ATLANTIS PRESS
*Advances in Intelligent Systems Research, volume 164*
International Conference on Mathematics, Modeling, Simulation and Statistics Application (MMSSA 2018)

# Design Method and Example of a Simple Educational Robot

Tongxing Jin, Wei Xie*, Jiachen Ma and Kuangzheng Ye
School of Information Science and Engineering, Harbin Institute of Technology at WeiHai, 264209, China
*Corresponding author

***Abstract*—Nowadays educational robots have begun to walk into campuses and households, changing the traditional way of teaching and learning. There are many physical or virtual robot education platforms on the market now, whose perfect systems and mature hardware or software not only reduced the difficulty of learning but also limited the creativity of users. Taking this as a starting point, this paper proposed a method of designing a simple robotic arm system, including motion analysis and control algorithm design, and verified it through an example. Generally, the method proposed in this paper can help robot enthusiasts, especially beginners, to explore and practice the process of building robots just with simple materials.**

***Keywords—educational robot; robotic arm; steering gear; fuzzy PID; musician robot***

## I. INTRODUCTION

The development level of robot technology is an important manifestation of a country's scientific and technological innovation capability and industrial competitiveness. With the development of science and technology, robots have begun to enter the educational environment, changing the traditional teaching methods. Educational robots have gradually aroused people's interest and gained more and more attention, the *Development Planning for a New Generation of Artificial Intelligence* [1] proposed by the State Council of P.R.C in 2017 clearly states that national intelligence education should be implemented. At present, many primary and middle school students have carried out robot education. It is estimated that in 2021, the global educational robot market will reach 11.1 billion US dollars [2], which shows the huge market prospects of educational robots. However, robots design is a complex and systematic work, which combines many advanced technologies such as mechanics, electronics, computer hardware and software, sensors and intelligent control algorithms. Therefore, how to design a specific robot system quickly and effectively still faces many problems. Despite this, some current open source software and hardware modules [3] and the ubiquitous learning model in the Internet environment provided conditions for robot enthusiasts to design robots.

Carnegie Mellon University first explored the use of educational robots as an effective way to cultivate students' STEM (Science, Technology, Engineering, Mathematics) literacy, emphasized the important role of empirical research, and provided learners with a wealth of practical examples of robot teaching [4]. In order to allow students at different levels to fully engage with robotics, Matsushita et al. designed a fast, inexpensive robotic application toolbox [5]. There are many platforms for robot education at present [6-8], including both physical robots and virtual robots (including simulation and game software), such as LEGO in Denmark, FISCHER in Germany, and Uptech Harmony Ltd robot development kit from China, etc. However, the integrity of the platform and the limitations of hardware and software often limit the user's creative ability. To improve students' STEM literacy in all aspects, we must consider the design methods of educational robots from the above four dimensions and guide students to start from the bottom. This article takes this as an entry point, focusing on college students and robot enthusiasts, and proposes a design method for a simple educational robot (robotic arm). In section II, a simple robotic arm mechanism and its modeling analysis will be elaborated, then the design method of joints especially its control algorithm will be introduced in detail in section III In the specific design process, this paper combines theoretical research with engineering practice, and gives a living example to implement a musician robot in IV, followed by the conclusion in section V.

## II. SIMPLE ROBOTIC ARM MECHANISM AND MODELING ANALYSIS

Like the human arm, a robotic arm mechanism consists of arms (links) and joints (steering gears). The modeling and kinematics analysis of the manipulator structure studies the mapping relationship between the end position of the arm and the output of every joint. Generally speaking, it can be divided into two different sub-problems: the forward and the reverse problem.

If we take vector $\theta$ as the output of all joints in an arm, $p$ as the pose of arm end, then the forward problem can be expressed as $p = f(\theta)$. Given the output of each joint, it solves the pose of the arm in three-dimensional space. However the inverse problem solves the problem of $\theta = f(p)$, that is, when the pose of the arm end is known, the output angle of each joint should reversely be solved. Obviously, the reverse problem is more complicated but more practical, in this section we will take a simple three-axis robotic arm as an example to introduce its kinematic and geometric modeling methods.

