Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 4, No. 1, 2025

Development of a Virtual Simulation System for Industrial Robot Kinematics Based on Unity 3D

Elias Sorensen

University of Alberta, Edmonton, Canada

EliasSorensen@ualberta.ca

Abstract: With the rapid expansion of industrial robotics driven by the "Made in China 2025" initiative, there is an urgent demand for innovative and cost-effective training methods to address the significant talent gap in related industries. This study presents the design and implementation of a virtual simulation system for industrial robot motion, using the ABB IRB 120 robot and the Unity 3D engine. The proposed system replicates real robot kinematics with high accuracy while offering cost-effective and reusable training solutions. Experimental tests demonstrate that the platform supports both forward and inverse kinematics with efficient and user-friendly operation. The system is valuable for enhancing human-robot interaction and operator training across industrial settings.

Keywords: Virtual Simulation; Industrial Robots; Unity3D; Kinematics.

1. Introduction

To cope with the needs of the new round of technological revolution and industrial transformation worldwide, and to accelerate the transition from a manufacturing country to a manufacturing powerhouse, China has proposed the "Made in China 2025" strategy, which has led to the increasingly wide application of industrial robots in various fields. According to statistics from the International Federation of Robotics (IFR), China has ranked first in the world in terms of the stock of industrial robots for eight consecutive years since 2013[1]. In 2021, the annual installation of industrial robots in China increased by 51% year-on-year, resulting in a talent gap of up to 5 million in related industries[2,3], and enterprises need to pay high training costs for employee skill development. With the rapid development of internet technology and computer capabilities, robot virtual simulation technology can effectively overcome the limitations of equipment and venue in talent training[4], and can meet the industry's demand for cultivating more relevant talents, avoiding the lack of practical operation opportunities for operators due to the high cost of equipment and venue restrictions in training. In this paper, based on the Unity 3D engine and using the ABB IRB 120 industrial robot as the object of study, we research virtual simulation and motion control of industrial robots, and design a robot kinematics simulation system that maintains high consistency with the motion of real robots, and has the advantages of low cost and reusability.

2. System Functional Structure Design

To achieve efficient operation of industrial robots and visualized monitoring of their motion status, the Unity 3D engine is adopted for the development of the robot system. The development of the virtual simulation system consists of two main parts. The first part is the scene area, which includes the construction of the entire scene, including: 1) System scene building: industrial robot model construction, lighting effects, shadow effects, scene baking. 2) UI interface design: start menu and buttons for different functional modules. 3) Camera control: implementation of scene roaming, allowing users to observe the robot's status from different angles. The second part is the functional area, which includes the following two contents: 1) Forward and inverse kinematics calculation: to achieve a better reproduction of the real robot's motion posture in the virtual simulation. 2) Motion control: the virtual system allows operators to control the robot through computer input devices. The implementation of system functions is mainly through the development of functional modules using C# language. The functional structure of the system is shown in Figure 1.

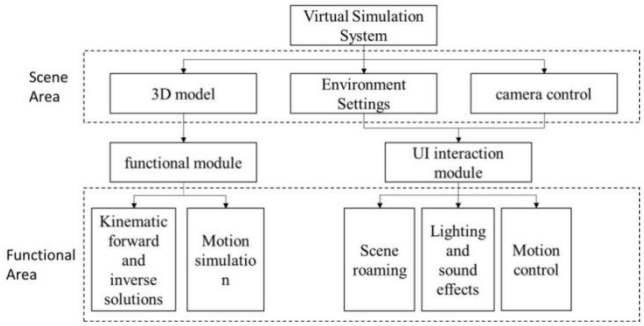

Figure 1. Functional Module Illustration of Virtual System

3. 3D Model Creation

3.1 Creation of 3D Model for the Robot

The creation of a 3D model for an industrial robot is the foundation for realizing the virtual simulation system. On one hand, in order to enhance the realism of the virtual industrial robot model and accurately represent the pose information of the real robot, the industrial robot needs to be accurately modeled in a 1:1 scale, including texture processing and lighting rendering. On the other hand, in order to lightweight the virtual industrial robot model, it is not necessary to model the internal parts of each axis, only the external surfaces of each joint need to be processed. To meet these requirements, this paper first uses SolidWorks to create models of the six joints of the robot with accurate dimensions, based on measurements of the dimensions of the real robot, and models the six axes according to real dimensions. Then, the assembly function of SolidWorks is used to assemble the six joints of the robot to obtain a complete industrial robot model. Since the virtual model format created by SolidWorks cannot be read by Unity, it is necessary to use 3Ds Max software for format conversion and lightweight processing. In 3Ds Max, the model is optimized by deleting redundant faces to reduce the CPU burden and achieve the goal of lightweighting the model[5]. The pivot positions of each joint are adjusted for control in Unity[6]. As the Y-axis in the world coordinate system of Unity engine is upward direction in a left-handed coordinate system, while the world coordinate system of 3Ds Max software is Z-axis upward, it is necessary to convert the Z-axis and Y-axis during the export from 3Ds Max to .FBX format[7,8]. The model exported in .FBX format has good compatibility with Unity engine and does not result in loss of part details. Material rendering and environmental lighting settings for the robot model are performed in Unity. The specific process of industrial robot modeling is shown in Figure 2.

