Design and Fabrication of Robotic arm for Multiple Degrees of Freedom Control

Xinwen Yao

Abstract

This paper details the design, construction, and evaluation of a highly flexible, high-precision robotic arm primarily used in prosthetic and rehabilitation engineering applications. The robotic arm was designed using 3D printing and SolidWorks and features a modular design that simplifies the manufacturing process and improves scalability. The arm has three degrees of freedom and is realized by a well-designed circuit program integrating ESP32 and STM32 microcontrollers. The structural integrity and functionality of the arm were verified through finite element analysis and kinematic simulation. The study also presents a novel path-planning algorithm for the robot arm, which is based on linear position and Gaussian velocity models and is effective in generating smooth, continuous, and physically constrained paths. This work provides an open-source solution that offers a cost-effective and robust platform for diverse robotics applications.

Keywords: robotic arm, modular design, ESP32 and STM32 microcontrollers, path planning, multi-degree freedom.

1. Introduction

The robotic arm is robotics technology's most widely used automated mechanical device. It can be seen in industrial manufacturing, medical treatment, entertainment services, military, semiconductor manufacturing, and space exploration (Moran, 2007). Its unique operational flexibility can adapt to many fields that traditional automation equipment cannot (Malvezzi et al., 2019). It can save a lot of repetitive, uncreative labor of workers. It can also reduce labor costs and increase production. This robotic system has multiple joints that allow it to move in a plane or three-dimensional space. It can have flexible and high-precision operation through control methods such as PID (Hussain et al., 2015). Using servo motors or FOCs, the robotic arm can achieve force feedback and precisely control the magnitude and direction of the force on an object (Krausz & Hargrove, 2019). Robotic arms can also replace humans to work in environments unsuitable for humans, such as small and dangerous (Dominijanni et al., 2021).

3D printing is a front-end industry that integrates models that cannot be processed by CNC and pouring by stacking, and its application prospects are very broad. The integrated processing makes it have good performance in structural mechanics. Its structure can be drawn directly in computer-aided drawing (CAD) and sliced for fabrication (Mick et al., 2019). 3D printing can now be used in industrial design, jewelry, architecture, engineering and construction (AEC), geographic information systems, aerospace, civil engineering, biology, etc (Siemasz et al., 2020). It has excellent performance in structure. The current metal 3D printing can make up for the shortcomings of traditional plastic 3D printing in terms of strength (Maity et al., 2019). 3D printing of certain biological materials can even make medical materials such as artificial organs. In the food industry, there is also the field of 3D food printing.

This robotic arm uses modular ideas, minimizes the structure types, and reuses some structures. It uses very few structural parts but still can move in all x-axis, y-axis, and z-axis. There are only three kinds of structural parts in the entire robotic arm, which can reduce the complexity of the assembly line in manufacturing and lower the demand for purchasing. Also, this design idea makes this robotic arm have very strong scalability; it's easy to add more and makes it longer. The program is also relatively easy to write; there is no need to consider too many differences between different structures, and the length of the robotic arm can be easily extended. However, in this structure, the proportion of length increased by each additional structure will be lower and lower, forming a larger marginal effect. Moreover, this structure has higher requirements for the supporting parts, and the strength requirement is greater than that of ordinary commercially available mechanical arms.

This paper presents highly flexible and high-precision robotic arms, which have a high degree of freedom, a simple structure, and an easy manufacturing process. Relying on technical solutions drawn from similar works, we aimed to design a robot using Solidworks modeling and 3D printing technology. Firstly, this work analyzed and summarized the development status of robotic arms and 3D printing technology and determined the research content of this paper based on the current status of research on robotic systems. Secondly, based on the degrees of freedom analysis, this work designed the movement types and mechanisms of robotic arms, and the kinematic theory analysis and kinematic simulation verification are carried out, respectively. Moreover, based on the analysis of process requirements and the bionic principle of the mechanism for scale synthesis, determine the size of each linkage; determine the drive mode of the robot, several transmission options for each joint to compare, determine a reasonable transmission mode, and finally completed the structural design of the robot arm, and in the 3D software to complete the virtual assembly of the welding robot arm. Finally, based on the finite element analysis method, the static analysis of the components of the robot arm in this design, the analysis of the stress and strain of the structural components to verify the rationality of the structural design, combined with the virtual prototype technology, given the speed of the joints, the observation of the joints in the process of movement of the force, to complete the analysis of the dynamics of the robot arm. This work presents an open-source solution that can be used to control the robots and implemented at relatively low costs. Additionally, even though its use cases are not limited to this field, this robotic platform is primarily intended for prosthetics and rehabilitation engineering applications.

