , Indonesia
Email: nuryono.sw@uad.ac.id

1. INTRODUCTION

Humanoid Robot is a robot whose shape and appearance resembles the human body and is equipped with technology so that it can carry out basic functions like humans in general[1]. In a soccer game between humanoid robots, the robot is required to be able to walk dynamically, run and kick the ball while maintaining balance, visual perception of the ball and the pattern of the field, self-localization and teammates and recognizing opponents[2][3]. RoboCup is a humanoid robot football competition in the world, while in Indonesia itself, there is the KRSBI-Humanoid (Kontes Robot Sepakbola Indonesia - Humanoid).

Robot Soccer Universitas Ahmad Dahlan (R-SCUAD) is a robot team from Ahmad Dahlan University that conducts research and development in the field of humanoid soccer robots. Currently, R-SCUAD has five robots with capabilities such as ball detection and recognition, field image recognition, dynamic walking, ball kicking, ball chasing, getting up from a fall, and other abilities. The technology used by R-SCUAD undergoes continuous development every year to optimize the robots' potential and prepare them for participation in the Indonesia Robot Contest. The robot's movements are created using RoboPlus software. RoboPlus is a software developed by ROBOTIS that facilitates interaction with all ROBOTIS hardware devices, including control, dynamixel, sensors, and other components.

The current R-SCUAD robots have implemented several methods for generating movements. These include parametric walking motion generation using inverse kinematics and sinusoidal methods, as well as manual generation of basic motions using the record and play method. In the inverse kinematics method, motion generation is performed by analyzing the angle coordinates of the servos used within the robot's range of motion [4][5][6]. This method involves trial and error to achieve precise movement goals. The drawback of this method is the time-consuming process and significant memory usage[7].

On the other hand, in the record and play method, motion generation is based on the existing servos[8][9]. Each servo has its initialization with a unique ID number. Since the framework used is from Robotis, the robot comes equipped with basic motions like "Hi," "Bye," and "Talk". In the Indonesia Robot Contest, several specific motions are required for the robot. Meanwhile, other motions such as getting up when the robot falls, body rotation, arm extension for the goal keeper robot, and body dropping and kicking motions are still created using the record and play method. This involves recording the angle values of the servo's tilt and then calling the motion using motion commands in the main program[10]. The drawback of this method is that it takes a relatively long time, which can cause the servos to overheat and affect their performance[11].

Generating humanoid robot motion trajectories based on polynomials can be a solution to overcome constraints related to short and asymmetric movements[12]. Issues such as time-consuming motion calling often occur in short-distance movements[13][14]. This algorithm is proposed to generate smoother motion on trajectories defined by polynomial functions by combining two arbitrary conditions determined by the position of each joint[15][16].

2. METHODS

The research was conducted using a humanoid robot, R-SCUAD, which moves using 12 MX-64 servo motors and 8 MX-28AR servo motors. The robot moves with OpenCM9.04 as the sub-controller, Odroid XU-4 as the central controller, and Gyroscope and Accelerometer as the robot's balance sensors. The robot's design is shown in **Fig 1.** The robot's framework is made of stainless steel with a thickness of 2.5mm for the robot's body, carbon fiber material with a thickness of 1.5mm for the robot's foot sole and 1mm for the robot's hand, acrylic with a thickness of 5mm for the robot's head support, and nylon with a thickness of 2mm as the robot's body protector.

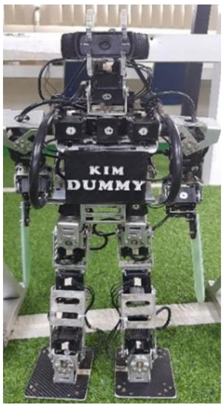


Fig 1. The Robot

The size of the robot's framework greatly affects every movement it performs. The dimensions of the robot are presented in **Table 1.**

Table 1. Robot Dimension

Dimension	Unit (cm)
Height	62
Hleg	28,5
Hhead	14,5
The width of the robot when the arms are stretched	69
The width of the foot sole	8
The length of the foot sole	12
The length of the thigh links	11,5
The length of the knee links	12,5
The length of the foot sole link	4,2

The robot is actuated by 20 servo motors distributed throughout its body. Each servo is assigned a unique ID number during initialization, ensuring accurate data transmission and information retrieval between the servos during motion control. The detailed servo numbering is presented in **Table 2.**

Table 2. Servo Numbering

Servo ID	Location
1	Right Shoulder
2	Left Shoulder
3	Right Arm
4	Left Arm
5	Right Elbow
6	Left Elbow
7	Right Waist
8	Left Waist
9	Right Back Thigh
10	Left Back Thigh
11	Right Front Thigh
12	Left Front Thigh
13	Right Knee
14	Left Knee
15	Right Front Ankle
16	Left Front Ankle
17	Right Back Ankle
18	Left Back Ankle
19	Neck
20	Head

2.1. The Robot Control sistem

Fig 2 shown the system of the utilized robot, wherein the servo angle coordinates are captured through Gyroscope and Accelerometer sensors located on the sub-controller board. The data is subsequently processed on the ODROID XU-4 as the central controller. The processed data is then transmitted to the OpenCM9.04 as the sub-controller, which further drives the actuators.

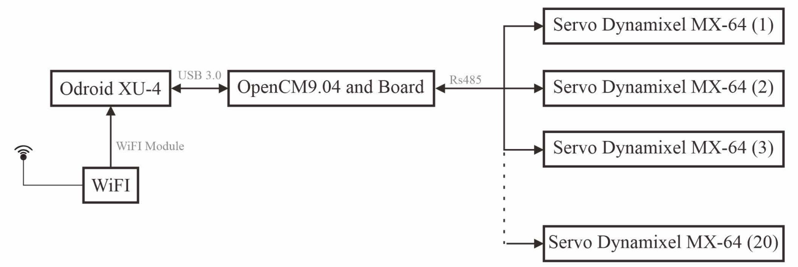

Odroid XU-4
USB 3.0
OpenCM9.04 and Board
Rs485
WiFI Module
WiFI
Servo Dynamixel MX-64 (1)
Servo Dynamixel MX-64 (2)
Servo Dynamixel MX-64 (3)
Servo Dynamixel MX-64 (20)

Fig 2. Block Diagram

Every planned process in the research can be observed through the flowchart in **Fig 3.** Once the robot reaches the desired position (close to the ball), the robot operator will initiate the selection of a specific action. Button 1 is used for a left kick, button 2 for a right kick, and button 3 for a side kick. Subsequently, the robot will stop after completing its task.

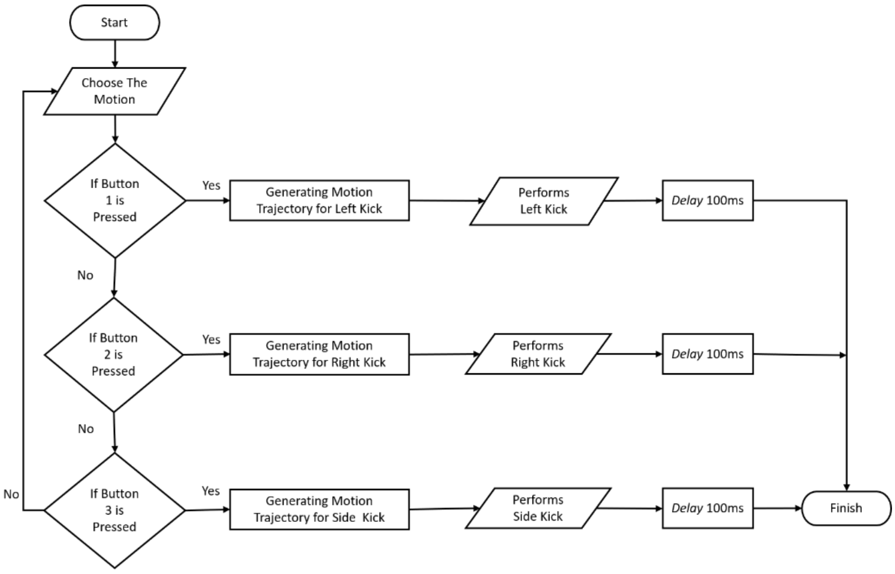


Fig 3. Flowchart

2.2. End effector and Trajectory Generation

In order to develop movement patterns for the robot, it is essential to conduct an analysis of its legs to determine the range of motion and rotation that can be achieved. This analysis is crucial to ensure that the robot's movements are within the desired limits and are stable and precise[17][18]. **Fig 4.** shown diagram of the robot's leg is provided, with a yellow straight line representing the x-axis and a red line representing the z-axis[19]. The direction of rotation for the robot's leg is established as the z-axis. The robot's leg is designed with six degrees of freedom, which are allocated to each joint, including hip pitch, hip yaw, hip roll, knee pitch, ankle pitch, and ankle roll[20][21]. This journal emphasizes the importance of analyzing the robot's legs and joints in the development of movement patterns and ensuring the robot's stability and accuracy in its movements.

