https://doi.org/10.31436/iiumej.v24i2.2296

HIGH ACCURACY HUMAN MOTION TRAJECTORY GENERATION FOR EXOSKELETON ROBOT USING CURVE FITTING TECHNIQUE

MUHAMMAD ABDUL JALIL[1*], MUHAMMAD FAHMI MISKON[2], BAZLI BIN BAHAR[2]

[1]Mechanical Engineering Department, Polytechnic Port Dickson, Port Dickson, Malaysia
[2]Center of Excellence in Robotic and Industrial Automation (CeRIA), Faculty of Electrical Engineering, Universiti Teknikal Malaysia, Melaka, Malaysia

**Corresponding author: muhammadaj@polipd.edu.my*

(Received: 24 January 2022; Accepted: 22 May 2023; Published on-line: 4 July 2023)

ABSTRACT: Robotic systems often require trajectory planning algorithms that can generate natural human-like movements for tasks such as grasping and manipulation. However, conventional trajectory planning methods may not accurately capture the complex movement patterns observed in humans. In this paper, we present a trajectory planning algorithm based on polynomial curve fitting that aims to address this issue. The algorithm determines the polynomial coefficient values that accurately match the natural human trajectory profile and is evaluated using MATLAB simulations. We compare the proposed algorithm to the conventional quintic polynomial trajectory method, analysing the accuracy, precision, and via-point continuity. The result shows that the algorithm has the ability to generate a trajectory profile with accuracy of 99.8% and a precision of 0.002°. However, the result for via-point continuity shows an error on every sub-phase transition, with the lowest error of 0.0031 between the transition of sub-phases 1 and 2. The result also shows that the lowest fitting error recorded is 0.00014°. The results demonstrate that our algorithm can generate trajectory profiles with higher accuracy and naturalness, potentially improving the performance and usability of robotic systems.

ABSTRAK: Sistem robotik sering memerlukan algoritma perancangan trajektori yang dapat menghasilkan gerakan semulajadi seperti manusia bagi tugas seperti memegang dan memanipulasi objek. Walau bagaimanapun, kaedah perancangan trajektori konvensional mungkin tidak dapat merekodkan pola gerakan kompleks seperti yang dihasilkan manusia secara tepat. Kajian ini adalah berkenaan algoritma perancangan lintasan berdasarkan penyepaduan lengkung polinomial bagi menyelesaikan masalah ini. Algoritma ini menentukan nilai pekali polinomial yang sepadan dengan profil gerakan semulajadi manusia dan dinilai menggunakan simulasi MATLAB. Algoritma yang dicadangkan ini telah dibandingkan dengan kaedah perancangan lintasan polinomial kuintik konvensional, dianalisis kejituan, ketepatan, dan keberterusan titik lalu. Keputusan menunjukkan bahawa algoritma tersebut mampu menghasilkan profil lintasan dengan kejituan sebanyak 99.8% dan ketepatan sebanyak 0.002°. Walau bagaimanapun, dapatan kajian mengenai keberterusan titik lalu menunjukkan ralat pada setiap peralihan fasa-sub dengan ralat terendah sebanyak 0.0031 pada peralihan antara fasa-sub 1 dan fasa-sub 2. Dapatan kajian juga menunjukkan bahawa ralat penyepaduan terendah yang direkodkan adalah sebanyak 0.00014°. Keputusan ini menunjukkan bahawa algoritma ini mampu menghasilkan profil lintasan dengan ketepatan dan sifat semula jadi yang lebih tinggi, berpotensi meningkatkan prestasi dan kegunaan sistem robotik.

KEYWORDS: *trajectory generation; polynomial; curve fitting; via-point; exoskeleton*

1. INTRODUCTION

Trajectory refers to a time history of the position, velocity, and acceleration for each degree of freedom (DOF) [1]. In trajectory generation, the desired trajectory for motions is generated based on the prediction of how the system (robot) responds to the input trajectory. Generally, trajectory generation deals with the problem of 1) how to specify a trajectory with a simple description, 2) how the trajectory is represented, 3) how to generate the trajectory in real-time [1], and 4) how to find a relationship between two domains: time and space [2].

According to Miskon et al. [3], there are three strategies for generating a trajectory for robot applications: off-line, on-line, and combined or hybrid. The off-line strategy is a strategy that uses either a mathematical model such as a polynomial equation [4], Fourier Transform [5], Central Pattern Generation (CPG) [6], Neural Oscillator [7], or uses recorded or normalized human motion data [8]. The advantage of this strategy is that it does not require any dynamic relationship between the robot and the environment. However, the main drawback of this strategy is adaptation due to environmental uncertainties. Also, some of the off-line methods required very accurate robot modelling before the trajectory was planned.

On the other hand, the on-line strategy does not need a predefined trajectory to generate the motion. It has the capability to produce the trajectory according to the working space environment. The implementation of Neural Network [9] and Fuzzy Logic [10] attached with an additional sensor [11,12] is used to improve the accuracy of the generating trajectory. However, this strategy must consider issues such as the fastest time adapting for real-time application and accuracy. Meanwhile, the hybrid strategy used both advantages of off-line and on-line strategy to generate a trajectory for the robot.