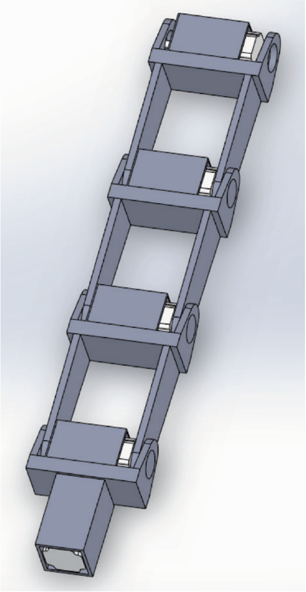


Figure 1. Structure diagram of the completed robotic arm.

2. Method

This project intends to design a wearable forearm robotic arm with three degrees of freedom: the slide's expansion and contraction, the up and down rotation of the wrist part, and the grasping and releasing of the robotic hand part. AutoCAD and SolidWorks carried out the preliminary structural modeling and design, and part of the structure was made of existing materials. Circuit program design was carried out through Arduino.

2.1. Structural Design

The advantage is that the structure is simple and easy to manufacture, and the disadvantage is that the space utilization rate is low. The quick-release and quick-change design is adopted on the work platform, which makes it easier to update the use of the robotic arm, but it also increases the complexity of the program.

2.1.1. Robotic Arm Base

Figure 2 shows a detailed design concept for a robot arm base component that rotates along the z-axis. The rectangular shape of the base design is not only aesthetically pleasing but also practical - it is shaped and sized for easy integration with other devices. This design detail contributes to the base's versatility in various application scenarios.

Four connection holes have been purposely designed where the base connects to the motor. These holes provide a solid fixation for the motor and can also be used as support posts to connect other hardware devices or system components, thus increasing the base's versatility.

Regarding the choice of motor, we used a conventional brushless AC motor. This type of motor is easy to maintain and has a long lifespan and an excellent energy efficiency ratio.

Lastly, we used advanced magnetic field orientation control (FOC) technology regarding the control section. This type of control can regulate the motor's corner and output force more precisely, ensuring the accuracy and reliability of the robot arm operation.

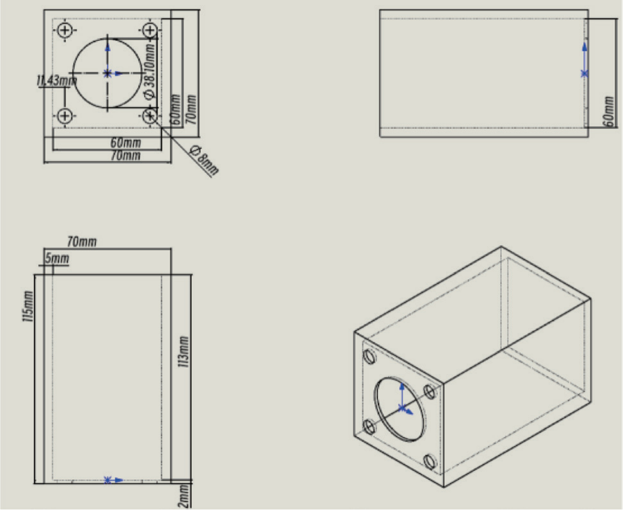


Figure 2. The Base #1 part of the robotic arm rotates along the z-axis.

2.1.2. The Multifunctional Universal Connection Part

This section focuses on the Universal Attachment, a critical part of the robot arm. This multifunctional part can be connected to another Universal Connection Part or a second base assembly, Base #2. Figure 3 illustrates the Universal connection part with a motor inside, rotating along the x-axis. One of the most significant advantages of this design is its contribution to simplifying the manufacturing process. By using this standardized, reusable component, the production and assembly of the robot arm is simpler, reducing the number of parts that need to be manufactured. In addition, this reusability feature adds a strong element of scalability to the system. This modular design approach makes system expansion and upgrades easier and less costly. Additionally, the component is designed to support rotation along the z-axis, adding another layer of flexibility to the motion capabilities of the robot arm.