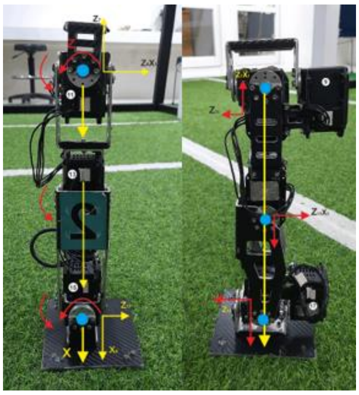


Fig 4. Rotation point on Robot Leg

Thus, **Table 3**. shown the coordinates of each frame that is connected to the servo dynamixel and the joints contained in the robot, each representing the angle of rotation (θ) and the range of motion that can be performed[19]. The Denavit-Hartenberg (D-H) table is a method used in robotics to model and describe the relationships between various joints in a robot. This table utilizes four parameters known as the Denavit-Hartenberg parameters. These parameters are as follows:

α : Twist angle
The twist angle represents the angle between the z_i-1 axis and the z_i axis, measured along the x_i axis. Geometrically, it is the rotation angle between the i-1th revolute axis and the ith revolute axis around the x_i axis.

a : Link length
The link length refers to the distance between the z_i-1 axis and the z_i axis, measured along the x_i axis. Geometrically, it is the distance between the i-1th revolute axis and the ith revolute axis, which are mutually perpendicular.

d : Link offset
The link offset represents the distance between the x_i-1 axis and the x_i axis, measured along the z_i-1 axis. Geometrically, it is the distance between the i-1th revolute axis and the ith revolute axis, which are parallel to each other

θ : Sudut *joint*
The joint angle describes the rotation between the x_i-1 axis and the x_i axis, measured along the z_i-1 axis. Geometrically, it is the rotation angle between the i-1th revolute axis and the ith revolute axis around the z_i-1 axis

In this process, the desired movement trajectory is created based on a polynomial function to generate movement from the initial position to the final position (destination point) [22]. Before carrying out the movement, the set point value, the intermediate angle position, and the continuation position and angle in generating a movement must be calculated in advance so that the resulting movement can be smoother[23][24]. The trajectory is designed using the 4 degree polynomial method where the 4th degree polynomial function represents the process initial to lift-off, set-down to the final position.

$$p = a_1t^4 + a_2t^3 + a_3t^2 + a_4t + p_0 \tag{1}$$

In this process, the problem of saving the robot's last movement when kicking must be solved. The tilt angle of the robot's body and legs will be used as a parameter so that the robot can generate movements according to the angle accurately and in a balanced position[19][25]. This problem can be solved using the rules of the 4x4 transformation matrix with the equation is as follow :

$$T = \begin{pmatrix} \cos\theta & -\sin\theta^{*}\cos\alpha & \sin\theta^{*}\sin\alpha & a * \cos\theta \\ \sin\theta & \cos\theta * \cos\alpha & -\cos\theta * \sin\alpha & a * \sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

The matrix is presented in 4x4 form as a representation of the position of the joints, and the orientation of the end-effector on the robot's legs mathematically. This matrix consists of a 3x3 rotation matrix, namely the orientation of the end-effector in space, and a 3x1 translation vector, namely the position of the end-effector relative to the reference. The last row [0 0 0 1] is used to maintain the homogeneity of the matrix.

2.3. MotionTime Step

The speed of motion invocation is measured based on the angular values generated divided by the time taken by each joint to reach each position, from the initial pose to the final position. The position values are then converted and expressed in degrees (°) using Equation (3).

$$x = \text{value in position } (360/\text{Vmax}) \tag{3}$$

Where:

X = angle value of the servo in degrees
Vmax = maximum position value (4096)

2.4. Motion Time Step comparison

The dynamic factor in this study is measured by dividing the speed of motion invocation by a unit of time. The result is then compared with the previous method, which is the record and play method. The speed of motion invocation from the record and play method is compared with the polynomial-based trajectory method using Equation (4), which calculates the average.

$$\text{Average} = (\text{Sum of Invocation Time}) / (\text{Number of data}) \tag{4}$$

3. RESULTS AND DISCUSSION

The testing was conducted in the Robotics Laboratory of the Electrical Engineering Department, 1st Floor, Campus 4, Ahmad Dahlan University, Yogyakarta. The robot was placed on a synthetic grass carpet, and the motion execution was performed by invoking pre-programmed movements through a dedicated software, developed specifically for the purpose of the experiment.

3.1 End Effector and Motion trajectory

Table 2. shown the Denavit Hartenberg on robot right leg.

Table 3. Denavit Hartenberg Table

Joints	Transformation Matrix			
	α	a	d	θ
Hip yaw	+90	0	d1	θ 1 + 90
Hip Roll	-90	0	0	θ 2 - 90
Hip Pitch	0	L1	0	θ 3
Knee Pitch	0	L2	d4	θ 4
Ankle Pitch	+90	0	d5	θ 5
Ankle Roll	0	L3	0	θ 6

From **Table 2**, it can be observed that several variables have a value of 0. Note that for the “a” variable on Hip yaw and Hip Roll are two opposing joints but have different joint configurations, as illustrated in **Fig 5.**

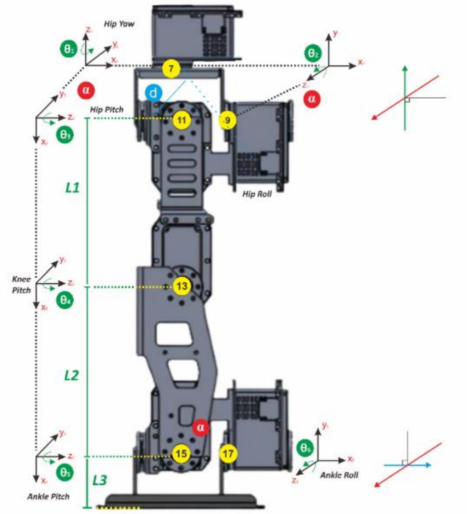


Fig 5. Joints, Links,and Rotation

The yellow circles in **Fig 5.** represent the joints on the robot leg, with corresponding numerical labels indicating the ID of the Dynamixel servo used. Theta (θ) denotes the angle at the joint around the rotational axis (z). Hip Yaw (servo ID 7) moves left and right within the horizontal plane with a range of +90 degrees (relative to Z0), while Hip Roll (servo ID 9) tilts sideways within the vertical plane by -90 degrees. The parameter α takes a value of 0 when the rotational axes of links i and i+1 do not intersect or align. There are three main links in the robot leg: from hip to knee, knee to ankle, and ankle to foot (with the foot base shaped like a square). The variable a represents the link length. Hip Pitch (servo ID 11) and Knee Pitch (servo ID 13) are connected by L1, Knee Pitch (servo ID 13) and Ankle Pitch (servo ID 15) are connected by L2, and Ankle Roll (servo ID 17) is connected to Ankle Pitch (servo ID 15) with the help of an H-shaped stainless structure at the front, which then extends directly to the foot using L3. The variable d is 0 because the rotational axis (z) intersects with the translational axis. The following is the process of determining the homogeneous transformation matrix of each joint using the formula in equation (2) :

$$T1 = \begin{pmatrix} cos\,\theta_1 + 90 & (-\sin\theta_1 + 90) & 0 & (90)(cos\,\theta_1 + 90) \\ sin\,\theta_1 + 90 & -(cos\,\theta_1 + 90) & 0 & (90)(sin\,\theta_1 + 90) \\ 0 & 0 & 1 & d1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T2 = \begin{pmatrix} cos\,\theta_2 - 90 & (-\sin\theta_2 - 90) & 0 & (-90)(cos\,\theta_2 - 90) \\ sin\,\theta_2 - 90 & -(cos\,\theta_2 - 90) & 0 & (-90)(sin\,\theta_2 - 90) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T3 = \begin{pmatrix} \cos\theta_3 & -\sin\ \theta_3 * \cos\alpha_3 & \sin\theta_3 * \sin\alpha_3 & 0 \\ \sin\theta_3 & \cos\theta_3 * \cos\alpha_3 & -\cos\theta_3 * \sin\alpha_3 & 0 \\ 0 & \sin\alpha_3 & \cos\alpha_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T4 = \begin{pmatrix} \cos\theta_4 & -\sin\ \theta_4 & 0 & 0 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T5 = \begin{pmatrix} \cos\theta_5 & -\sin\ \theta_5 & 0 & 90 * \cos\theta_5 \\ \sin\theta_5 & \cos\theta_5 & 0 & 90\ * \sin\theta_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T6 = \begin{pmatrix} \cos\theta_6 & -\sin\ \theta_6 * \cos\alpha_6 & \sin\theta_6 * \sin\alpha_6 & 0 * \cos\theta_6 \\ \sin\theta_6 & \cos\theta_6 *\ \cos\alpha_6 & -\cos\theta_6 * \sin\alpha_6 & 0 * \sin\theta_6 \\ 0 & \sin -\frac{\pi}{2} & \cos -\frac{\pi}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The motion is generated using a fourth-degree polynomial-based trajectory method. **Tables 4.** and **5** below show the positions of each joint during the left kick motion, expressed in degrees (^{0}). The advantage of using a fourth-degree polynomial to generate the motion is the ability to smooth between key points and its computational efficiency. However, it may not be suitable for all types of trajectories, especially those with complex geometries or multiple obstacles. In this case, other mathematical functions or motion planning algorithms can be used.

```
    double a1 = j0/24;
    double a2 = a0/6 - j0*pow(tf,2)/120 - jf*pow(tf,2)/24 - af*tf/12 + j0*tf/6 + 
jf*tf/4;
    double a3 = 0.5*j0*pow(tf,3)/6 + jf*pow(tf,3)/6 + 0.5*a0*pow(tf,2) - 
0.5*af*pow(tf,2) + j0*pow(tf,2)/2 + jf*pow(tf,2)/2 + v0*tf - vf*tf/2 - a0*tf/6 + 
af*tf/6;
    double a4 = -j0*pow(tf,4)/24 - jf*pow(tf,4)/24 - a0*pow(tf,3)/6 + 
af*pow(tf,3)/6 - j0*pow(tf,3)/6 - jf*pow(tf,3)/4 - v0*pow(tf,2)/2 + 
vf*pow(tf,2)/4 + a0*tf/6 - af*tf/12 - p0 + pf;
```