The normative human trajectory has been studied to ensure that the robot (in this case, bipedal or exoskeleton) can follow the nature of human motion. Two types of trajectory approaches can generate trajectory profiles using the cartesian and joint space approaches. The cartesian space trajectory generation approach will involve inverse kinematics. There will be multiple solution problems in which matching with human motion will be an issue. Craig [1] stated that there are three general problems in the cartesian space trajectory scheme: first, unreachable intermediate points; second, high joint rates near a singularity; and lastly, start and goal being reachable in different solutions. Meanwhile, in the joint space trajectory approach, the robot's motion design is to be made using joint space values such as joint space position, velocity, etc. The general problem of this approach is the lack of visualization of joint motion and position of the end-effector during the time.

Various methods of implementation have already been established. Within that, the accuracy of the trajectory is an important parameter when designing the trajectory of the robot that operates alongside humans, such as an exoskeleton. The trajectory's accuracy can be viewed as an error between generated (target) trajectory and the reference robot trajectory. The less error shows a higher accuracy of the generating trajectory. The more accurate trajectory generated, the more naturally it follows human motion.

Many methods have been discussed to improve the accuracy of trajectory in exoskeleton robots. In Gomes et al. [13], Gait Pattern Adaptation (GPA) was designed to generate natural human motion for the LOKOMAT rehabilitation robot. The GPA adjusted the generating trajectory to suit the desired trajectory based on torque interaction between humans and robots. This method was also studied in different approaches [14-16]. The NaTUre-gaits is also a technique that is used to generate the trajectory that follows natural human motion for rehabilitation studies [17]. These two methods used predefined trajectories generated from mathematical (cubic) or recorded motion data. Other methods like Complementary Limb Motion Estimation (CLME) [18], Neural Oscillator [7], Gait Phase Switching Algorithm (GPSA) [6], Radial Basis Function (RBF) [19], Neural Network [9], polynomial [20,21], Probabilistic Foam Method (PFM) [22] have also been used to improve accuracy of the generated trajectory profiles to the wearer.

However, all the methods discussed so far require redefining constraint parameters (i.e., start time, stop time, start velocity, stop velocity, etc.) before implementing the method in the robot can be done. There are advantages of having many constraints that need to be considered, such as smooth trajectory, accuracy, etc. However, these constraints can increase the trajectory generator's computation cost and computation error [2].

This paper presents the trajectory generation algorithm that generates an accurate human-like trajectory profile to overcome the limitations stated. Unlike other methods, this method does not require additional parameters such as velocity or acceleration at the trajectory via-point to design trajectory profile. The only parameter required in this design is the joint displacement of the actuator. The complete cycle of time series data consisting of natural human motion data is mapped using the curve fitting approach; then, the quintic polynomial coefficient is produced. The polynomial trajectory is used because of its high precision and ability to calculate the kinematics, dynamics, and control parameters [23].

This paper's main contribution lies in proposing a trajectory generation algorithm that generates highly accurate and natural human-like trajectory profiles. This algorithm can potentially improve the performance and usability of robotic systems in a range of applications, such as grasping and manipulation tasks.

2. POLYNOMIAL CURVE FITTING TRAJECTORY PLANNING (PCFTP)

In robotics, trajectory planning is designed by assigning the initial (t_0) and final time (t_f) and other constraints on position, velocity, acceleration and so on at t_0 and t_f [2]. For the robot application that works alongside humans, such as an exoskeleton, the performance of the trajectory generator is how close the generated trajectory is to the human reference trajectory. Figure 1 shows the application of an exoskeleton robot in the rehabilitation process taken from [24]. In this application, the robot must generate a similar joint trajectory profile to the normal human walking joint to ensure the patient recovers quickly and avoids discomfort.

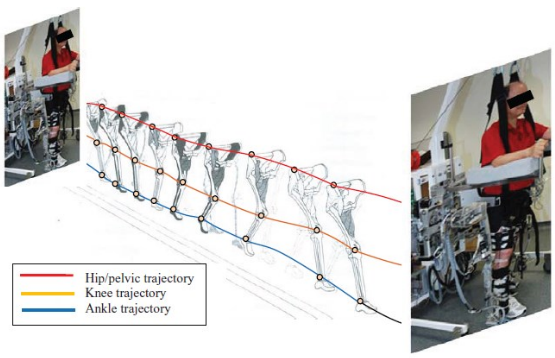

Fig. 1: Illustration of hip, knee and ankle joint trajectory in exoskeleton robot application taken from [24].

https://doi.org/10.31436/iiumej.v24i2.2296

The trajectory generation technique requires some parameters to be set up before generation of the required trajectory. Table 1 shows the initial trajectory parameters. The trajectory generation technique requires the generation of a reference trajectory. It is shown that the current trajectory technique requires multiple initial setup trajectory parameters. Having multiple parameters in the initial setup will improve the accuracy performance of the trajectory generator. However, this will burden the system for processing and require exact mathematical modeling of the robot.