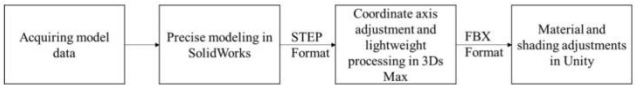

Figure 2. Robot Modeling Workflow Diagram

3.2 Basic Principles of Motion Simulation

In the motion process of a real robot, there is a parent-child relationship between the six axes of the robot. The rotation of each joint axis is relative to the parent object. Therefore, in order to achieve the effect of the virtual robot end-effector following the rotation of the joints, Unity's parent-child relationship is utilized, as shown in Figure 3. The hierarchy in Unity includes parent nodes and child nodes, and child nodes can also contain other child nodes[9] .The characteristics of the parent-child relationship are that when the parent object moves or rotates, the child objects will move or rotate accordingly, but when the child objects move or rotate, it will not affect the position and state of the parent object. By correctly setting the hierarchy relationship between the axes, a model tree of the robot is established. Through scripting to control the rotation direction and speed of each axis, various robot motions can be initially presented.

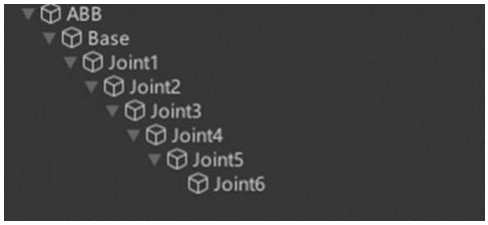

Figure 3. The hierarchical relationship between joints

4. Robot Kinematics Analysis

4.1 Robot Kinematic Modeling

The ABB IRB 120 robot is a serial-link type industrial robot with six rotational joints. The motion of the robot is achieved by rotating each joint to different angles to transform the position and orientation of the end effector. In order to describe the relationship between the joint variables of the robot and the pose of the end effector, the Denavit-Hartenberg (DH) method is used to establish the kinematic model of the robot. In the process of DH parameterized modeling, the establishment of coordinate systems can be done using either the standard DH method or the modified DH method. The main difference between these two methods lies in the definition of the coordinate systems for the links: the coordinate systems of the standard DH method are defined at the back end of the links, i.e., the end farthest from the base. On the other hand, the coordinate systems of the modified DH method are defined at the front end of the links, i.e., the joint closer to the base. In this paper, the standard DH method is adopted to establish the coordinate systems of the robot links, as shown in Figure 4. Fixed reference coordinate systems are established at the joint locations of the links, and the pose relationship between adjacent links is described using 4x4 homogeneous transformation matrices, which are then used to derive the pose relationship of the end effector relative to the fixed reference coordinate system [10].

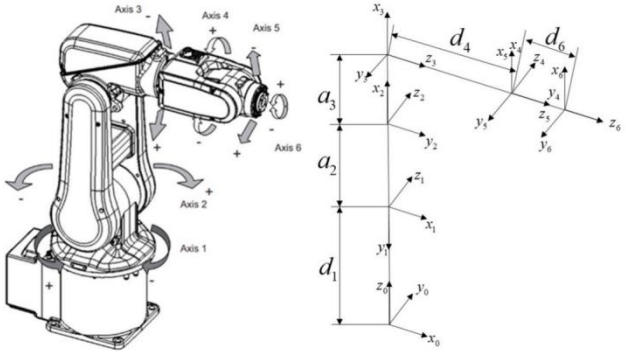

Figure 4. ABB IRB 120 robot link coordinate systems

In the figure, d_1 =290mm, a_2 =270mm, a_3 =70mm, d_4 =302mm, d_6 =72mm. The meanings of various parameters in the D-H (Denavit-Hartenberg) method are as follows:

θ_i :The angle of rotation from axis X_{i-1} to axis X_i , with positive rotation defined as rotating in the positive direction about axis Z_{i-1} .

d_i :The distance between axis X_{i-1} and axis X_i along the direction of axis Z_{i-1} , with positive distance defined as moving in the direction of axis Z_{i-1} .