The scipt above defines a function that takes the initial and final states of a single degree of freedom, including position, velocity, acceleration, and jerk at time zero and the final position, velocity, acceleration, and jerk at the end of the trajectory, as well as the total trajectory time. The formula in equation (3.2) is used to create the trajectory pattern and is added to Listing 4.1 as follows:

```
    for(double t = 0; t <= tf; t += 0.01)
    {
        double p = a1*pow(t,4) + a2*pow(t,3) + a3*pow(t,2) + a4*t + p0;
        traj.push_back(p); }
```

The coefficients a1, a2, a3, and a4 are calculated based on the initial and final states as well as the total trajectory time. These coefficients are used to generate trajectory points for the degree of freedom over time using a loop that iterates from zero to the total trajectory time (tf) with an increment of 0.01 seconds. The trajectory points are stored in a vector called traj, which is returned at the end of the function. **Table 4.** shows the joint positions (in degrees/0) in the fourth-degree polynomial.

Table 4. Left Kick Position (degree) with Fourth Degree Polynomial

Joints		Initial Value (º)	Lift Off (º)	Interval	Final Value (º)
hip yaw	R	180	178.869	200	181.854
	L	180	164.307	200	178.256
hip roll	R	180	184.422	200	188.141
	L	165	178.256	200	169.538
knee	R	125	121.231	200	123.744
	L	230	142.161	200	228.462
ankle pitch	R	230	156.993	200	230.458
	L	125	72.065	200	131.704
Ankle yaw	R	200	195.980	200	203.960

Joints		Initial Value (°)	Lift Off (°)	Interval	Final Value (°)
	L	160	196.382	200	162.010
Ankle roll	R	180	167.538	200	190.452
	L	180	152.261	200	178.995

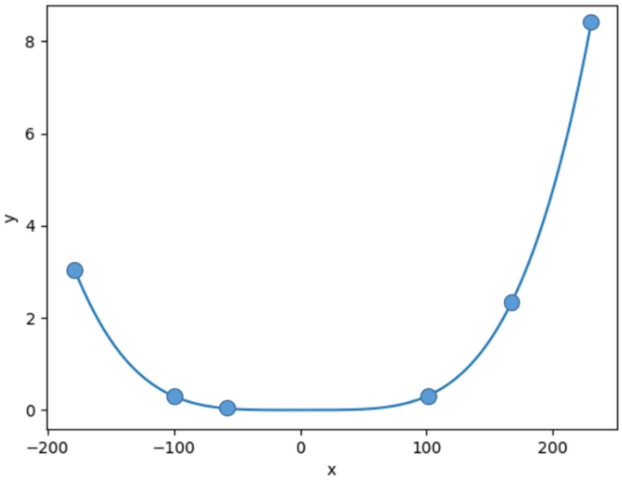


Fig 6. Polynomial Left Kick Graph

The initial value of the position can be calculated using Equation (3) to convert the servo position value (pose in) from the RoboPlus Motion software, which is presented as a value ranging from 0 to 4096, to degrees. For example, if the position value is 2048, the calculation for the angle value is as follows:

$$x = 2048\ (360 / 4096)$$
$$x = 180$$

Therefore, the servo position value of 2048, when expressed in degrees, is 180 degrees.

Table 5. Pose In Value with Record and Play

Joints		position (°)					
		step 0	step 1	step 2	step 3	step 4	step 5
hip yaw	R	182.988	180.000	180.000	180.000	177.363	183.113
	L	180.000	170.596	180.000	167.959	178.330	180.123
hip roll	R	182.109	187.207	187.207	187.207	185.977	186.544
	L	177.012	169.014	167.607	174.463	181.318	177.133
knee	R	124.541	123.311	124.541	122.432	129.199	124.626
	L	232.910	131.660	192.920	242.578	233.262	233.069
ankle pitch	R	229.834	229.834	141.943	229.834	230.889	229.991
	L	133.857	164.971	69.258	173.936	132.275	133.949
Ankle yaw	R	200.566	196.787	197.139	193.711	202.324	200.704
	L	159.697	175.781	85.078	204.873	151.875	159.807
Ankle roll	R	185.977	167.344	166.553	167.256	168.750	186.104
	L	175.693	148.184	153.984	172.705	172.969	175.814

The motion is manually created by adjusting the initial position of the robot to the position when the robot is in the walk-ready position. **Figures 7. to 11** below depict the trajectory patterns from the record and play method, with each figure representing one joint, as each joint has a different pattern.