Table 1: Initial parameter setup for trajectory generation technique

Trajectory Generation Technique	Initial Trajectory Parameters
Quintic Polynomial	Start Boundary Parameter: i.e., Starting angular position, velocity, acceleration, jerk, t_{Start} Stop Boundary Parameter: i.e., Stoping angular position, velocity, acceleration, jerk, t_{Stop}
Neural Oscillator	Neuron Parameters: Frequency, coupling coefficients, different phase matrix CoM Trajectory Parameters:
Gait Phase Switching Algorithm (GPSA)	Neuron Parameters: Frequency, coupling coefficients, different phase matrix Gait Parameters: i.e., amplitude, regulating signal

Figure 2 shows the algorithm based on a curve fitting approach to identify the polynomial coefficients. This algorithm can be applied to all strategies discussed in [3] since it requires complete data to represent the motion. These data can be off-line, obtained from biomechanical studies, or come from an on-line time-series sequence received from the actuator encoder. However, in this paper, our algorithm will implement and evaluate only off-line situations. This paper used the human hip motion profile data from a biomechanical study [24] as a reference trajectory.

Algorithm 1: PCFTP

```
Determine Data size of motion profile, mData
Rearrange motion data followed by phases boundary parameter i = phases number
Determine the size of phases data mS_i
while (i < i_max)
    Select polynomial power (n)
    while (∑mS_i < mData)
        Calculate the polynomial coefficient for selected n, mData, and mS_i
    end
end
```

Fig. 2: PCFTP Algorithm.

Step 1: Data Acquisition

The dataset used in this paper was based on a biomechanics study conducted by [25], which employed 40 healthy subjects. The dataset consisted of two different groups, adult and young, with different walking speeds recorded and labeled as normal walking (N), very slow walking (XS), slow walking (S), medium walking (m), fast walking (L), walking on toe (T), walking on heel (H), stair ascending (U) and stair descending (D). In this paper, we used an adult hip normal walking profile. This paper does not cover how joint trajectory is generated from 40 healthy subjects and the criteria for selecting their subject. Other datasets can also be used and are not rigid from the stated researcher.

Step 2: Divided Motion Data Based on Biomechanical Gait Cycle

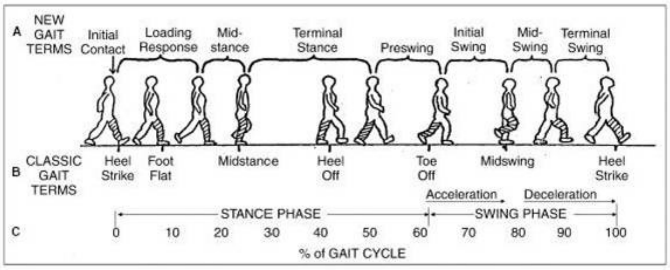

Fig. 3: Phases of human walking gait.

In human biomechanical studies, gait is defined as any method of locomotion characterized by periods of loading and unloading of the limbs [26]. This definition is not restricted to walking; it includes running, hopping, skipping, swimming, and cycling. Since walking is the most frequently used gait in the activities of daily living (ADLs), most of the research definition of gait refers to walking.

The gait cycle in walking is defined as a time interval or sequence of motion occurring from heel strike to heel strike [27]. Fig. 3 shows the complete phases of the human walking gait, with seven phases in the human walking motion [28]. In general, human walking data shows that there is smooth and continuous motion between phase transitions (via-point section). This became the most challenging parameter for both methods to design trajectory profiles while accurately ensuring motion continuity and smoothness.

Table 2 shows the complete walking cycle's human hip motion parameter phase. Each phase has its own parameter starting and ending. Based on [1], these starting and ending parameters are critical when designing the via-point.

Table 2: Human hip phases joint angle parameters based on [28] and [25]

Gait Phases	*Gait Cycle (%)*	*Starting Angle (Deg)*	*Ending Angle (Deg)*	*Starting time (s)*	*Ending time (s)*
Initial Contact	0 – 10	25.20	21.00	0.00	0.10
Mid Stance	10 – 30	21.00	-1.30	0.10	0.30
Terminal Stance	30 – 50	-1.30	-16.10	0.30	0.50
Pre Swing	50 – 60	-16.10	-10.40	0.50	0.60
Initial Swing	60 – 73	-10.40	15.10	0.60	0.73
Mid Swing	73 – 87	15.10	25.10	0.73	0.87
Terminal Swing	87 - 100	25.10	24.10	0.87	1.00

Step 4: Polynomial Curve Fitting Trajectory Planning

The curve fitting technique is the main core of our proposed algorithm. It determines the polynomial coefficient from the normative hip walking profile. The curve fitting method is chosen because of its ability to deal with a series of data. There are two approaches in the curve fitting technique that have the ability to generate a polynomial coefficient. The first is Least-squares regression (LSR), or regression in short, and the second is an interpolation. This paper used the least-square regression approach to generate a polynomial coefficient.