### A. Simple Robotic Arm Mechanism

A typical tandem robotic arm consists of a set of links (rigid bodies) that meet end to end through joints. Each joint in the

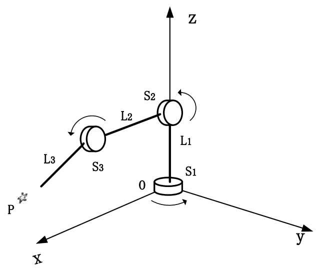


FIGURE I. A 3-AXIS ROBOT ARM

robotic arm provides a rotational or translational freedom to the arm, and their output changes the relative position and orientation between adjacent two links. Therefore, as long as the robotic arm has enough degrees of freedom, it is possible to control the end of the arm to reach any position in the three-dimensional space just by adjusting the output angle of each joint.

Take the simple robotic arm in figure I for example. The robot arm is composed of three joints $S_1$, $S_2$ and $S_3$, three links $L_1$, $L_2$ and $L_3$. Joint $S_1$ is fixed on the ground, and its output axis ($L_1$) coincides with the z-axis and is fixed to joint $S_2$. Joint $S_2$ and $S_3$ are in the same vertical plane as the whole robot arm, and their output axes are all parallel to the horizontal plane, so that they can cooperate with each other to adjust the angle between $L_1$ and $L_2$, as well as $L_2$ and $L_3$, thereby driving the end of the arm move in a vertical plane.

### B. Kinematic Principles

The kinematics is to study the mapping relationship between the motion of the multi-link (rigid body) structure and the output of the arm without considering the mass of the manipulator and the magnitude of the force and moment. Its analytical methods are mainly divided into two categories: analytical method and graphical method.

The analytical method, also named D-H parameter method, is a method proposed by Denavit and Hartenberg in 1955 to describe the geometric relationship between the links and joints in a tandem robot arm. It uses the link length $a_j$ and the torsion angle $\alpha_j$ to represent the *j*-th link, link offset $d_j$ and the joint angle $\theta_j$ to represent the *j*-th joint, so the transformation relationship between the adjacent two joints could be determined by these four parameters. In the process of solving the inverse problem, it is necessary to carry out the inverse motion decomposition of Euler transform or RPY (roll, pitch and yaw) transform frequently. The kinematics equation is very complicated, and the computer operation precision and speed are mutually restricted, so it is not friendly to beginners.

The graphical method is to simplify the system into simple geometric figures and use mathematical theorem such as Pythagorean theorem to solve the position and orientation relationships between the mechanical links. This method is intuitive and easy to understand, but it is strictly limited to some simple arm structures in practical applications. However, for the simple three-axis robotic arm shown in figure I, the graphical method can complete the kinematic analysis of it easily, and the modeling process of the robotic arm will be elaborated below.

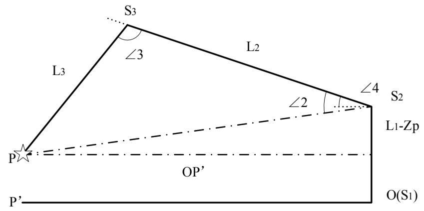


FIGURE II. SCHEMATIC DIAGRAM OF VERTICAL PLANE

### C. Geometric Modeling Method

To explain the mapping relationship between the position of the arm end and the output of each joint, we take point P in 3-dimensional space as the target point. In order to make the arm reach the point P, joint $S_1$ is first driven to keep point P in the same vertical plane as the robot arm, then joint $S_2$ and $S_3$ will help the arm end move freely in the vertical plane until it reaches the target point. The output angle of the three joints are calculated as follows:

- With the projection point of point P on plane oxy, it is easy to figure out the required output of joint $S_1$.
- Then joint $S_2$, $S_3$, and point P are in the same vertical plane, and the schematic diagram of the plane is shown in figure II, where P' is the projection of P in the horizontal plane. With the coordinates of P $\left(x_p, y_p, z_p\right)$ and $S_2\left(0,0,L_1\right)$, the Euclidean distance from the target point P to joint $S_2$ can be easily obtained. Then, the lengths of the three sides of triangle $\Delta PS_2S_3$ are known, so it is easy to solve $\angle 2$ and $\angle 3$ with cosine theorem as is shown in (1) (2).