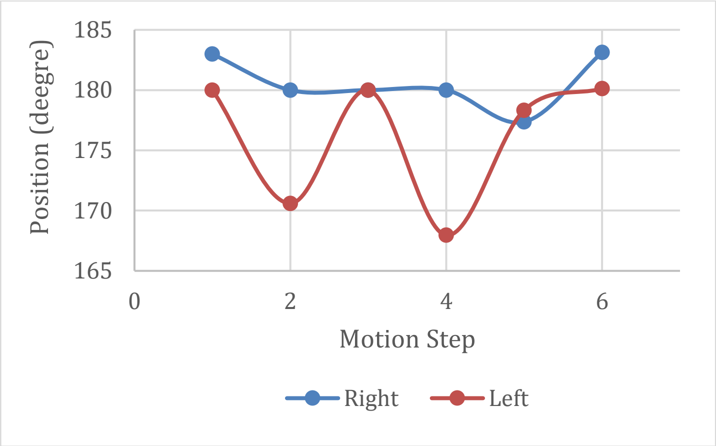


Fig 7. Left Kick – Hip Yaw

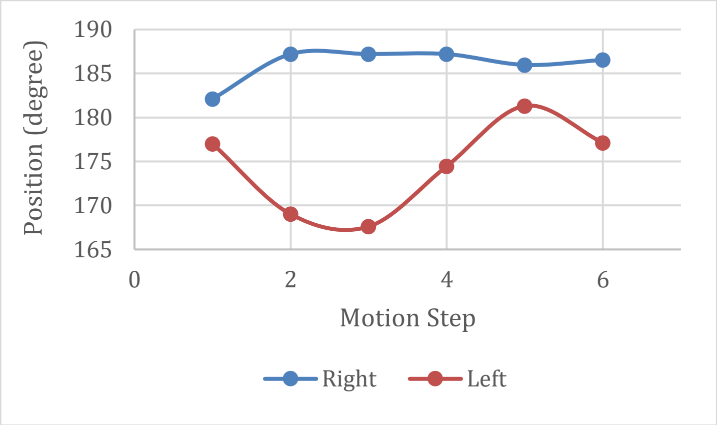


Fig 8. Left Kick – Hip Roll

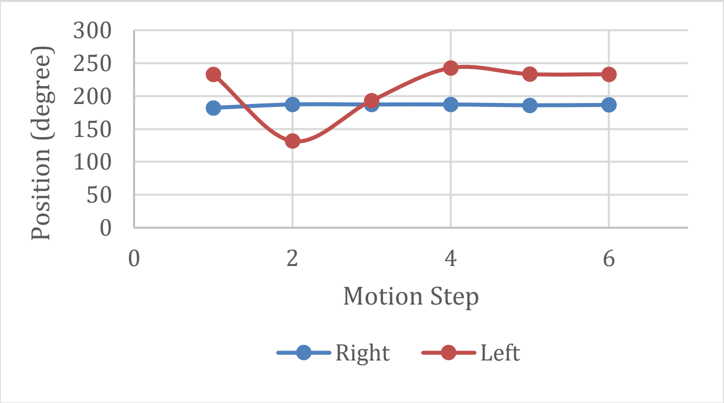


Fig 9. Left Kick – Hip Pitch

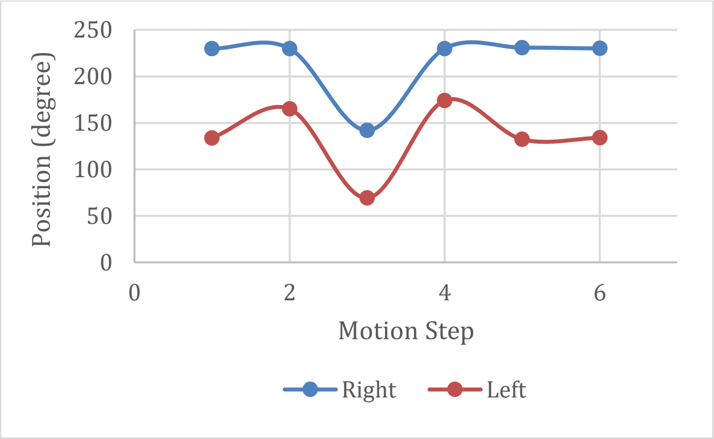


Fig 10. Left Kick – Ankle Pitch

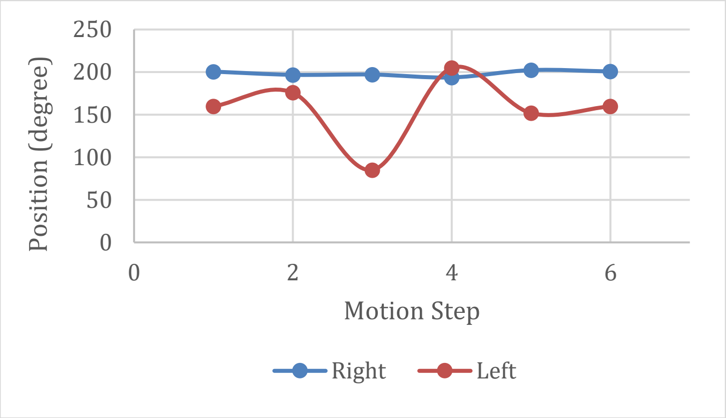


Fig 11. Left Kick – Ankle Yaw

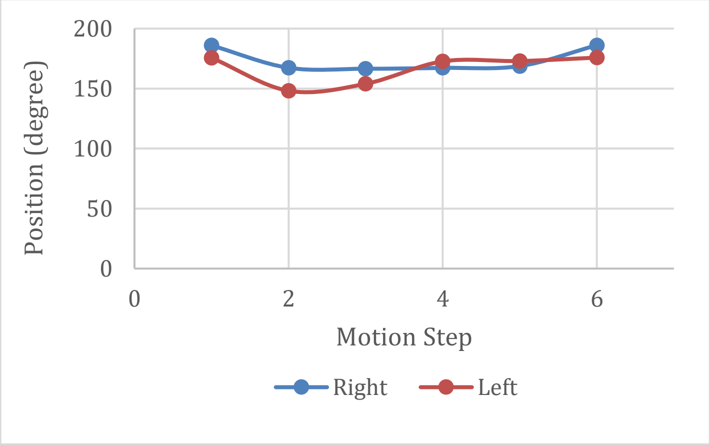


Fig 12. Left Kick – Ankle Roll

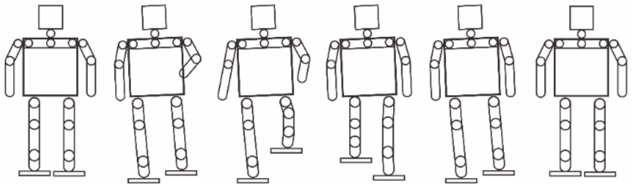


Fig 13. Left Kick – Front View

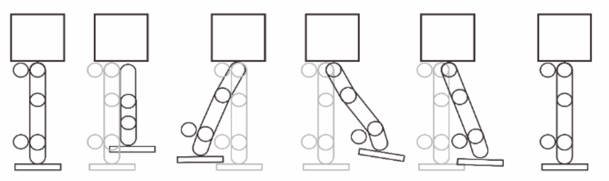


Fig 14. Left Kick – Side View

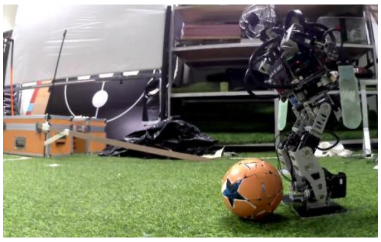


Fig 15. Walk Ready

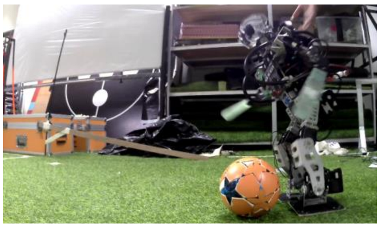


Fig 16. Lifting Left Leg

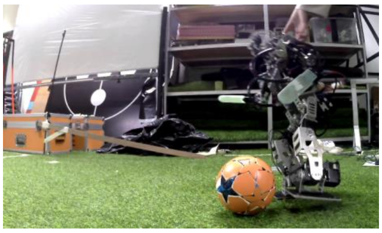


Fig 17. Swinging Left Leg

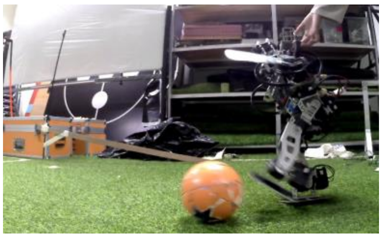


Fig 18. Kick The Ball

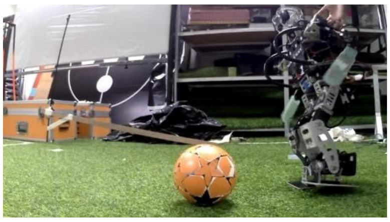


Fig 19. Swinging Left Leg

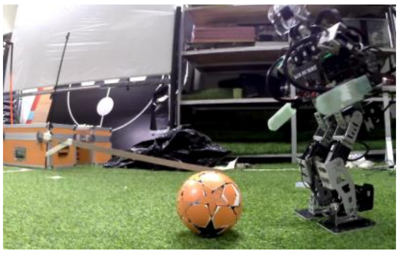


Fig 20. Back to Walk Ready

Table 6. Right Kick Position (degree) with Fourth Degree Polynomial

Joints		Initial Value (º)	Lift Off (º)	Interval	Final Value (º)
hip yaw	R	180	175.176	200	198.090
	L	180	180.000	200	180.000
hip roll	R	180	190.452	200	184.724
	L	165	173.266	200	180.905
knee	R	125	243.518	200	131.156
	L	230	211.106	200	228.392
ankle pitch	R	230	300.180	200	229.482
	L	125	129.316	200	136.528
Ankle yaw	R	200	262.337	200	192.621
	L	160	151.753	200	162.170
Ankle roll	R	180	201.436	200	183.806
	L	180	197.429	200	177.396

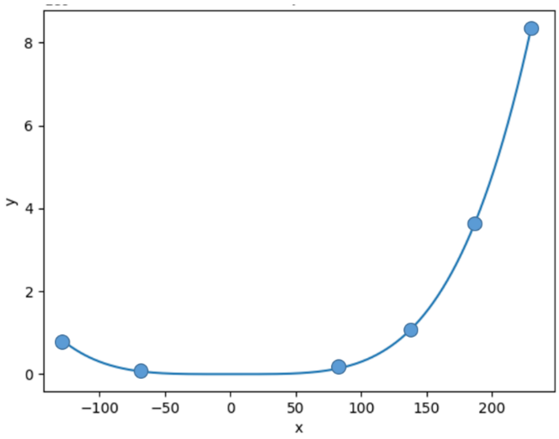


Fig 21. Polynimial Right Kick Graph

Table 7. Pose In Value with Record and Play

Joints		position (º)					
		step 0	step 1	step 2	step 3	step 4	step 5
hip yaw	R	182.988	180.000	180.000	180.000	177.363	183.113
	L	186.152	180.000	181.318	189.141	185.889	186.152
hip roll	R	180.000	180.000	180.000	180.000	180.000	180.000
	L	182.021	186.416	187.559	187.559	183.603	182.021
knee	R	177.012	174.902	176.836	176.924	171.914	177.891

Joints		**position (°)**					
		step 0	**step 1**	**step 2**	**step 3**	**step 4**	**step 5**
	L	125.684	129.727	243.105	122.871	123.750	129.111
ankle pitch	R	234.932	232.910	228.252	222.012	238.184	232.910
	L	225.967	234.844	288.369	181.143	232.910	225.967
Ankle yaw	R	133.857	124.980	131.836	127.002	127.881	133.857
	L	199.951	205.752	270.967	143.789	199.424	196.611
Ankle roll	R	158.643	153.633	157.412	154.600	158.906	159.697
	L	185.977	204.785	177.012	188.965	202.764	185.977