The polynomial regression curve fitting technique is formulated in Eq. (1). Where a_i is a polynomial coefficient. The core idea of this technique is the regression that is used to minimize the sum of the squares of residuals, S_r between the desired y_i and the forecast y' as shown in Eq. (2), where all summation from i =1 through m (number of data) and n (polynomial order).

$$\theta(t)=\sum_{i=0}^{n} a_i t^i \tag{1}$$

$$S_r=\sum_{i=1}^{m}\sum_{j=0}^{n}(y_i-a_j x_i^j)^2 \tag{2}$$

Equation (3) shows the partial derivative of the sum of the residuals over the coefficient $(\frac{\partial S_r}{\partial a_j})$ in Eq. (2). and also in a matrix form of Eq. (3) as shown in Eq. (4).

$$\frac{\partial S_r}{\partial a_j}=-2\sum_{i=1}^{m}\sum_{j=0}^{n} x_i^j(y_i-a_j x_i^j) \quad \begin{cases} i=1,2,\ldots..m \\ j=1,2,\ \ldots..n \end{cases} \tag{3}$$

$$\begin{bmatrix} m & \sum x_i^j & \ldots & \sum x_m^n \\ \sum x_i^j & \ldots & \ldots & \sum x_m^{n+1} \\ \vdots & \ldots & \ldots & \vdots \\ \sum x_i^n & \ldots & \ldots & \sum x_m^{2n} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \vdots \\ \sum x_i^n y_i \end{bmatrix} \tag{4}$$

The standard error $S_{y/x}$, Eq. (5) and the coefficient correlation, Eq. (6), is used as a performance evaluation for the PCFTP method.

$$S_{y/x}=\sqrt{\frac{S_r}{m-(n+1)}} \tag{5}$$

$$r^2=\frac{\Sigma(y_1-\bar{y})-\Sigma S_r}{\Sigma(y_1-\bar{y})} \tag{6}$$

3. SIMULATION SETUP

We used MATLAB to demonstrate the accuracy of the trajectory generated from the PCFTP algorithm and the effectiveness of generation using the trajectory algorithm compared to the quintic polynomial method.

3.1 Comparison of Quintic Trajectory Generation for Generating Hip Joint Trajectory Profile

The quintic polynomial is a fifth-degree polynomial formulated in Eq. (7). Besides position, other constraints, such as velocity and acceleration, must be considered when designing the trajectory. A higher-order polynomial gives more constraints (time derivatives) that can be used to adjust the trajectory to suit the application requirement. However, the more constraints are considered, the more complex it is to determine the unknown coefficients. This constraint provided high computation cost and produced a numerical error for a higher value of polynomial degree [2].

The quintic polynomial can provide five constraints instead of three, as mentioned in Eqs. (7) to (9). These three constraints are enough to obtain a smooth trajectory profile. The suitable initial and final constraints for the position Eq. (7), velocity Eq. (8), and acceleration

Eq. (9) [2] are determined first. Besides, the number of boundary constraints is usually even, and the degree of the polynomial function is odd.

$$\theta(t) = \sum_{i=0}^{n} a_i t^i \tag{7}$$

$$\dot{\theta}(t) = \sum_{i=1}^{n} i a_i t^{i-1} \tag{8}$$

$$\ddot{\theta}(t) = \sum_{i=2}^{n} i(i-1) a_i t^{i-2} \tag{9}$$

Therefore, from Eq. (7) to (9), there are six boundary constraints (starting position, θ_0, ending position, θ_f, starting velocity, v_0, ending velocity, v_0, starting acceleration, s_0 and ending acceleration, s_f) of a quintic polynomial as shown in Eq. (10), that needs to be determined.

$$\begin{aligned}
\theta(t_0) &= \theta_0 \\
\theta(t_f) &= \theta_f \\
\dot{\theta}(t_0) &= v_0 \\
\dot{\theta}(t_f) &= c \\
\ddot{\theta}(t_0) &= s_0 \\
\ddot{\theta}(t_f) &= s_f
\end{aligned} \tag{10}$$

This constraint is then used to determine the quintic coefficient, as shown in Eq. (11). The first three coefficients, a_0, a_1, and a_2, are the initial values of the generated trajectory's position, velocity, and acceleration.

