Grab application of remote operation service robot

Xia Wang¹,², Tong Zhao³ and Ding Wang¹,²,⁴

¹College of Electronic Engineering, Heilongjiang University, Harbin, China
²Key Laboratory of Information Fusion Estimation and Detection, Heilongjiang Province, Harbin, China
³China Nuclear Engineering Consulting Co., Ltd., China
⁴E-mail: 782939172@qq.com

Abstract. The remote operation service robot can realize remote wireless control through the handle to help people with inconvenient movement to grasp objects. In this paper, based on the robot, the kinematic analysis is carried out to obtain the coordinate transformation of the mechanical fingers relative to the ground, and the dynamic analysis is carried out to obtain the driving torque required by each motor. The inverse kinematics of the manipulator is solved by MATLAB to obtain the joint parameters of the manipulator. Raspberry PI is used as the main control board, the handle sends out instructions, Raspberry PI judges instructions, and communicates with C8051F040 development board located in the manipulator through CAN bus. The mechanical finger realizes open-loop control through micro stepping motor, and finally realizes the remote operation of the service robot to grab objects.

1. Introduction

Remote operation service robot, that is, the service robot controlled by the handle, can combine communication, automation, intelligence and other activities. It involves the kinematics, dynamics and remote control of the robot. For the inverse kinematics solution, Shiqi Lian [1] solves the inverse kinematics of the high degree of freedom manipulator by referring to the inverse Jacobian method [2] and the pseudo inverse method [3], and proposes a method to search the inverse kinematics solution, which reduces the number of iterations of the transposed IK method and improves the solution efficiency. For dynamics, Kvetoslav Belda uses the dynamic mathematical model of the manipulator to carry out predictive control of the manipulator [4], and simulates the manipulator composed of drive, joint and arm through dynamic analysis. For remote control, Rodica Mihaela [5] proposed a remote control method of mobile robot based on XBEE wireless communication, which realized the two-way communication between mobile robot and remote control equipment. Weixuan Sun [6] uses wireless network to interact with the mobile robot. The robot is equipped with a camera, which can synchronize the scene to the terminal remotely, return the scene video in real time, and the handle can accurately control the movement of the robot. In this paper, a remote control robot based on CAN bus is realized.

2. Kinematics analysis

2.1. Direct kinematics analysis

The robot mainly consists of four parts. Firstly, the coordinate system is established for each part, then the coordinate transformation of each part is considered, and the kinematics analysis is carried out according to the D-H parameter. The coordinate system definition is shown in Figure 1.
Fig 1. Service robot coordinate system definition map.

\( T_s \) is the coordinate system of the ground; \( T_r \) is the coordinate system of the mobile platform; \( T_0 \) is the coordinate system of the fixed base of the mechanical arm; \( T_1 \sim T_6 \) is the coordinate system of the first link to the sixth link, that is, the coordinate system of the shoulder joint, the upper arm, the elbow joint, the lower arm, the wrist joint, and the manipulator; \( T_7 \) is the coordinate system of the mechanical finger. The length, width and height of the mobile platform respectively are \( a_length, a_width \) and \( a_height \); The height of the base of the manipulator is \( h \); \( x_r \) and \( y_r \) are the displacement components of the platform in the \( x \) and \( y \) directions of the geodetic coordinate system. \( \phi \) is the rotation angle of the mobile platform. A mobile platform can be obtained with respect to the earth and the top of the pedestal relative to the mobile platform transformation matrix:

\[
s_{T_r} = \begin{bmatrix} \cos \phi & -\sin \phi & 0 & x_r \\ \sin \phi & \cos \phi & 0 & y_r \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \text{top}T_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a_height + h \\ 0 & 0 & 0 & 1 \end{bmatrix}
\]  

In order to establish a coordinate system, the table can be filled out to determine the D-H model. The Z axis is established as follows: If the joint is a rotating joint, the Z axis determines the direction according to the right-hand rule, and the angle \( \theta \) is the rotation angle. If the joint is a sliding joint, the Z axis is the direction of the linear motion and \( d \) is the distance between two adjacent male vertical lines. Measure the actual size of the arm, the D-H parameters can be obtained as shown in Table 1:

| joint | \( \theta \) | \( d \) (mm) | \( a \) | \( \alpha \) |
|-------|------------|-------------|--------|--------|
| 0-1   | \( \theta_1 \) | d1=105     | 0      | 90°    |
| 1-2   | \( \theta_2 \) | d2=105     | 0      | -90°   |
| 2-3   | \( \theta_3 \) | d3=50      | 0      | -90°   |
| 3-4   | \( \theta_4 \) | d4=105     | 0      | 90°    |
| 4-5   | \( \theta_5 \) | d5=50      | 0      | -90°   |
| 5-6   | \( \theta_6 \) | 0          | 0      | 0°     |

Direct kinematics is the length of all the links and the angle of each joint of the known robot, calculating the position and attitude of the end effector. According to Table 1, the kinematic equation for the service robot can be derived as follows:
\[ sT_7 = sT_r^\top 0T_0 0T_1 sT_2 3T_3 4T_4 5T_5 6T_7 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \] (2)