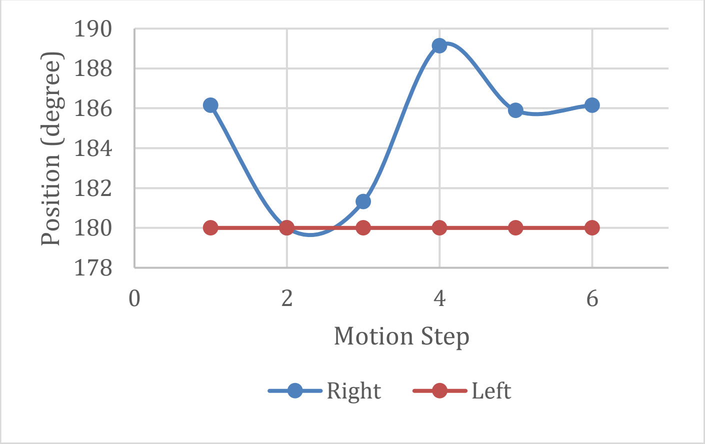


Fig 22. Right Kick – Hip Yaw

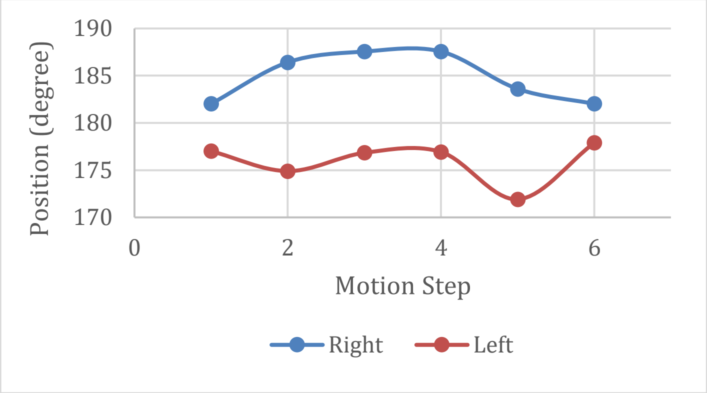


Fig 23. Right Kick – Hip Roll

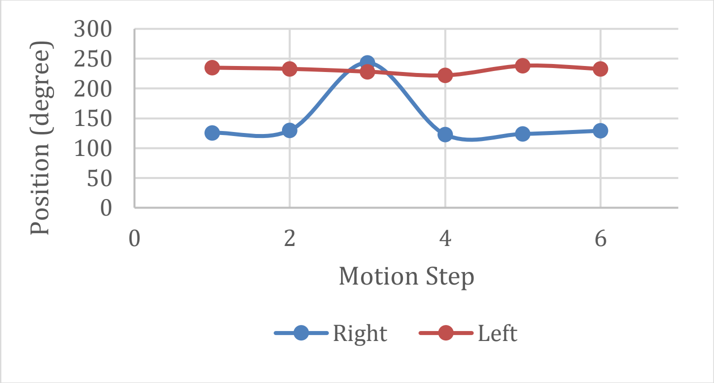


Fig 24. Right Kick – Hip Pitch

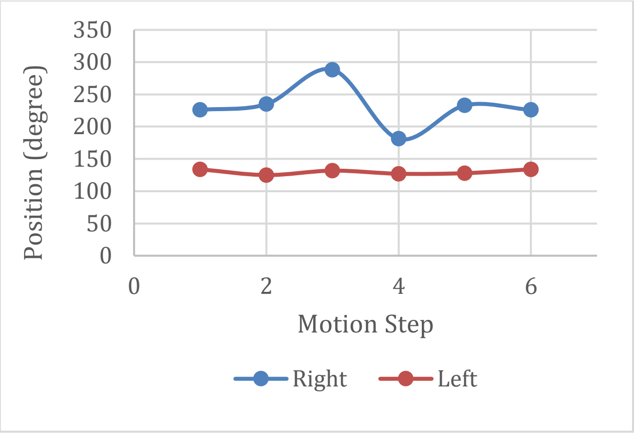


Fig 25. Right Kick – Knee Pitch

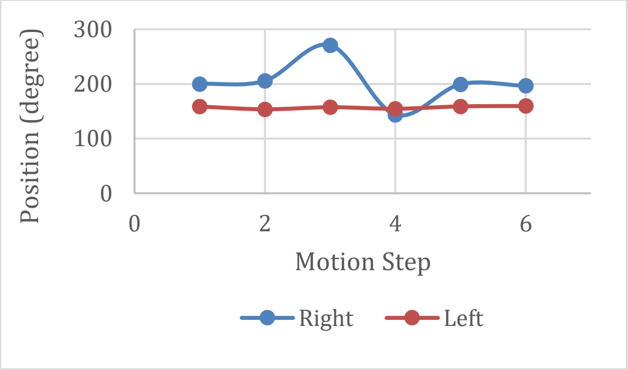


Fig 25. Right Kick – Ankle Pitch

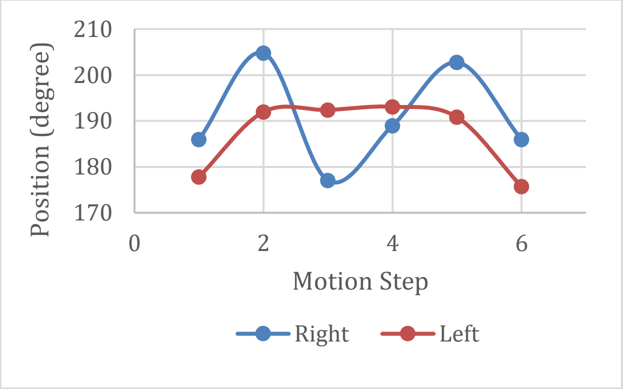


Fig 27. Right Kick – Ankle Roll

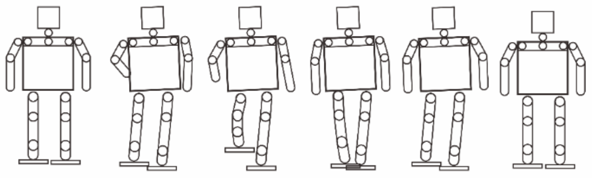


Fig 28. Right Kick – Front View

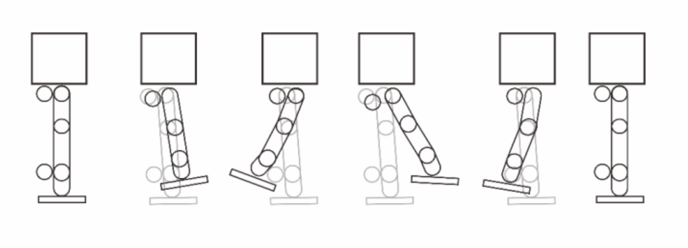


Fig 29. Right Kick – Side View

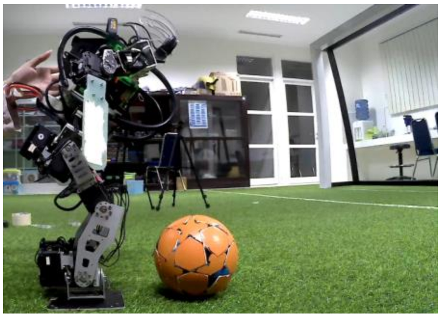


Fig 30. Walk Ready

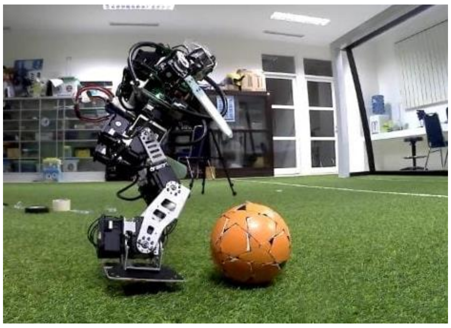


Fig 31. Stand on One Leg

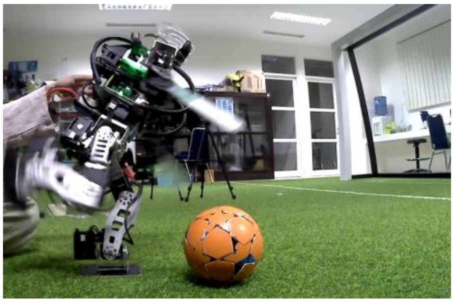


Fig 32. Swinging The Right Leg

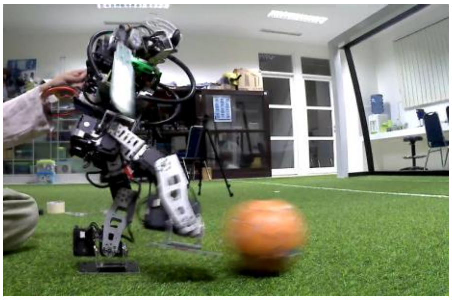


Fig 33. Kick The Ball

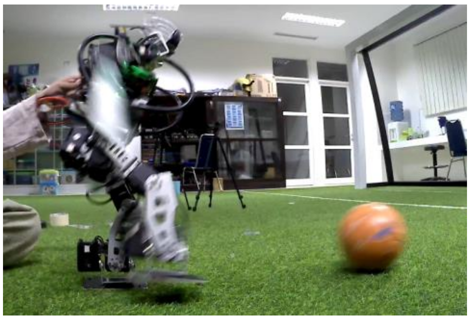


Fig 34. Swinging The Leg

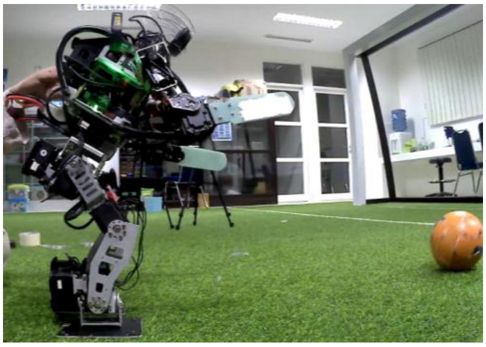


Fig 35. Back to Walk Ready

Table 8. Side Kick Position (degree) with Fourth Degree Polynomial

Joints		Initial Value (º)	Lift Off (º)	Interval	Final Value (º)
hip yaw	R	180	210.250	200	190.854
	L	180	180.000	200	180.000
hip roll	R	180	199.833	200	174.372
	L	165	98.063	200	180.553
knee	R	125	106.077	200	124.925
	L	230	230.284	200	199.095
ankle pitch	R	230	194.224	200	227.940
	L	125	129.316	200	129.045
Ankle yaw	R	200	145.342	200	203.216
	L	160	173.389	200	166.131
Ankle roll	R	180	203.038	200	190.854
	L	180	191.018	200	178.492