Table 3 shows the polynomial coefficients generated using quintic polynomial trajectory planning for each sub-phase. Table 4 shows the polynomial coefficients using curve fitting. All these coefficients are then used again to generate the trajectory. This trajectory is then compared to the recorded trajectory profile to validate the accuracy and effectiveness of the proposed method.

$$\begin{aligned}
a_0 &= \theta_0 \\
a_1 &= \dot{\theta}_0 \\
a_2 &= \frac{\ddot{\theta}_0}{2} \\
a_3 &= \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_f - \ddot{\theta}_0){t_f}^2}{2{t_f}^3} \\
a_4 &= \frac{30\theta_f - 30\theta_0 - (14\dot{\theta}_f + 16\dot{\theta}_0)t_f - (3\ddot{\theta}_f - 2\ddot{\theta}_0){t_f}^2}{2{t_f}^4} \\
a_5 &= \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f - (\ddot{\theta}_f - \ddot{\theta}_0){t_f}^2}{2{t_f}^5}
\end{aligned} \tag{11}$$

4. RESULTS

This section comprehensively analyses a significant problem known as the via-point disjointed problem. Within this context, we explore trajectory accuracy, employing two distinct methods: the quintic polynomial method and the PCFTP method. By thoroughly examining and comparing the outcomes generated by these methods, we aim to understand better their respective strengths, limitations, and overall effectiveness in achieving accurate trajectories. **Error! Reference source not found.** shows the hip joint motion simulation r esult based on the quintic polynomial trajectory method, and **Error! Reference source not found.** shows the hip joint motion simulation result based on the polynomial curve fitting method.

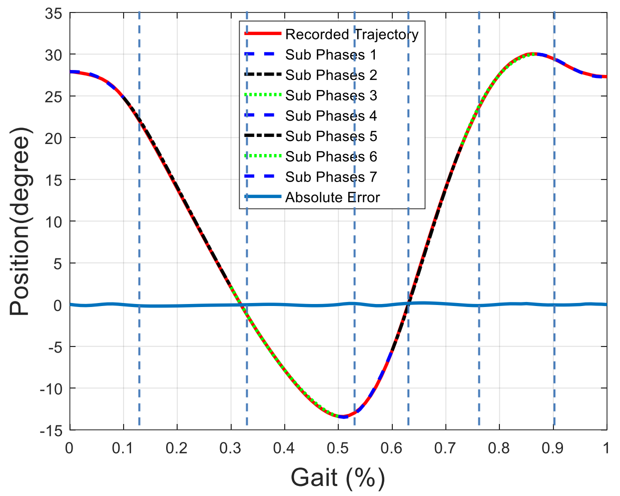

Fig. 4: Hip joint trajectory profile generated using quintic polynomial.

4.1 Via-point Disjointed Analysis

Via-point is an intermediate position that the system or object needs to pass through during its movement from the initial position to the final destination. The via-point is located at every phase's transition. Based on Winter [28], the seven phases of human walking require six via points for the complete cycle. As mentioned in the previous section, the ending position for Phase 1 is a starting position for Phase 2.

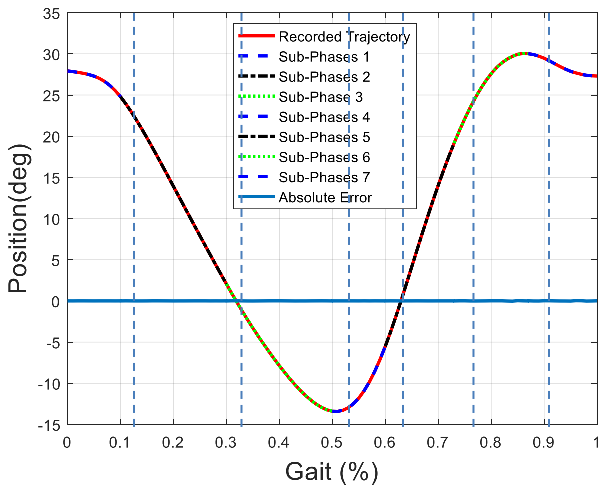

Fig. 5: Hip joint trajectory profile generated using PCFTP.

Table 3: Coefficient of a polynomial using quintic polynomial trajectory method

Phases	***Coefficient***					
	a_5	a_4	a_3	a_2	a_1	a_0
1	-8.102 x10^5	2.108 x10^5	17178	0	0	25.2
2	32128	-35861	15985	-3572.4	285.07	15.497
3	-2268.6	4303.8	-2828.9	958.46	-298.37^4	48.981
4	-8.039 x10^6	-2.216 x10^6	2.439 x10^6	1.339 x10^6	3.666 x10^5	-40040
5	1.932 x10^5	-6.268 x10^5	8.072 x10^5	-5.152 x10^5	1.630 x10^5	-20480
6	-99297	4.0306 x10^5	-6.5399 x10^5	5.2952 x10^5	-2.136 x10^5	34301
7	64871	--3.364 x10^5	6.915 x10^5	-7.052 x10^5	3.571 x10^5	-71816

Table 4: Coefficient of a polynomial using the PCFTP method

Phases	***Coefficient***					
	a_5	a_4	a_3	a_2	a_1	a_0
1	75178	-82804	14364	-1158.9	6.0584	25.193
2	91252	-88042	32870	-5978.7	424.14	13.481
3	-88293	1.72 x10^5	-1.32 x10^5	50020	-9507.7	732.37
4	2.140 x10^5	-5.62 x10^5	5.90 x10^5	-3.09 x10^5	80235	-8293.4
5	1.61 x10^5	-5.34 x10^5	7.02 x10^5	-4.58 x10^5	1.48 x10^5	-19082
6	84648	-3.37 x10^5	5.33 x10^5	-4.22 x10^5	1.67 x10^5	-26431
7	2.17 x10^5	-1.06 x10^6	2.08 x10^6	-2.02 x10^6	9.84 x10^5	-1.91 x10^5

Table 5 compares generated via-point for each sub-phase between the quintic polynomial and PCFTP. It shows that the quintic polynomial method handles via-points without encountering problems during each phase. However, the PCFTP method demonstrates a notable concern with disjointed via-points in the phase transitions. This disparity highlights the quintic method's ability to seamlessly incorporate via-points across all phases, while the PCFTP method experiences challenges in maintaining continuity and cohesiveness in the trajectory when navigating through different phases.