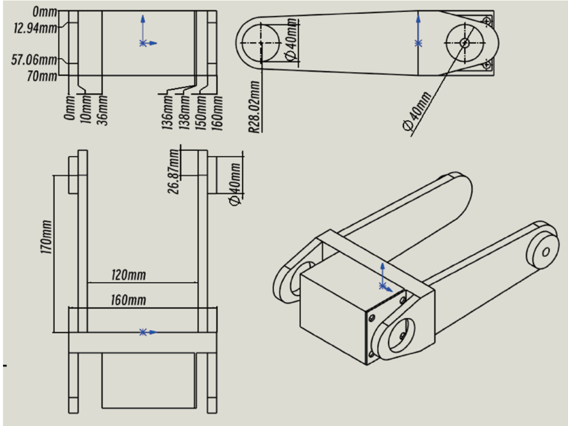


Figure 3. Universal connection part with the motor inside, rotating along the x-axis.

2.1.3. The Second Layer of Stability and Functionality

Figure 4 shows "Base #2" on top of "Base #1". This specially designed base will be attached to "Base #1" and the "Universal Connection Part." When "Base #1" is used in conjunction with "Base #2", these two components together allow the robot arm to rotate along the z-axis. This design not only enhances the overall system's stability but also offers the possibility of subsequent upgrades and modular expansion. This also means the robot arm offers greater flexibility and adaptability to complex tasks or changing environments.

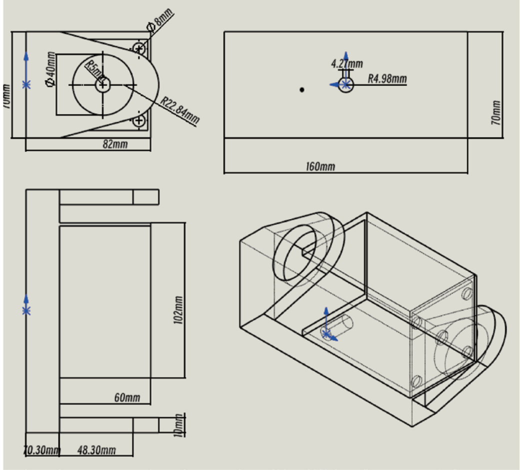
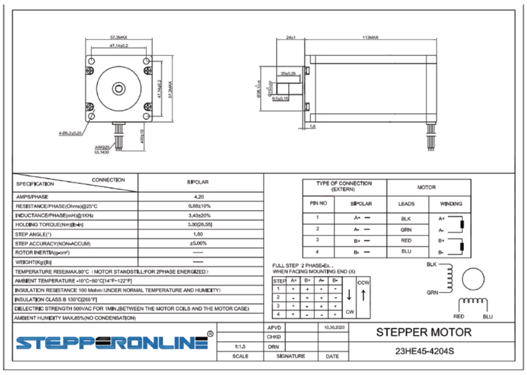


Figure 4 base #2 which is on the top of the base #1

2.2 Circuit Design

2.2.1. Motor and Driver

Figure 5 shows the motor used for this robot arm - a stepper motor model 32HE45-4304S with a step angle of 1.8 degrees and a torque of 3NM. This motor is ideal for applications that require precise control and high torque. Its official driver makes control easier, optimizing overall system reliability and actuation accuracy.



SPECIFICATION / CONNECTION	BIPOLAR
AMPS/PHASE	4.20
RESISTANCE/PHASE(Ohms)@25°C	0.88±10%
INDUCTANCE/PHASE(mH)@1KHz	3.40±20%
HOLDING TORQUE(Nm)[lb-in]	3.00[26.55]
STEP ANGLE(°)	1.80
STEP ACCURACY(NON-ACCUM)	±5.00%
ROTOR INERTIA(g-cm²)	——
WEIGHT(Kg)[lb]	——