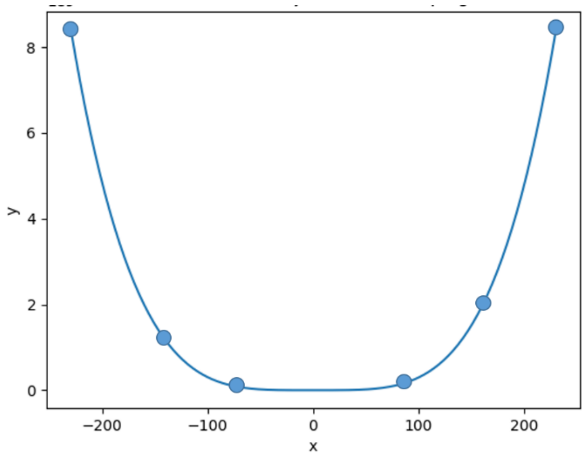


Fig 36. Polynomial Graph Side Kick

Table 9. Pose In Value with Record and Play

Joints		position (°)					
		step 0	step 1	step 2	step 3	step 4	step 5
hip yaw	R	186.152	182.900	202.061	177.012	185.889	186.152
	L	180.000	180.000	180.000	180.000	180.000	180.000
hip roll	R	182.021	193.887	211.025	180.000	183.603	182.021
	L	177.012	180.527	180.527	92.637	171.914	177.012
knee	R	129.111	129.111	108.984	108.984	122.959	129.111
	L	232.910	231.592	225.352	225.352	238.184	225.000
ankle pitch	R	225.967	229.482	182.812	182.812	232.734	225.967
	L	133.857	133.857	133.857	133.857	215.771	133.857
Ankle yaw	R	196.611	198.721	136.055	136.055	200.127	196.611
	L	159.697	161.543	161.543	161.543	158.906	159.697
Ankle roll	R	185.977	219.463	180.703	199.336	199.600	185.977
	L	175.693	193.008	194.766	193.008	188.877	175.693