Table 5: Start and Stop Position at via-point

Phases	**Quintic Polynomial (deg)**		**PCFTP (deg)**	
	Start	***Stop***	***Start***	***Stop***
1	25.200	21.000	25.193	21.046
2	21.000	-1.300	21.087	-1.265
3	-1.300	-16.100	-1.303	-16.167
4	-16.100	-10.400	-16.143	-10.443
5	-10.400	15.100	-10.453	13.473
6	15.100	25.100	13.480	25.316
7	25.100	24.100	25.310	24.086

The issue with our algorithm is rooted in the fundamental characteristics of the curve fitting technique. This method utilizes a best-fitting strategy to effectively incorporate and precisely depict the data within a designated interval, as outlined in Eq. (5) and Eq. (6). On the other hand, the quintic method integrates the initial and final values as constraints in Eq. (11). The utilization of constraints guarantees that the trajectory's initiation and termination occur at predetermined positions, thereby enabling accurate management of the start and end points.

In Table 6, the absolute errors associated with each via-point are presented, reflecting the performance of the PCFTP method. Notably, the most significant error is observed during the transition from phase 4 to phase 5, specifically for the stop position, with an error magnitude of approximately 1.627°. Additionally, the transition from phase 5 to phase 6 exhibits an error of approximately 1.620°. It is important to note that these errors directly impact the overall smoothness and continuity of the trajectory, underscoring the need for further analysis and potential improvements in the PCFTP method to mitigate such errors and enhance trajectory quality.

Table 6: Absolute Error of a Via-point in Each Phases Transition for Curve Fitting

Phases Transition	**Absolute Error Start Position (deg)**	**Absolute Error Stop Position (deg)**
1 – 2	0.007	0.046
2 – 3	0.087	0.035
3 – 4	0.003	0.067
4 – 5	0.043	1.627
5 – 6	1.620	0.216
6 – 7	0.210	0.014

4.2 Overall Fitting Performance Analysis

An extensive evaluation was performed to compare the fitting errors of the quintic polynomial method and PCFTP methods' fitting errors to obtain significant insights into the precision and efficiency of trajectory generation techniques. Fig. 6 compares fitting errors between the quintic method and the curve fitting method. Based on Fig. 6, the PCFTP approach has a lower fitting error than the quintic trajectory generation.

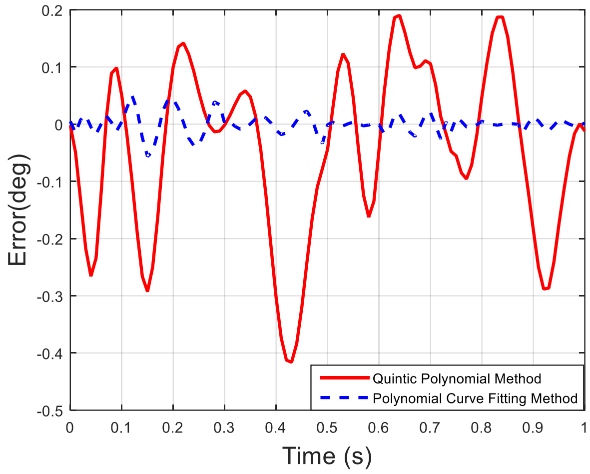

Fig. 6: Fitting error comparison between quintic polynomial and PCFTP methods.

Table 7 shows the comparison of MAE for each phase for both methods. MAE is the mean of fitting error for each phase. Table 6 shows that the polynomial curve fitting method gives a high efficiency with the lowest generation error 0.002° phase. However, the lowest generation error for the quintic polynomial approach is 0.086° phase. The overall MAE for the quintic polynomial approach is 0.112° phase and 0.011° phase for the polynomial curve fitting approach. The MAE represents the difference between the desired trajectory and generated trajectory.

Table 8 shows the Root Mean Square Error (RMSE) comparison between the quintic polynomial and PCFTP methods. These values indicate the accuracy of the trajectory profile. It shows that the PCFTP method exhibits superior precision in trajectory profile compared to the quintic polynomial method.