TEMPERATURE RISE:MAX.80°C（MOTOR STANDSTILL;FOR 2PHASE ENERGIZED）
AMBIENT TEMPERATURE -10°C~50°C[14°F~122°F]
INSULATION RESISTANCE 100 Mohm（UNDER NORMAL TEMPERATURE AND HUMIDITY）
INSULATION CLASS B 130°C[266°F]
DIELECTRIC STRENGTH 500VAC FOR 1MIN.(BETWEEN THE MOTOR COILS AND THE MOTOR CASE)
AMBIENT HUMIDITY MAX.85%(NO CONDENSATION)

TYPE OF CONNECTION (EXTERN)		MOTOR	
PIN NO	BIPOLAR	LEADS	WINDING
1	A+ —	BLK	A+
2	A- —	GRN	A-
3	B+ —	RED	B+
4	B- —	BLU	B-

FULL STEP 2 PHASE-Ex. ,
WHEN FACING MOUNTING END (X)

STEP	A+	B+	A-	B-		
1	+	+	-	-	↓	CCW
2	-	+	+	-		↑
3	-	-	+	+		
4	+	-	-	+	CW	

	APVD		10.30.2020	STEPPER MOTOR
	CHKD			
1:1.5	DRN			23HE45-4204S
SCALE	SIGNATURE		DATE	

Figure 5. Motor 32HE45-4304S 1.8-degree 3NM

Figure 6, on the other hand, reveals the stepper motor driver's internal working mechanism, which converts digital control signals into analog signals and isolates the digital and analog signals. The controller sends three signals to the driver: PUL (pulse), DIR (direction), and ENA (enable), which is the enable pin used to activate the driver to control the motor, and DIR, which determines the direction of rotation of the motor, providing a high degree of flexibility to the system.

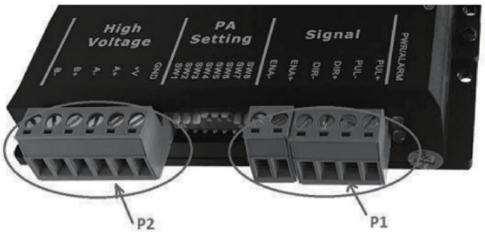


Figure 6. stepper motor driver digital to analog

2.2.2. Electronic Control Module

For the control aspects of the robotic arm, I conducted a comprehensive comparison of multiple controllers to determine the most suitable option. These controllers include but are not limited to STM32, ATmega, RP2040, and ESP32. Table 1 shows an in-depth look at the specifications and features of these controllers. This detailed comparison aids in making an informed decision for selecting the most appropriate controller based on various factors such as processing power, memory, frequency, GPIO count, and additional features like Wi-Fi and Bluetooth capabilities.

All the microcontrollers under consideration offer cost-effectiveness to varying degrees. After an in-depth analysis, it became evident that while the ATMEGA series boasts low price points, their performance metrics are inadequate for the application. Specifically, they lack the computational power needed for motor control and real-time coordinate system conversions, which involve extensive matrix calculations.

Therefore, my focus shifted to the more robust options among the ESP32, STM32, and RP2040 controllers. Although the RP2040 presents an attractive cost-to-performance ratio, its lack of built-in storage disqualified it from my application. Upon further research, I came across the cl100 from Chiploop Tech, which intriguingly incorporates both ESP32 and STM32 chips.

The ESP32 excels in computational capabilities, making it ideal for complex matrix calculations. On the other hand, STM32's rich set of GPIOs makes it well-suited for more general controls and tasks. Moreover, ESP32 has added advantages, such as 2.4 GHz Wi-Fi and Bluetooth Low Energy (BLE) 5 capabilities, offering the potential for internet connectivity and remote control of the robotic arm.

My project uses a stepper motor without an encoder, so advanced control schemes like Field Oriented Control (FOC) and PID control aren't applicable. The driver operates in open-loop mode, making these control methodologies unfeasible for this specific application.