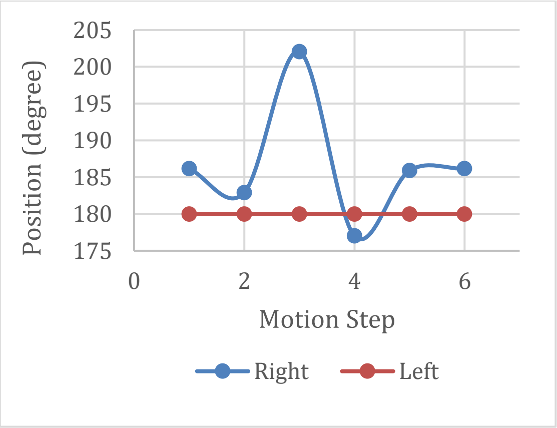


Fig 37. Side Kick – Hip Yaw

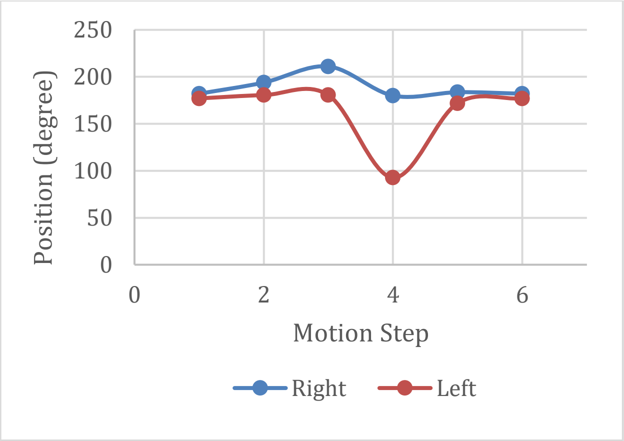


Fig 38. Side Kick – Hip Roll

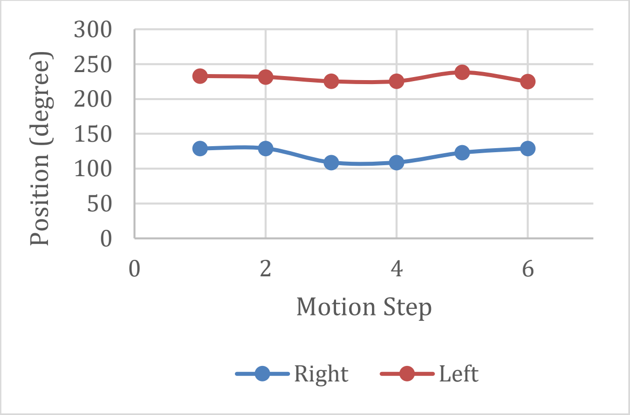


Fig 39. Side Kick – Hip Pitch

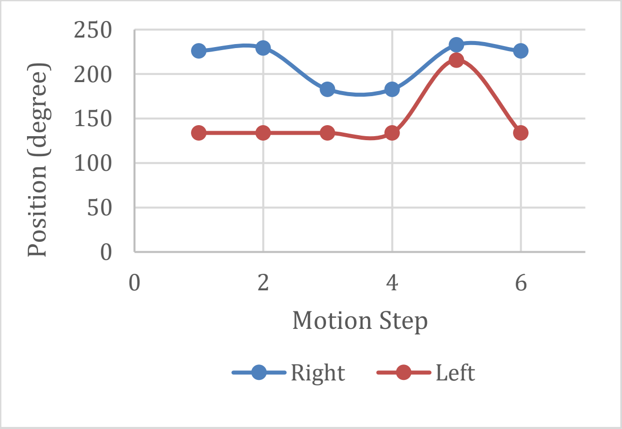


Fig 40. Side Kick – Knee Pitch

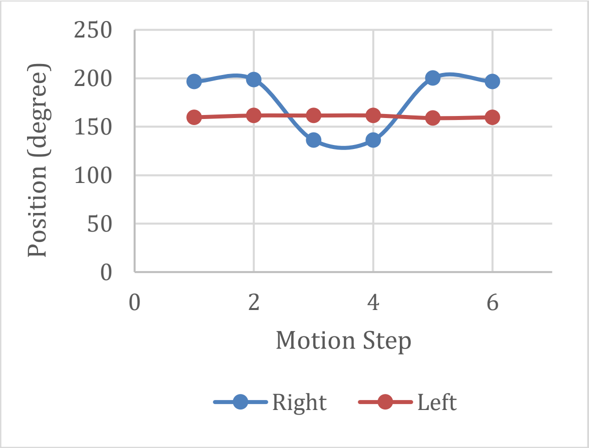


Fig 41. Side Kick – Ankle Pitch

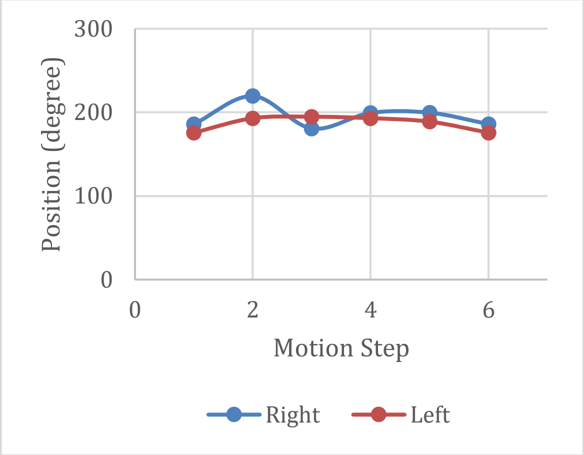


Fig 42. Side Kick – Ankle Roll

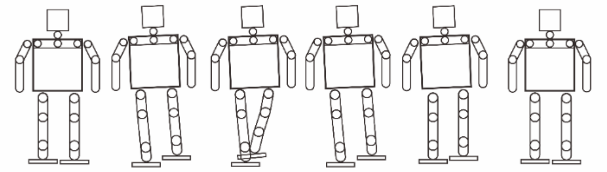


Fig 43. Side Kick Front View

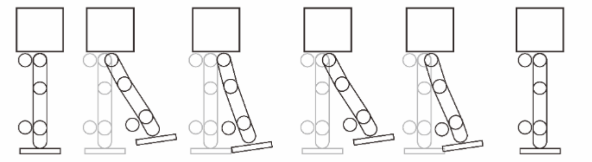


Fig 44. Side Kick Side View

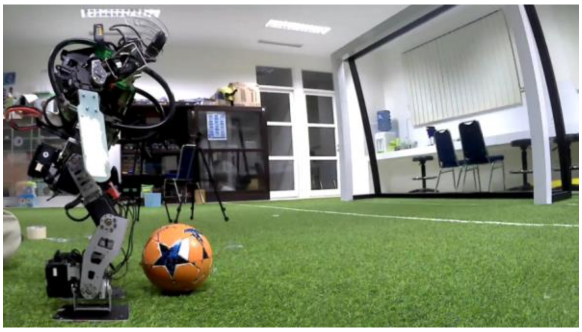


Fig 45. Walk Ready

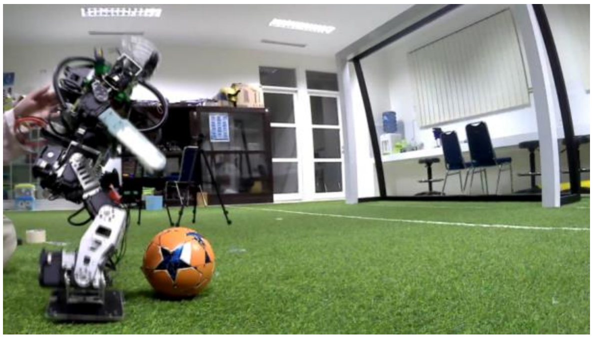


Fig 46. Stand on One Leg

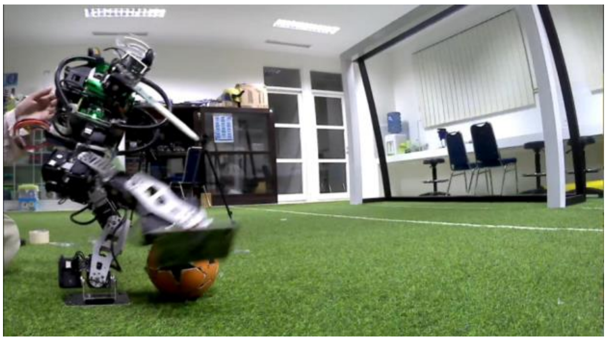


Fig 47. Swinging The Leg

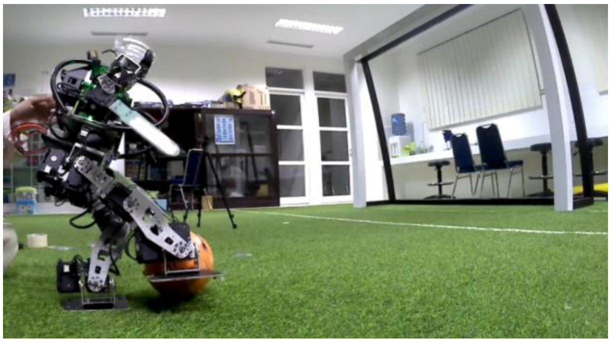


Fig 48. Kick The Ball

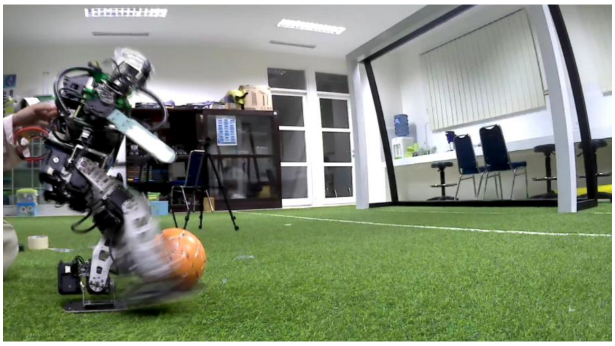


Fig 49. Swinging Back

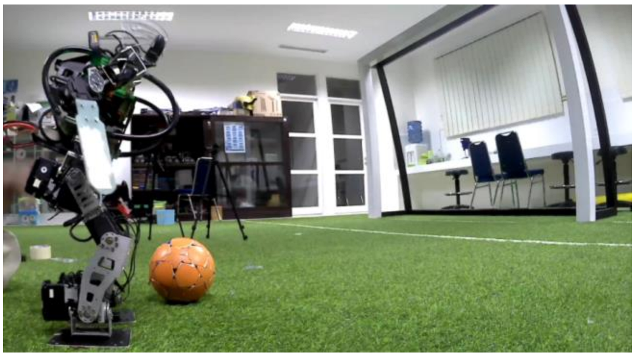


Fig 50. Back to Walk Ready

he velocity of motion is measured based on the angle values generated and the accumulated travel time to complete one full motion for each joint. The travel time is measured using a stopwatch.

Tabel 10. Step Time Motion Polynomial

Data	Time (s)		
	L	R	S
1	4.63	6.86	3.93
2	5.23	5.61	3.58
3	4.98	5.43	4.16
4	6.05	6.02	3.35
5	5.88	5.25	4.04
6	5.44	5.58	3.36
7	5.32	6.17	4.32
8	4.91	5.61	3.50
9	5.35	8.41	3.91
10	5.55	6.05	4.65
total	53.34	60.99	38.8

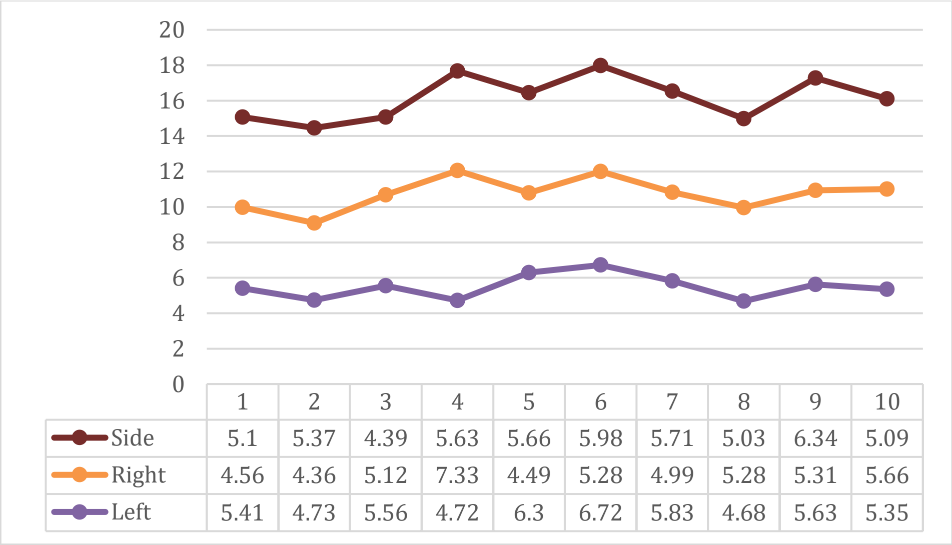
In **Table 10**, the average time to generate a complete motion for each kick (left kick, right kick, and side kick) using the fourth-degree polynomial trajectory-based method can be observed. The table shows the results obtained from conducting 10 experiments and determined using Equation (4).

$$R\ (L) = \frac{53.34}{10} = 5.334\ s$$

$$R\ (R) = \frac{60.99}{10} = 6.099\ s$$

$$R\ (S) = \frac{38.8}{10} = 3.88\ s$$

Based on 10 experimental trials, the average time to generate a complete motion for each kick was as follows: for the left kick, the average time was 5.334 seconds, for the right kick, the average time was 6.099 seconds, and for the side kick, the average time was 3.88 seconds.



	1	2	3	4	5	6	7	8	9	10
● Side	5.1	5.37	4.39	5.63	5.66	5.98	5.71	5.03	6.34	5.09
● Right	4.56	4.36	5.12	7.33	4.49	5.28	4.99	5.28	5.31	5.66
● Left	5.41	4.73	5.56	4.72	6.3	6.72	5.83	4.68	5.63	5.35

Fig 51. Polynomial Step Time Graph

The time to generate motion using the polynomial trajectory method was compared to the previous method, which was record and play. The speed of motion generation using the polynomial trajectory method is shown in **Table 10.** In the record and play method, the time to generate motion is manually inputted during the creation of the motion. The time to generate motion using the record and play method can be observed in **Table 11.**

Table 11. Step Time Motion Record and Play

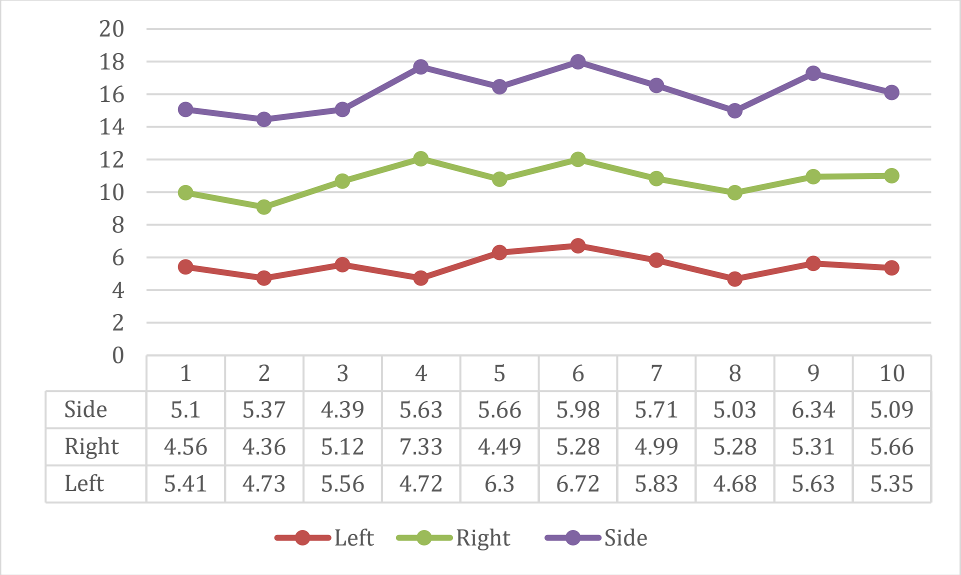
Data	**Time (s)**		
	L	**R**	**S**
1	5.41	4.56	5.10
2	4.73	4.36	5.37
3	5.56	5.12	4.39
4	4.72	7.33	5.63
5	6.30	4.49	5.66
6	6.72	5.28	5.98
7	5.83	4.99	5.71
8	4.68	5.28	5.03
9	5.63	5.31	6.34
10	5.35	5.66	5.09
total	54.93	52.38	54.3

Meanwhile, the average time required is calculated is :

$$R\ (L) = \frac{54.93}{10} = 5.493\ s$$

$$R\ (R) = \frac{5.238}{10} = 5.238\ s$$

$$R\ (S) = \frac{54.3}{10} = 5.430\ s$$



	1	2	3	4	5	6	7	8	9	10
Side	5.1	5.37	4.39	5.63	5.66	5.98	5.71	5.03	6.34	5.09
Right	4.56	4.36	5.12	7.33	4.49	5.28	4.99	5.28	5.31	5.66
Left	5.41	4.73	5.56	4.72	6.3	6.72	5.83	4.68	5.63	5.35

Fig 52. Step Time Comparison Graph

The test results indicate that the record and play method requires 5.493 seconds for a left kick, 5.238 seconds for a right kick, and 5.430 seconds for a side kick. These values represent the average time taken to execute each kick using the record and play method.