Table 7: Mean absolute error (MAE) comparison between quintic and PCFTP methods

Sub-Phase	**Method**	
	Quintic (deg)	***Curve Fitting (deg)***
1	0.115	0.009
2	0.101	0.032
3	0.164	0.012
4	0.086	0.002
5	0.101	0.013
6	0.089	0.004
7	0.129	0.006

Table 8: RMSE comparison between quintic polynomial and PCFTP methods

Sub-Phase	**Method**	
	Quintic (deg)	***Curve Fitting (Deg)***
1	7.030×10^{-2}	6.783×10^{-15}
2	2.900×10^{-2}	4.289×10^{-14}
3	1.431×10^{-1}	4.602×10^{-13}
4	1.440×10^{-2}	4.151×10^{-11}
5	9.210×10^{-2}	1.221×10^{-11}
6	3.730×10^{-2}	6.500×10^{-11}
7	1.267×10^{-1}	3.035×10^{-10}

Considering the RMSE values obtained, we can observe that the PCFTP method consistently outperforms the quintic polynomial method regarding trajectory accuracy. This outcome underscores the efficacy of the PCFTP method in achieving a closer fit to the desired trajectory, resulting in minimized deviations from the intended motion path.

These findings highlight the potential advantages of adopting the PCFTP method over the quintic polynomial method, particularly in applications with high trajectory accuracy, such as robotics, motion planning, and autonomous navigation systems.

5. CONCLUSION

In conclusion, the proposed PCFTP algorithm generates natural human motion profiles more accurately than the quintic polynomial trajectory generation method. The algorithm does not require constraint parameters such as angular position, velocity, and acceleration, which is a significant advantage. However, it requires the complete datasets or trajectory profiles to be modelled, which can sometimes be challenging. The proposed algorithm achieved a trajectory profile accuracy of 99.8% and precision of 0.002°, significantly improving over existing methods. Moreover, the performance validation is based on human biomechanics walking data from [25]. This walking data is not based on a single person but on the normalization of 40 healthy people. Different biomechanics data can be used as well instead of this data. However, the result might vary slightly due to a few factors. First, the curve fitting technique performance is based on data trends in each segment or phase. If the data trend is well segmented, the fitting performance is high. Second, the polynomial degrees are used to model the trajectory data.

However, an open question for a proposed method is how to improve, especially in disjointed via-point problems on every phase transition. This disjointed issue will affect the overall smoothness of the motion profile. The future work for this research is the new extension of this method that determines the numbers of via-points based on best fitting accuracy (Eqs. (5) and (6)) and continuity. Also, implementing the proposed algorithm to real-time data will guide us to a new dimension of using the curve fitting approach in trajectory planning since the curve fitting approach works only if all fitting data are complete.

ACKNOWLEDGMENT

The authors would like to thank the Ministry of Higher Education Malaysia for funding this project under the RAGS research grant scheme RAGS/1/2014/TK03/FKE/B00054. This project was conducted at the Center of Excellence in Robot and Industrial Automation (CERIA), Universiti Teknikal Malaysia Melaka.

REFERENCES

[1] Craig JJ. (2014) Introduction to Robotics: Mechanics and Control, 3rd Edition Pearson.

[2] Biagiotti L, Melchiorri C. (2009) Trajectory Planning for Automatic Machines and Robots. Berlin, Heidelberg, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-540-85629-0.

[3] Miskon MF, Yusof MBAJ. (2014) Review of trajectory generation of exoskeleton robots., in 2014 IEEE International Symposium on Robotics and Manufacturing Automation (ROMA), IEEE, 12-17. doi: 10.1109/ROMA.2014.7295854.

[4] Wang H, Zhao Q, Li H, Zhao R. (2022) Polynomial-based smooth trajectory planning for fruit-picking robot manipulator. Information Processing in Agriculture, 9(1): 112-122. doi: 10.1016/j.inpa.2021.08.001.