α_i :The angle of rotation from axis Z_{i-1} to axis Z_i, with positive rotation defined as rotating in the positive direction of axis X_i.

a_i :The distance between axis Z_{i-1} to axis Z_i, with positive value defined as pointing in the direction of axis X_i.

The parameters of the robot in the home position are shown in Table 1.

Table 1. Link parameter of IRB 120

Connecting rod i	α_i (°)	d_i (mm)	θ_i (°)	a_i (mm)
1	0	290	-90	0
2	-90	0	0	270
3	0	0	-90	70
4	0	302	90	0
5	0	0	-90	0
6	0	72	0	0

4.2 Inverse Kinematic of Robot

Inverse kinematics of a robot refers to the calculation of a series of joint angle variables that satisfy the desired requirements, given the pose of the end-effector of the robot. It has wider applications in practical production compared to forward kinematics. The solution of inverse kinematics is usually the solution of a nonlinear system of equations[11], making it complex and non-unique. Due to the different mechanical structures of robots from different manufacturers, and the different coordinate systems established for linkages, there are multiple methods for solving inverse kinematics[12]. In motion simulation systems with high real-time requirements, the cyclic coordinate descent (CCD) method, which has lower computational complexity and faster computation speed, is more suitable for solving the inverse kinematics problem[13]. CCD is an iterative method that starts from the position of the end-effector of the robot, and adjusts the angle of each joint one by one. For each joint, the error of the end-effector's orientation in that joint direction is calculated after each iteration, and then refined until the error in the end-effector's orientation in that joint direction is less than a preset threshold or reaches the maximum iteration times, in order to minimize the difference between the end-effector's orientation and the desired orientation.

As shown in Figure 5, the iterative process of cyclic coordinate descent (CCD) method is explained with an example of two joints and one end-effector. Where P_e is the current position of the end-effector, P_f is the position of the target point, J_1 and J_2 are the first and second joint angles, respectively. V_{i-e} is the vector from the i joint to the end-effector, and V_{i-f} is the vector from the i joint to the target point. As shown in Figure 5(a), first, calculate the angle α_1 between the line connecting the joint and the target point and the line connecting the joint and the end-effector, and rotate the joint J_1 to the position shown in Figure 5(b) to bring the joint end closer to the target point.

Then calculate the angle α_2 between the line connecting joint J_2 and the end-effector and the line connecting joint J_2 and the target point, and rotate joint J_2 to the position shown in Figure 5(c).

Repeat this process multiple times, as shown in Figure 5(d), the distance between the robot's end- effector and the target point will reach its minimum value[14].

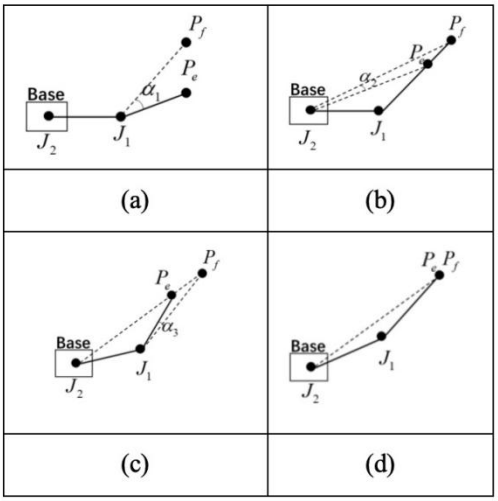

Figure 5. The iterative process of cyclic coordinate descent (CCD) method

5. Robot Motion Simulation Experiment

The design of the user interface (UI) is a crucial step in the development of a robot motion simulation system. The UI serves as a medium for interaction between users and the system, and a well-designed UI can make the interaction between operators and robots simpler and more natural. In the robot kinematic control UI of this system, as shown in Figure 6, a button-based approach is used to simplify the control of robot motion and improve interaction efficiency. The buttons on the control panel are correspondingly mapped to each axis of the robot. The "+" and "-" buttons are used to control the forward and backward motion of the robot joints, respectively. For example, when the user presses the "-" button of Joint1, the first axis of the robot will move in a clockwise direction.