4. CONCLUSION

After completing this research, it can be concluded that:

1. Motion trajectory generation based on fourth degree polynomial method is effective in generating motion for R-SCUAD humanoid robot. The test results show that the trajectories generated using this method exhibit smoother and more consistent motion compared to the record and play method.
2. The use of polynomial trajectory-based method provides better control over each joint of the robot. This is evident from the comparison graph that demonstrates significant differences between the motion generated by the trajectory-based method and the record and play method. The polynomial trajectory method is capable of generating trajectories that align with the research objectives.
3. The time required by the polynomial-based trajectory method to generate movements for the left and right kicks, based on the collected data from 10 trials, is significantly lower compared to the time taken by the record and play method. However, for the side kick movement, the polynomial-based trajectory method proves to be effective.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have contributed in this research project. We would like to extend our sincere appreciation to Mr. Nuryono Satya Widodo, S.T., M.Eng. as R-SCUAD supervisor for the invaluable guidance, mentorship, and unwavering support throughout the entire research process. Their expertise and insightful feedback played a crucial role in shaping the direction and outcomes of this study. Furthermore, we would like to express our appreciation to R-SCUAD Robotic Team and Alumni who provided assistance, meaningful discussions, and encouragement during the course of this research. Their valuable insights and input enriched our understanding and improved the quality of our work.

REFERENCES

[1] M. Asada, I. Noda, S. Tawaratsumida, and K. Hosoda, "Robot Humanoid," IEEE Robotics & Automation Magazine, vol. 10, no. 2, pp. 42-50, 2003. [Online]. Available: http://doi.org/10.1109/MRA.2003.1213561

[2] H. Teixeira, T. Silva, M. Abreu, and L. P. Reis, "Humanoid Robot Kick in Motion Ability for Playing Robotic Soccer," in 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 34-39, 2020. http://doi.org/10.1109/ICARSC49921.2020.9096073

[3] Pitowarno, E., & Nasrullah, H. (2019). Buku Panduan KRSBI Humanoid 2019 versi 27.12.18. In ATURAN PERTANDINGAN KONTES ROBOT SEPAK BOLA INDONESIA (KRSBI) Humanoid 2019 (pp. 1-14). Jakarta: Direktorat Kemahasiswaan Direktorat Jenderal Pembelajaran dan Kemahasiswaan Kementerian Riset, Teknologi dan Pendidikan Tinggi.

[4] Irsyad, M. (2020). Walk And Kick Motion Generation For Robot KRSBI Humanoid R-SCUAD (Undergraduate Thesis, Universitas Ahmad Dahlan).

[5] Iskandar, R., Fathur, D., & others. (2020). Penerapan Metode Invers kinematik Pada Kontrol Gerak Robot Lengan Tiga Derajat Bebas. Jurnal Inovasi Fisika Indonesia, 64-71.

[6] S. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in IEEE International Conference on Intelligent Robots and Systems, vol. 1, no. 1, pp. 298-303, 2001, https://doi.org/10.1109/iros.2001.973374

[7] Otani, K., Bouyarmane, K., & Ivaldi, S. (2018). Generating Assistive Humanoid Motions for Co-Manipulation Tasks with a Multi-Robot Quadratic Program Controller. Retrieved from https://hal.archives-ouvertes.fr/hal-01590678v2

[8] V. T. Nguyen and J. H. Kim, "A Review of Robot Soccer Systems and Approaches," International Journal of Advanced Robotic Systems, vol. 16, no. 3, pp. 1-21, 2019.

[9] Hernandez, A., Pardo, D., & Bellas, F. (2018). A Survey on Robot Soccer Simulators: State of the Art and Challenges. Robotics, 7(3), 1-24.

[10] A. Hernandez, D. Pardo, and F. Bellas, "A Survey on Robot Soccer Simulators: State of the Art and Challenges," Robotics, vol. 7, no. 3, pp. 1-24, 2018. [Online]. Available: https://doi.org/10.3390/robotics7030045

[11] Huang, J., Xiao, Y., Zong, Y., & Li, R. (2017). A Review of Robot Soccer Research. International Journal of Advanced Robotic Systems, 14(6), 1-18.

[12] S. Fang et al., "Study on High-Speed and Smooth Transfer of Robot Motion Trajectory Based on Modified S-Shaped Acceleration/Deceleration Algorithm," IEEE Access, vol. 8, pp. 199747-199758, 2020. https://doi.org/10.1109/ACCESS.2020.3039534

[13] Y.-H. Chen, W.-T. Yang, B.-H. Chen, and P.-C. Lin, "Manipulator Trajectory Optimization Using Reinforcement Learning on a Reduced-Order Dynamic Model with Deep Neural Network Compensation," Machines, vol. 11, no. 3, p. 350, 2023, https://doi.org/10.3390/machines11030350

[14] W. Wang, Q. Tao, Y. Cao, X. Wang, and X. Zhang, "Robot Time-Optimal Trajectory Planning Based on Improved Cuckoo Search Algorithm," in IEEE Access, vol. 8, pp. 86923-86933, 2020. doi: https://doi.org/10.1109/ACCESS.2020.2992640

[15] Craig, J. J. (2005). Introduction to Robotics: Mechanics and Control. Addison Wesley. (pp. 15-96).

[16] D. Sidobre and K. Desormeaux, "Smooth Cubic Polynomial Trajectories for Human Robot Interactions," Journal of Intelligent and Robotic Systems, vol. 95, pp. 851-869, 2019, https://doi.org/10.1007/s10846-018-0936-z

[17] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in 2014 IEEE-RAS International Conference on Humanoid Robots, pp. 295-302, 2014, https://doi.org/10.1109/HUMANOIDS.2014.7041375

[18] G. Ficht and S. Behnke, "Online Balanced Motion Generation for Humanoid Robots," 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), Beijing, China, 2018, pp. 1-9, https://doi.org/10.1109/HUMANOIDS.2018.8624945

[19] S. Dikmenli, "FORWARD & INVERSE KINEMATICS SOLUTION OF 6-DOF ROBOTS THOSE HAVE OFFSET & SPHERICAL WRISTS," Eurasian Journal of Science Engineering and Technology, vol. 3, no. 1, pp. 14-28, 2022, https://doi.org/10.55696/ejset.1082648.

[20] Ramuzat, N., & others. (2021). Comparison of Position and Torque Whole Body Control Schemes on the Humanoid Robot TALOS Comparison of Position and Torque Whole-Body Control Schemes on the Humanoid Robot TALOS. Retrieved from https://hal.archives-ouvertes.fr/hal-03145141v2

[21] Denavit, J., & Hartenberg, R. S. (1955). A kinematic notation for lower pair mechanisms based on matrices. Journal of Applied Mechanics, 77, 215-221.

[22] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Souères, "A Reactive Walking Pattern Generator Based on Nonlinear Model Predictive Control," IEEE Robotics and Automation Letters, vol. 2, no. 1, pp. 10-17, 2017. doi: https://doi.org/10.1109/LRA.2016.2518739.

[23] Gulo, S. I. C., & Tamba, T. A. (2021). 10 Jurnal Nasional Teknik Elektro dan Teknologi Informasi | (Design of Trajectory Tracking Controller for Differential-Drive Autonomous Wheeled Mobile Robots).

[24] Maroger, N. Ramuzat, O. Stasse, and B. Watier, "Human Trajectory Prediction Model and its Coupling with a Walking Pattern Generator of a Humanoid Robot," IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 6361-6369, 2021. [Online]. Available: https://doi.org/10.1109/LRA.2021.3092750 (Accessed: August 16, 2022).

[25] E. Cuevas, D. Zaldivar, M. Cisneros, and M. Ramirez-Ortegon, "Polynomial Trajectory Algorithm for a Biped Robot," International Journal of Robotics and Automation, vol. 25, pp. 206-3240, 2010. [Online]. Available: https://doi.org/10.2316/Journal.206.2010.4.206-3240