[5] Kulak T, Silverio J, Calinon S. (2020) Fourier movement primitives: an approach for learning rhythmic robot skills from demonstrations., in Robotics: Science and Systems XVI, Robotics: Science and Systems Foundation. doi: 10.15607/RSS.2020.XVI.056.
[6] Wei S, Wu H, Liu L, Zhang Y, Chen J, Li Q. (2022) A CPG-based gait planning and motion performance analysis for quadruped robot. Industrial Robot: the international journal of robotics research and application, 49(4): 779-797. doi: 10.1108/IR-08-2021-0181.
[7] Liu C, Yang J, Bu W, Chen Q. (2016) A trajectory generation method for biped walking based on neural oscillators., in 2016 IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC), IEEE, 1-6. doi: 10.1109/ICNSC.2016.7478995.
[8] Wang Q, Qian J, Zhang Y, Shen L, Zhang Z, Feng Z. (2007) Gait trajectory planning and simulation for the powered gait orthosis., in 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, 1693-1697. doi: 10.1109/ROBIO.2007.4522420.
[9] Watanabe T, Johnson EN. (2018) Trajectory Generation using Deep Neural Network. 2018 AIAA Information Systems-AIAA Infotech @ Aerospace, (January): 1-12. doi: 10.2514/6.2018-1893.
[10] Aly AA, Vu MT, El-Sousy FFM, Alotaibi A, Mousa G, Le D, Mobayen S. (2022) Fuzzy-Based Fixed-Time Nonsingular Tracker of Exoskeleton Robots for Disabilities Using Sliding Mode State Observer. Mathematics, 10(17): 3147. doi: 10.3390/math10173147.
[11] Kundu AS, Mazumder O, Chattaraj R, Bhaumik S, Lenka PK. (2014) Trajectory generation for myoelectrically controlled lower limb active knee exoskeleton., in 2014 Seventh International Conference on Contemporary Computing (IC3), IEEE, 230-235. doi: 10.1109/IC3.2014.6897178.
[12] Schuy J, Mielke T, Steinhausen M, Beckerle P, Rinderknecht S. (2015) Design and evaluation of a sensor minimal gait phase and situation detection Algorithm of Human Walking. IEEE-RAS International Conference on Humanoid Robots, 2015-Dec: 20-25. doi: 10.1109/HUMANOIDS.2015.7363517.
[13] Gomes MA, Silveira GLM, Siqueira AAG. (2011) Gait pattern adaptation for an active lower-limb orthosis based on neural networks. Advanced Robotics, 25(15): 1903-1925. doi: 10.1163/016918611X588899.
[14] Siqueira AAG, Jardim B, Vilela PRC, Winter TF. (2008) Analysis of gait-pattern adaptation algorithms applied in an exoskeleton for lower limbs., in 2008 16th Mediterranean Conference on Control and Automation, IEEE, 920–925. doi: 10.1109/MED.2008.4602081.
[15] Gomes MA, Silveira GLM, Siqueira AAG. (2009) Gait-pattern adaptation algorithms based on neural network for lower limbs active orthoses., in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009, IEEE, 4475-4480. doi: 10.1109/IROS.2009.5354232.
[16] Silveira GLM, Gomes MA, Siqueira AAG. (2010) First experimental results on adaptation algorithms for lower limbs active orthoses., in 18th Mediterranean Conference on Control and Automation, MED'10, IEEE, 1316-1321. doi: 10.1109/MED.2010.5547868.
[17] Wang P, Low KH, McGregor AH. (2011) A subject-based motion generation model with adjustable walking pattern for a gait robotic trainer: NaTUre-gaits., in 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 1743-1748. doi: 10.1109/IROS.2011.6048809.
[18] Vallery H, van Asseldonk EHF, Buss M, van der Kooij H. (2009) Reference Trajectory Generation for Rehabilitation Robots: Complementary Limb Motion Estimation. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 17(1): 23-30. doi: 10.1109/TNSRE.2008.2008278.
[19] Daachi ME, Madani T, Daachi B, Djouani K. (2015) A radial basis function neural network adaptive controller to drive a powered lower limb knee joint orthosis. Applied Soft Computing, 34: 324-336. doi: 10.1016/j.asoc.2015.04.034.
[20] Mohammed MQ, Miskon MF, Ali SA. (2017) The effectiveness of kinematic constraints on the accuracy of trajectory profile of human walking using PSPB technique. International Journal of Mechanical and Mechatronics Engineering, 17(6): 86-96

[21] Sapiee MR, Marhaban MH, Ishak AJ, Miskon MF. (2020) Trajectory Generation for Hip Rehabilitation Exoskeleton Using Trajectory Morphing Method. International Journal of Human and Technology Interaction, 4(1): 45-51
[22] Nascimento LBP, Barrios-Aranibar D, Alsina PJ, Santos VG, Fernandes DHS, Pereira DS. (2020) A Smooth and Safe Path Planning for an Active Lower Limb Exoskeleton. Journal of Intelligent and Robotic Systems: Theory and Applications, 99(3-4): 535-553. doi: 10.1007/s10846-019-01134-7.
[23] Sidobre D, Desormeaux K. (2019) Smooth Cubic Polynomial Trajectories for Human-Robot Interactions. Journal of Intelligent & Robotic Systems, 95(3-4): 851-869. doi: 10.1007/s10846-018-0936-z.
[24] Low KH. (2011) Subject-oriented overground walking pattern generation on a rehabilitation robot based on foot and pelvic trajectories. Procedia IUTAM, 2: 109-127. doi: 10.1016/j.piutam.2011.04.012.
[25] Bovi G, Rabuffetti M, Mazzoleni P, Ferrarin M. (2011) A multiple-task gait analysis approach: Kinematic, kinetic and EMG reference data for healthy young and adult subjects. Gait & Posture, 33(1): 6-13. doi: 10.1016/j.gaitpost.2010.08.009.
[26] Kirtley C. (2006) Introduction., in Clinical Gait Analysis, Elsevier, 5-14. doi: 10.1016/B978-0-443-10009-3.50004-7.
[27] Malanga G., Delisa J. (1998) Clinical Observation. In Gait analysis in the science of rehabilitation. Edited by Delisa J. Baltimore, Diane Publishing; pp 1-10.
[28] Winter DA. (2009) Biomechanics and Motor Control of Human Movement, 4th ed.: New Jersey: John Wiley & Sons, Inc. doi: 10.1002/9780470549148.