Table 1. Comprehensive Comparison of Microcontrollers for Robotic Arm Applications

Type	ESP32-S3-WROOM-1U-N16R8	STM32F446RET6	RP2040 (Raspberry Pi Pico)	ATMEGA2560(Arduino Mega)	ATMEGA328P-AU (Arduino uno)
Processor	dual-core 32-bit LX7 microprocessor	Arm Cortex-M4 32-bit MCU+FPU	Dual ARM Cortex-M0+	Low Power AVR® 8-Bit Microcontroller	Low Power AVR® 8-Bit Microcontroller
Frequency	Up to 240Mhz	Up to 180 MHz	Up to 133MHz	Up to 16MHz	Up to 20Mhz
Performance	N/A (600MIPS for ESP32)	225 DMIPS	N/A	16 MIPS	20 MIPS
Internal ROM	384 KB	512 KB	N/A	256KB Flash+4KB EEPROM	32KB Flash+1KB EEPROM
Internal RAM	512 KB SRAM+16 KB SRAM in RTC	128 KB SRAM	264KB SRAM	8KB	2KB

Type	ESP32-S3-WROOM-1U-N16R8	STM32F446RET6	RP2040 (Raspberry Pi Pico)	ATMEGA2560(Arduino Mega)	ATMEGA328P-AU (Arduino uno)
External ROM	16MB Flash	N/A	N/A	N/A	N/A
External RAM	8MB PSRAM	N/A	N/A	N/A	N/A
GPIOs	33	50	30	86	23
Price (from lcsc.com)	US$5.0671	US$6.0328	US$1.1073	US$11.4005	US$2.1525
WIFI	2.4 GHz Wi-Fi (802.11 b/g/n)	N/A	N/A	N/A	N/A
BLE	Bluetooth® 5 (LE)	N/A	N/A	N/A	N/A

The device consisted of Source power, ESP32, STM32, and IO, as shown in Figure 7. Each of these microcontrollers has its distinct role and set of capabilities, making the board versatile and robust.

The ESP32 is equipped with a dual-core Xtensa® 32-bit LX7 microprocessor, capable of operating at a maximum frequency of 240 MHz. This offers ample computational power, particularly for complex matrix calculations conducive to multi-threaded processing. While FPGAs or GPUs could offer higher performance for such computations, they may not be as cost-effective given that the robotic arm doesn't require extreme levels of segmentation. Additionally, the ESP32 features 18 MBytes of SPI flash memory and 8 MBytes of SPI RAM, offering sufficient storage for this specific project, especially compared to other microcontrollers offering no more than 1 MByte of storage.

While the ESP32 does not natively offer robust SD card support, this limitation is mitigated by the STM32, which can be interfaced with the ESP32 via protocols like SPI or I2C for read/write operations on an SD card.

The power supply integrated into the board is impressively efficient capable of delivering up to 3 amperes of power within a compact footprint. This makes it effective even at lower levels of power consumption. Additionally, the board features clearly marked components and connections, facilitating ease of hardware configuration and programming.

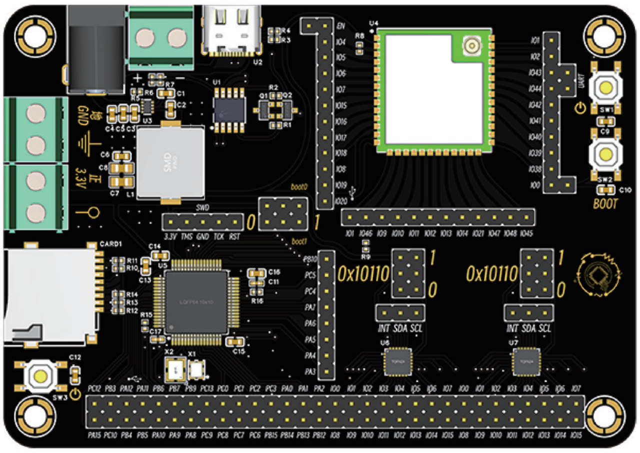


Figure 7. The device consisted of Source power, ESP32, STM32, and IO

In conclusion, the dual-controller architecture was utilized (ESP32S3 and STM32F446RET6). The ESP32S3 has dual cores and operates at a maximum frequency of up to 240 MHz, while the STM32F446RET6 operates at a maximum frequency of up to 180 MHz. In terms of storage, the ESP32S3 has up to 16 MBytes of SPI Flash and 8 MBytes of SPI RAM. Comparatively, the STM32F446RET6 offers 512 KBytes of Flash and 128 KBytes of SRAM and supports SD cards up to 4 GBytes. The board also has two 16-way GPIO expanders, providing up to 103 GPIO ports. The device supports Wi-Fi and BLE (low-power Bluetooth) for wireless connectivity. The power input range is 5~30V and provides up to 3A of 3.3V power output. The board features a low quiescent current, which helps reduce overall power consumption. Finally, the device has a clear and easy-to-read logo for easy hardware configuration and programming.