$$\angle 2 = \arccos \frac{PS_2^{\ 2} + L_2^{\ 2} - L_3^{\ 2}}{2 \times PS_2 \times L_2}. \tag{1}$$

$$\angle 3 = \arccos \frac{L_2^{\ 2} + L_3^{\ 2} - PS_2^{\ 2}}{2 \times L_2 \times L_3}. \tag{2}$$

- In the geometric sense, we have established a mathematical model of the system, however, in order to control the end of the arm to a certain position, the real output of steering gears is related with the practical installation. For joint $S_2$ whose output range is -150° to 150°, if its median position (output angle with none input) is parallel to the horizontal plane, then its output actually should be the angular difference between median position and the target position, that is, $\angle 4$.

Thus, given the position of the arm end, the output of all joints in the arm could be solved by the graphical method discussed above. On the contrary, if we know the angles of each pair of links, the position of arm end could also be figured out exactly. Furthermore, if the three attitude angles of the arm end are required, more joints have to be introduced into the arm to improve the degree of freedom, which is beyond the scope of this article.

## III. THE DESIGN OF JOINTS

Just like the joints of an arm, steering gears are the actuator that drives the movement of the manipulator. It is essentially a closed-loop position (angle) servo system consisting of a DC (Direct Current) motor, and suitable for conditions where the angle is changing constantly and can be maintained if needed. According to the input signal, the steering gears are mainly divided into analog steering gears and digital steering gears.

The input signal required by the analog servo steering gear is PWM( Pulse Width Modulation) signals, and it occupies a PWM port of the controller. If there are multiple steering gears, the controller needs to have enough output ports for PWM signals. However, digital steering gears can receive digital signals, and they are usually connected through a bus. Compared with the analog steering gear, the it has high control precision and adaptability, more importantly, it does not occupy the PWM interface of the controller, so it is more suitable for the control of robot arm that contains multiple actuators.

Distributed system connects the main controller with many digital steering gears through a bus. In actual working process, the controller issues control commands of output angle and speed, then the data is transmitted through the bus after being encoded. After receiving the data, steering gears could decode the data and respond to the command corresponding to its serial number. In this structure, the controller can easily control any actuator on the bus and it is very convenient for the system to add or delete a steering gear if needed. Therefore, the distributed architecture was also applied in this research to achieve the control of a robotic arm.

### A. Principles of Steering Gears

A steering gear is essentially a closed-loop position (angle) servo system, composed of a controller unit, a motor driver module, a DC motor, a gear reducer, position sensors, circuit protection module, and so on.

As is shown in figure III, the steering gear control unit accepts the control command from the main controller and controls the DC motor to drive the motion of links. In this process, the position sensor such as potentiometer detects the actual output angle of the steering gear in real time, and then their output is converted into a digital signal and fed back to the control unit. After comparing the feedback signal with the desired output, the control algorithm in the controller would adjust the output in real time according to the error, thereby the output of the system is controlled at the specified position.

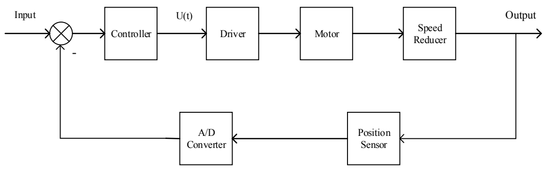


FIGURE III. WORKING PROCESS OF A STEERING GEAR

### B. Control Strategy of Steering Gears

The control algorithm is the core of the closed-loop position servo system. Its function is to adjust the input value of the controlled object in real time according to the difference between actual output and target output. PID (Proportion, Integration, and Derivation) control strategy is one of the earliest developed control strategies, and because of its simplicity, good robustness, and high reliability, it is widely used in industrial environments, especially for deterministic control systems that can establish accurate mathematical models.

PID defines the difference between the feedback *c(t)* and the desired output *r(t)* as the system error *e(t)*, and linearly combines *e(t)* with proportion, integration, and differentiation to obtain the input signal of controlled objects, its control laws can be expressed as:

$$u(t) = K_P[e(t) + \frac{1}{T_I}\int_0^t e(t)dt + \frac{T_D de(t)}{dt}]. \quad (3)$$

$K_p$ is the proportionality coefficient, $T_i$ and $T_d$ are the integration time and the derivative time respectively. The proportional element of the PID controller can reduce the deviation *e(t)* proportionally in real time, while the integral element is mainly used to eliminate the static error and improve the system type. The differential part can reflect the change trend of the deviation signal, thereby speeding up the system's response and diminishing the setting time.