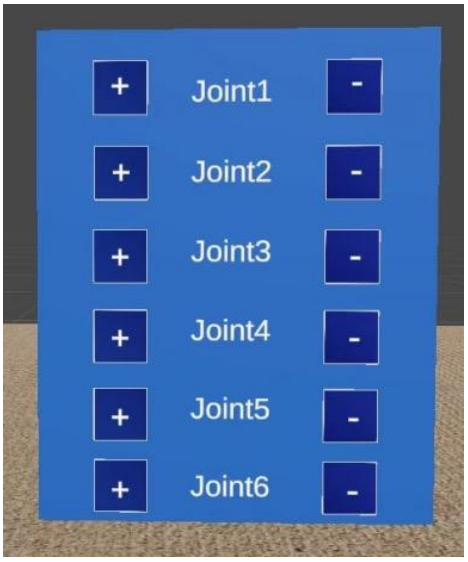

Figure 6. Forward kinematics control UI interface

The system uses a clickable and draggable sphere as the target point, and places the target point at the TCP (Tool Center Point) of the end effector, combined with the inverse kinematics (IK) calculation using the cyclic coordinate descent (CCD) algorithm to achieve the robot's inverse kinematics during the operation, as shown in Figure 7. The IK target point for the robot is represented by a blue sphere, and the operator can drag the target point to any position within the workspace. On the UI interface's inverse kinematics display panel, the operator can see the rotation angles of each axis during the IK calculation using the CCD algorithm, which helps the operator better understand the current posture of the robot in space, enhances operability, and provides a better interactive experience. Through experimental testing, the system has been proven to simulate the kinematics of the robot with high consistency to the actual robot's motion.

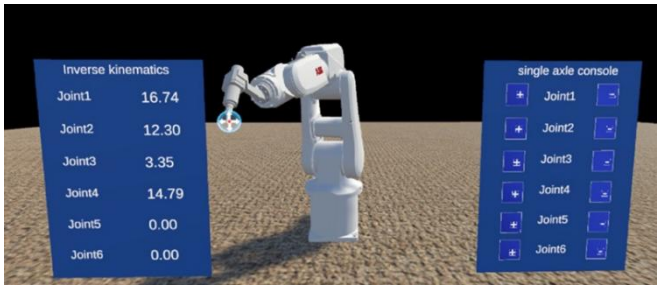

Figure 7. IK target point

6. Conclusion

The virtual simulation system for industrial human-robot interaction has broad application scenarios. After experimental testing, this platform can achieve simulation of both forward and inverse kinematics of robot motion, with simple operation and high efficiency, which can enhance the sense of presence in the interaction between operators and robots. It has practical application value in the training of industrial robot operators.

References

[1] International Federation of Robotics, "World Robotics Report 2022: Industrial Robots," IFR, 2022.

[2] Boston Consulting Group, "Global Adoption and Regional Trends of Industrial Robots," BCG Report, 2023.

[3] European Robotics Forum, "Strategic Development of the Industrial Robotics Sector: Global Outlook," ERF Annual Report, 2024.

[4] Siemens Digital Industries Software, "Transforming Manufacturing with Robotics Simulation," Siemens Technical Papers, 2023.

[5] M. Q. Tram, J. M. Cloud, and W. J. Beksi, "Integrating Virtual Reality for Human-Robot Collaboration," arXiv preprint arXiv:2305.15657, 2023.

[6] F. P. Audonnet, A. Hamilton, and G. Aragon-Camarasa, "Comparing Simulation Software for Industrial Robotics," arXiv preprint arXiv:2204.06433, 2022.

[7] E. Yoshida, "Human-Machine Interaction in Robot Simulations," Advanced Robotics, 2019.

[8] RoboDK Inc., "RoboDK: Advanced Simulation and Offline Programming for Robotics," RoboDK Documentation, 2025.

[9] NVIDIA Corporation, "NVIDIA Isaac Sim on Omniverse: Revolutionizing Industrial Simulations," NVIDIA Developer Blog, 2021.

[10] A. Goswami and P. Vadakkepat, "Kinematic Control and Motion Planning for Robotics," Springer Handbook of Robotics, 2016.

[11] P. Corke, "Robotics, Vision and Control: Fundamental Algorithms," Springer Robotics Series, 2021.

[12] B. Siciliano and O. Khatib, "Kinematics, Dynamics, and Motion Planning in Robotic Manipulators," Handbook of Robotics, 2016.

[13] B. Kenwright, "Inverse Kinematics — Cyclic Coordinate Descent (CCD)," Journal of Graphics Tools, vol. 16, no. 4, pp. 37-41, 2012.

[14] T. Tr ầ n, T. Tr ọ ng, and D. et al., "The CCD-Algebraic Algorithm to Solve the Inverse Kinematics of 6-DOF Redundant Manipulators," Journal of Computer Science, 2022.