2.3. Path planning

Path planning is a crucial research area in robotics and automation systems (Yan et al., 2019). Especially for high-precision and complex tasks, path planning needs to ensure accurate movement from the start point to the endpoint and satisfy a series of motion constraints, such as maximum velocity, acceleration, and so on (Gonzalez et al., 2016). This study will use a linear position model combined with a Gaussian velocity model to achieve smooth and accurate path planning.

The path planning method in this study is obtained by rewriting and optimizing it based on the ABR Control library, a Python package mainly used for controlling and planning the paths of robot arms in real or simulated environments. The library provides APIs for the Mujoco, CoppeliaSim (formerly VREP), and Pygame simulation environments, as well as configuration files for one-, two-, and three-jointed models, as well as UR5 and Kinova Jaco 2 arms. Users can extend the package to run customized arm configurations (He et al., 2020).

The original code has been rewritten and optimized to better suit specific application requirements and performance metrics. Part of the path_planners code of the robotic arms is shown below:

```
Import numpy as np
from Robotic_control. path_planners import PathPlanner
from Robotic_control. path_planners.position_profiles import Linear
from Robotic_control.path_planners.velocity_profiles import Gaussian
# Initialize position and velocity profile
PosProfile = Linear()
VelProfile = Gaussian(dt=0.001, acceleration=1)
# Initialize the path planner
routePlanner = PathPlanner(pos_profile=PosProfile, vel_profile=VelProfile, verbose=True)
# Generate the path
routePlanner.generate_path(
    start_position=np.zeros(3),
    target_position=np.array([3, 4, 5]),
    start_orientation=np.array([0, 0, 0]),
    target_orientation=np.array([3.14, 0, 1]),
    initial_velocity=0,
    final_velocity=0,
    plot=True,
)
```

3. Results and Discussion

This section focuses on the robotic arm path planning algorithm based on linear position and Gaussian velocity models. The algorithm considers the three-dimensional spatial motion of the robotic arm from the initial position to the target position and the initial and target directions. The experimental results show that the algorithm effectively generates smooth, continuous, and physically constrained paths.

In robotic arm path planning problems, we must consider multiple variables and constraints. In this study, we chose the PathPlanner class from the Robotic_control library, which allows us to flexibly choose the planning algorithms for position and velocity. Specifically, we chose Linear as the location planning algorithm and Gaussian as the velocity planning algorithm.

3.1. Linear position model

Figure 1 shows in detail how the linear model works in practice. The figure contains two parts: the Moving Curve and the Interpolated Position Path. Both follow the planning principle of the linear model.

The Moving Curve part shows the object's behavior throughout the planning time. It is clear from the figure that the object moves smoothly and linearly from the initial position to the target position. The Interpolated Position Path: This section emphasizes the path of the object at different points in time. These points are computed by linear interpolation and fall strictly on the movement curve.

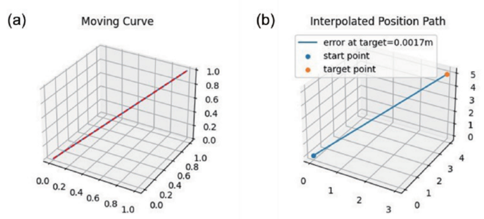


Figure 8 The robotic arms' set moving curve and interpolated position path.

3.2. Gaussian velocity model

Linear models are particularly suitable for systems with high real-time requirements and limited computational resources due to their simplicity and efficiency. However, its disadvantage is that path planning is limited to straight lines and does not apply to scenarios that require complex paths or obstacle avoidance capabilities.

A Gaussian model (Gaussian) is used to plan the speed of the robot arm along the path. The model allows us to set the maximum velocity and acceleration to generate a velocity profile that conforms to physical constraints, which benefits the robotic arms.

Figure 9 (a) depicts the 3D trajectory from the initial to the target position. Each point in the trajectory is accurately computed by Gaussian velocity modeling. The red, blue, and green markers indicate the starting and target positions on the x, y, and z axes. As can be seen from the figure, the trajectory exhibits significant smoothness in all dimensions, which helps to reduce mechanical wear and improve motion accuracy.