Modern PID control often uses a computer as its controllers, which is also called digital PID control. It is a kind of sampling control and can only obtain the deviation value at the sampling time, so the integral and derivative term in (3) should be discretized as follows:

$$u(k) = K_P e(t) + K_I T \sum_{j=0}^{k} e(j) + K_D \frac{e(k) - e(k-1)}{T}. \quad (4)$$

In order to meet some specific needs in practice, PID control algorithm has derived many variants, such as PID algorithm with integral separation or with a dead band, which have significant effects in reducing overshoot, suppressing oscillation, and shortening setting time. However, they all require designers to get the estimated value of *K* through complex calculations or experiments, which poses high requirements for inexperienced beginners. Therefore, this paper adopts the strategy of fuzzy adaptive PID control method.

Based on error *e* and the rate of change *et*, fuzzy adaptive PID control algorithm could adjust the values of the three parameters $K_p$, $K_i$, $K_d$ through a certain fuzzy rule. There are three main steps for fuzzy adaptive PID control algorithm: fuzzification, fuzzy inference, and defuzzification. The adjustment of the three parameters of the PID controllers $K_p$, $K_i$, $K_d$ mainly follows the following principles:

- When $|e|$ is big, $K_p$ should be enlarged to obtain a fast response, $K_i$ should be small enough to avoid big overshoot.

- When $|e|$ and $|et|$ are medium, we should reduce the value of $K_p$ to diminish the overshoot.
- When $|e|$ is small, the value of $K_p$ and $K_i$ should be enlarged to get good static performance. If $|et|$ is also small, it means the response gets close to steady state and $K_d$ should be enlarged; otherwise, $K_d$ should be smaller.
- When $|et|$ is large, $K_p$ should be diminished and $K_i$ ought to be enlarged.

Combined with the above tuning principles, this method has a series of fuzzy rules, for example:

If(e is PM)and(ec is NB)then(kp is PS)(ki is Z)(kd is PB)

The rule means that if the error is large, but the error is decreasing rapidly, then $K_p$ should take a relatively small positive value, $K_i$ should take *0*, and $K_d$ should take a larger positive value. The output of fuzzy inference is still fuzzy, so it should be defuzzified to get clear output.

The fuzzy adaptive PID control algorithm can change its parameters in real time according to system error and is more flexible compared with other PID variants. Furthermore, the fuzzy rules are close to our language, so beginners can design or adjust a fuzzy adaptive PID algorithm only with fundamental knowledge of PID. Therefore, this paper chooses the fuzzy adaptive control algorithm to implement the position servo of DC motor, which can achieve passable control performance and is fitted for beginners to learn, understand and master.

### C. Simulation of Control Algorithm

In this paper, we selected an appropriate DC motor and drive device. By consulting the parameters such as the mechanical and electrical time constant $T_m$, $T_e$ of the motor, we established the block diagram for the closed-loop control system of the steering gear which is shown in figure IV, and the model was also simulated in MATLAB.

Step signal was added to the system at time 0, and its response was shown in figure V. Under the action of the controller, the output of the system becomes stable within 2% error within 9ms, and the error is reduced to 0 after 17ms with an overshoot of 7.14%. Therefore, the control algorithm designed in this study can meet the requirements of robotic arm control in both static and dynamic performance, and the design