Fig. 9(b) shows the velocity variation of the robotic arm in the x, y, and z axes. The velocity gradually increases from zero to the maximum value in the form of an elegant Gaussian curve and then smoothly decreases to zero. Such a distribution of velocities helps to reduce the mechanical stress and increase the stability of the operation.

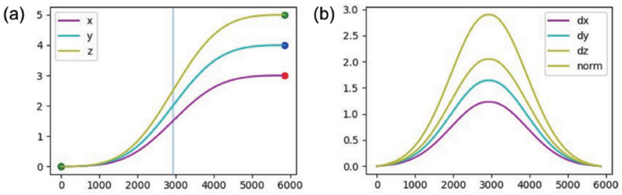


Figure 9. The Time-Dependent Position Curve and Velocity Profile of the Robotic Arms. (a) The position components along the x, y, and z axes over time. The red, blue, and green dots indicate the target positions along each axis. (b)The velocity components along the x, y, and z axes. It also shows the norm of the velocity vector as a purple line, providing an overview of the overall speed of the robotic arms.

If the path planning involves changes in direction, Fig. 10(a) will show how the direction angles (i.e., alpha, beta, gamma) change over time. Here, too, the change in direction is carefully planned by Gaussian modeling, thus ensuring smoothness and continuity throughout the process.

Figure 10(b) depicts the distribution of angular velocities (i.e., v_alpha, v_beta, v_gama) corresponding to the change in direction angle. Again, the changes in angular velocity are smooth and continuous, which is valuable in application scenarios where highly accurate and continuous angular velocity changes are required.

With Figures 9 and 10, this study comprehensively demonstrates the superior performance of the Gaussian velocity planning method in multidimensional trajectory generation. In particular, the Gaussian model shows its indispensable advantages regarding path smoothness and continuity.

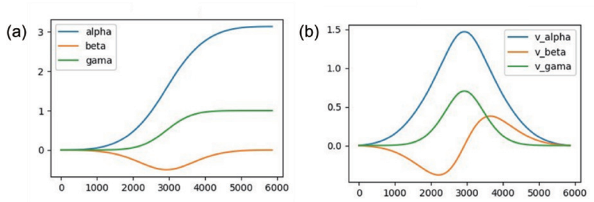


Figure 10. The Time-Dependent Orientation Changes and Velocity of the Robotic Arms. (a) The orientation angles (alpha, beta, gamma) change over time, providing insights into how the robotic arms adjust their orientation while moving. (b) The angular velocity components correspond to each orientation angle. It helps to understand how quickly the orientation is changing during the motion.

In path planning, visualization is a very important part, which can intuitively show the performance and effect of the algorithm. In order to better explain and analyze, we have drawn several diagrams, including a "Moving Curve" and "Interpolated Position Path".

Figure 11a shows the Given Moving Curve in 3D space. This figure shows a given moving curve. A linear path includes position changes on the x, y, and z axes.

Figure 11b shows the Interpolated Position Path. This figure uses a 3D curve to represent the path obtained by interpolation. Unlike the given linear path, this curve is smoother near the start and end points and accelerates faster in the middle part, with a shape similar to a Gaussian curve.

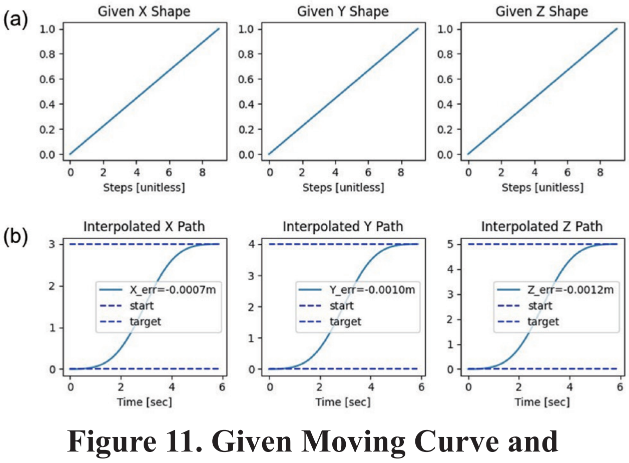


Figure 11. Given Moving Curve and interpolated Position Path in 3D space

By carefully analyzing these visualization results, we can understand the whole path planning process more intuitively and optimize it for different application scenarios. This provides strong support for further improving the accuracy and usability of path-planning algorithms.