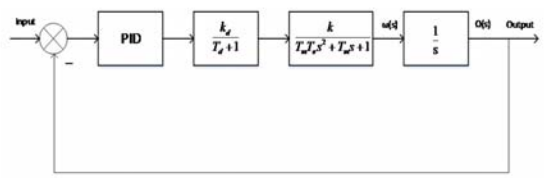


FIGURE IV. BLOCK DIAGRAM OF STEERING GEAR SYSTEM

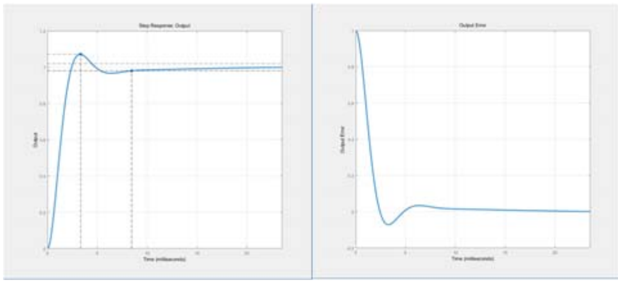

FIGURE V. OUTPUT AND ERROR OF SYSTEM

method is relatively simple, suitable for beginners to master.

## IV. CASE STUDIES

In order to verify the effectiveness of the modeling method and control strategy discussed before, this paper has built a three-axis robotic arm that has the function of playing an electronic organ and the arm covers the keys within an octave in sequence. In this section, the actual experiment will be carried out.

### A. Control Performance Verification

According to the method discussed before, by modeling and simulating the internal closed-loop control system of the CDS5500 steering gear, we have determined a set of optimal combination of $K_P$, $K_i$, and $K_d$, and the designed algorithm was downloaded to an actuator. In order to verify the stability, accuracy, and rapidity of the control algorithm, the following experimental verification was carried out.

When the steering gear output was 30°, we gave it an instruction to output an angle of 270°, at the same time the output of the potentiometer was sampled every 3ms. We took time as the horizontal axis, output error as the vertical axis and obtained figure VI.

As can be seen from the figure, the output error of the steering gear was reduced from 756 to 0 after 1.2s, and stable within ±0.3° range, that means, the setting time is 1.2s and the steady-state error is 0.3°. In the process of playing an electronic organ, the maximum adjustment angle of the actuator is always less than 5°, so the setting time is not more than 25ms, and the steady-state error is within 0.3°, which indicates that the algorithm can fully meet the requirements of stability, accuracy, and rapidity of the control system.

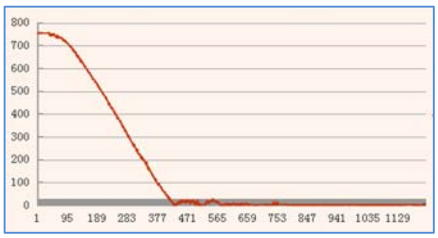


FIGURE VI. OUTPUT ERROR

## *B. Modeling Analysis of the Musical Robotic Arm*


FIGURE VII. MUSICIAN ROBOTIC ARM

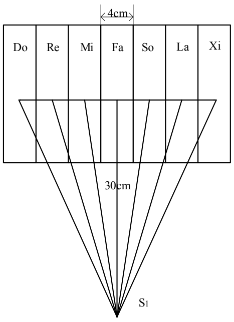


FIGURE VIII. OVERHEAD VIEW

The simple three-axis robot arm constructed according to section II is shown in figure VII. The fixed point of the arm is 30 cm away from the keyboard, and the lengths of the links are 8 cm, 20 cm, and 23 cm, respectively. The median position of the first steering gear corresponds to Fa, and the arm is parallel to the table when the second and third actuator have none input. The output range of the steering gear is from 0° to 300°, and the digital input is between 0 and 1023. The keyboard is 5cm away from the desktop, and a single key is 19.2 cm long and 2.4 cm wide.

The overhead view of the system is shown in figure VIII. In order to control the arm to touch the center of seven different keys, the output and required input of the first steering gear can be solved easily according to the geometric relationship, as well as the horizontal distance between the first steering gear and the target point. For example, in order to touch the key Mi, the angle between its median and the target direction is *arctan* 2.4/30 = 4.5739°, so it has to output an angle of 154.5739°. The statistical results of all keys are shown in table II.

According to the geometric modeling method introduced in section II, combined with the representation in figure II, we sequentially use the center of each key as the target point to calculate output angles and digital inputs of the second and third steering gears. The statistical results are shown in table I.

In order to play a certain note, the designer just needs to check the table to determine the digital input corresponding to each steering gear and program it into the controller according to the order of the tune. Therefore, the process is friendly for beginners with no experience.