Conclusion

This study centers on the comprehensive design and implementation of a highly flexible and accurate robotic arm for prosthetics and rehabilitation engineering applications. The robotic arm has a high degree of freedom and is easy to fabricate due to its modular design. Modeled using 3D printing technology and SolidWorks, the arm has three degrees of freedom, and the main idea is summarized as follows:

(1) The robot arm has a simple structure and uses a very small variety of structural components, making it cost-effective and easily scalable. Despite the simplicity of the structure, the arm meets the requirements of high mechanical strength and agility, making it adaptable to a variety of tasks and environments;

(2) The study also provided an in-depth analysis of the electronic circuit design, including microcontroller selection. After evaluating several types, the ESP32 and STM32 microcontrollers were chosen because of their optimal balance of computing power, I/O capabilities, and additional features such as Wi-Fi and Bluetooth. These controllers are programmed via Arduino and offer significant advantages in terms of both cost and performance;

(3) Besides the hardware design, this study proposes a robot arm path planning algorithm based on linear position and Gaussian velocity models. The algorithm is effective in generating paths that are smooth, continuous, and conform to physical constraints.

Bibliography

1. Dominijanni, G., Shokur, S., Salvietti, G., Buehler, S., Palmerini, E., Rossi, S., De Vignemont, F., d'Avella, A., Makin, T. R., Prattichizzo, D., & Micera, S. (2021). The neural resource allocation problem when enhancing human bodies with extra robotic limbs. Nature Machine Intelligence, 3(10), 850–860. https://doi.org/10.1038/s42256-021-00398-9

2. Gonzalez, D., Perez, J., Milanes, V., & Nashashibi, F. (2016). A review of motion planning techniques for automated vehicles. IEEE Transactions on Intelligent Transportation Systems, 17(4), 1135–1145. https://doi.org/10.1109/tits.2015.2498841

3. He, G., Kong, L., Yu, W., Jia, X., Yang, X., Xu, K., & Liu, C. (2020, October 17). A novel method for determining the spatial position of a mechanical arm based on an intelligent optimization algorithm. Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence. https://doi.org/10.1145/3438872.3439059

4. Hussain, I., Salvietti, G., Malvezzi, M., & Prattichizzo, D. (2015, September). Design guidelines for a wearable robotic extra-finger. 2015 IEEE 1st International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow (RTSI). https://doi.org/10.1109/rtsi.2015.7325071

5. Krausz, N. E., & Hargrove, L. J. (2019). A survey of teleceptive sensing for wearable assistive robotic devices. Sensors, 19(23), 5238. https://doi.org/10.3390/s19235238

Maity, A., Roy, K., & Gupta, D. (2019). Designing, 3d printing of a quadruped robot and choice of materials for fabrication. ArXiv.Org.

6. Malvezzi, M., Iqbal, Z., Valigi, M. C., Pozzi, M., Prattichizzo, D., & Salvietti, G. (2019). Design of multiple wearable robotic extra fingers for human hand augmentation. Robotics, 8(4), 102. https://doi.org/10.3390/robotics8040102

7. Mick, S., Lapeyre, M., Rouanet, P., Halgand, C., Benois-Pineau, J., Paclet, F., Cattaert, D., Oudeyer, P.-Y., & de Rugy, A. (2019). Reachy, a 3D-printed human-like robotic arm as a testbed for human-robot control strategies. Frontiers in Neurorobotics, 13. https://doi.org/10.3389/fnbot.2019.00065

8. Moran, M. E. (2007). Evolution of robotic arms. Journal of Robotic Surgery, 1(2), 103–111. https://doi.org/10.1007/s11701-006-0002-x

9. Siemasz, R., Tomczuk, K., & Malecha, Z. (2020). 3D printed robotic arm with elements of artificial intelligence. Procedia Computer Science, 176, 3741–3750. https://doi.org/10.1016/j.procs.2020.09.013

10. Yan, F., Razavi, S., Habibi, S., Divakarla, K. P., & Emadi, A. (2019). A review of autonomous vehicle technology landscape. International Journal of Electric and Hybrid Vehicles, 11(4), 320. https://doi.org/10.1504/ijehv.2019.10024315