TABLE I. STEERING GEAR S2 S3

| | **Do** | **Re** | **Mi** | **Fa** | **So** | **La** | **Xi** |
|---|---|---|---|---|---|---|---|
| **∠2(°)** | 40.2 | 40.9 | 41.4 | 41.5 | 41.4 | 40.9 | 40.2 |
| **∠3(°)** | 92.0 | 90.2 | 89.1 | 88.8 | 89.1 | 90.2 | 92.0 |
| **∠4(°)** | 84.4 | 84.4 | 84.3 | 84.3 | 84.3 | 84.4 | 84.4 |
| **Output $S_2$(°)** | 115.4 | 114.7 | 114.3 | 114.2 | 114.3 | 114.7 | 115.4 |
| **Input $S_2$** | 394 | 392 | 390 | 390 | 390 | 392 | 394 |
| **Output $S_3$(°)** | 238.0 | 239.8 | 240.9 | 241.2 | 240.9 | 239.8 | 238.0 |
| **Input $S_3$** | 812 | 819 | 822 | 824 | 822 | 819 | 812 |

TABLE II. STEERING GEAR S1

| | **Do** | **Re** | **Mi** | **Fa** | **So** | **La** | **Xi** |
|---|---|---|---|---|---|---|---|
| **Output $S_1$ (°)** | 163.5 | 159.1 | 154.6 | 150.0 | 145.4 | 140.9 | 136.5 |
| **Input $S_1$** | 558 | 543 | 527 | 512 | 497 | 481 | 466 |
| **Distance $PS_1$ (cm)** | 30.9 | 30.4 | 30.1 | 30.0 | 30.1 | 30.4 | 30.9 |

## V. CONCLUSION

In order to help beginners learn the design method of robot arms, this paper introduced the mechanism composition and kinematics of simple robotic arms, and expounded the method of modeling by geometric method. On this basis, the basic principles of steering gears, especially the designing method of control algorithm was described in detail and verified by simulation. This paper also has built a robot arm that can complete the function of playing an electronic organ. The performance of the control algorithm were verified by experiments, and the motion analysis of the system was also carried out, so readers can determine the input of every actuator just by referring to the table. The design methods and the example of this paper are feasible and practical, so it may help readers, especially beginners, to explore and practice the process of constructing robot arms independently.

## ACKNOWLEDGMENT

This research was financially supported by "the china" (Grant No.HIT.NSRIF.2016106) and Ministry of Education Cooperative Education Project of china (Grant No. 201702177001).

## REFERENCES

[1] Yonghe Wu and Tongtong Li, "Development status, practice, reflection and prospect of robot education from the perspective of machine intelligence," Journal of Distance Education, vol. 4, pp. 79-87, April 2018.

[2] Dejian Liu, Huairong Huang, Nianxing Chen, Lei Fan, Guoren Zhao, Mengkun Shen et al., Educational Robots White Paper 2016: The Global Development, Beijing: Smart Learning Institute of Beijing Normal University, September 2016.

[3] Takshi Kuremoto, Shigo Mabu, Kunikazu Kobayashi and Masanao Obayashi, "Creative design of robotics education using LEGO," International Journal of Engineering Innovation and Management, vol. 5, pp. 20-25, 2015.

[4] Juan Wang, Lailin Hu and Lida An, "Case study of educational robot course integrating STEM in foreign countries——Taking the robotic course of Carnegie Mellon University Robotics Academy as an Example", Modern Educational Technology, vol. 4, pp. 33-38, 2017.

[5] Matsushita Kojiro. "Quick, cheap, and creative development for robotics education: Understanding and experiencing prosthetics technology," Journal of Robotics and Mechatronics, vol. 5, pp. 850-858, 2011.

[6] Banduka Maja Lutovac, "Robotics first-a mobile environment for robotics education," International Journal of Engineering Education, vol. 2, pp. 818-829, 2016.

[7] Lujin Wang and Zhenyu Wang, "Design of a low-cost educational robot", China New Telecommunications, vol. 4, pp. 195, 2018.

[8] Deniz Cengiz, Cakır Mustafa. A novel designed interactive training platform for industrial robot offline programming and robotics education, International Journal of Robotics and Automation, vol. 6, pp. 665-672, 2017