2.2. Inverse kinematics analysis

Inverse kinematics is to determine the angle transformation of each joint by the desired position and attitude of the manipulator arm. The direct kinematics equation can be represented by Equation (2). The right side of the equation is a known quantity, but the left is unknown, and their size depends on the value \( \theta_i \). To solve the equation, first multiply the left matrix by an inverse matrix \( T_r^\top -1 \), then multiply the left matrix by \( T_0^{-1} \) and \( T_1^{-1} \), to get the equation:

\[ 0T_1^{-1} 0T_0^{-1} sT_r^{-1} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \] (3)

Think of \( sT_r \) as \( \begin{bmatrix} A \\ B \end{bmatrix} \) matrix, and find each inverse matrix to get:

\[ 0T_1^{-1} 0T_0^{-1} sT_r^{-1} = \begin{bmatrix} \alpha_1 & \alpha_2 & 0 & \omega_1 \\ \alpha_3 & \alpha_4 & 0 & \omega_2 \\ 0 & 0 & 1 & -a_{\text{heighth}} - h \\ 0 & 0 & 0 & 1 \end{bmatrix} \] (4)

Where:

\[ \alpha_1 = \cos \theta_1 \cos \phi - \sin \theta_1 \sin \phi \]
\[ \alpha_2 = \cos \theta_1 \sin \phi + \sin \theta_1 \cos \phi \]
\[ \alpha_3 = -\sin \theta_1 \cos \phi - \cos \theta_1 \sin \phi \]
\[ \alpha_4 = -\sin \theta_1 \sin \phi + \cos \theta_1 \cos \phi \]
\[ \omega_1 = -x_r \cos \theta_1 \cos \phi - y_r \cos \theta_1 \sin \phi + x_r \sin \theta_1 \sin \phi - y_r \sin \theta_1 \cos \phi \]
\[ \omega_2 = x_r \sin \theta_1 \cos \phi + y_r \sin \theta_1 \sin \phi + x_r \cos \theta_1 \sin \phi - y_r \cos \theta_1 \cos \phi \]

After the expansion, the values of the elements on the left and the right are equal, and we can get:

\[ -x_r \sin \theta_1 \cos \phi + y_r \cos \theta_1 \sin \phi = a_{\text{heighth}} + h \]

(5)

\[ \theta_1 \] can be solved:

\[ \theta_1 = \arctan \left( \frac{x_r, y_r}{a_{\text{heighth}} + h} \right) - \arctan (a_{\text{heighth}} + h, \pm \sqrt{p_x^2 + p_y^2 - a_{\text{heighth}} + h^2}) \]

In the same way, the other five joint variable angles can be obtained.

3. Establishment of dynamic equation

Energy balance is a characteristic of Lagrange's equation and is suitable for multi-link movements under mutual constraints. To get it, we need the kinetic and potential energy of the entire system.

3.1. Calculation of total kinetic energy and total potential energy of mobile chassis

![Figure 2. Moving chassis coordinate system.](image-url)
As shown in Figure 2, the moving chassis establishes a coordinate system: $P_1$: the ground coordinate system; $P_2$: the center of gravity of the mobile chassis, and the distance between $P_0$ and $P_c$ is $d$, and the distance between the drive wheel and the x-axis is $b$.

In the chassis, the directions of the two drive wheels on the same side are substantially the same, so they are combined into one. The moving chassis can be described in five generalized coordinates, three of which are variables describing the position and orientation of the moving chassis, and two are angular variables of rotation of the drive wheel. Therefore, the Lagrangian coordinates are: $q = (x_c, y_c, \varphi, \theta_r, \theta_l)$, where $(x_c, y_c)$ is the position of the center of gravity of the moving chassis, and $\varphi$ is the heading angle of the mobile chassis, $\theta_r$ and $\theta_l$ indicate the angular position rotated by the right and left drive wheels. The drive wheel rolls (no slippage occurs) and three constraints are met. First, the velocity of the moving chassis $P_0$ must match the direction of the axis of symmetry, then: $\dot{x}_c \cos \varphi - x_c \sin \varphi - \dot{\varphi} b = 0$. In addition, if the drive wheel does not slip, there are:

$$\dot{x}_c \cos \varphi + \dot{y}_c \sin \varphi + \dot{\varphi} b = r \dot{\theta}_l, \quad \dot{x}_c \cos \varphi + \dot{y}_c \sin \varphi - \dot{\varphi} b = r \dot{\theta}_r (6)$$

This from it is possible to derive the kinetic equation and move the kinetic energy expression of the chassis body: $K_b = \frac{1}{2} m_r (\dot{x}^2 + \dot{y}^2) + \frac{1}{2} I_{rz} \dot{\varphi}^2$

Where, $I_{rz}$ is the moment of inertia of the car body rotating around the Z axis.

The kinetic energy of the two drive wheels in the mobile chassis is:

$$K_l = \frac{1}{2} m_w (\dot{x} - b \dot{\varphi} \cos \varphi + d \dot{\varphi} \sin \varphi)^2 + \frac{1}{2} m_w (\dot{y} - b \dot{\varphi} \sin \varphi - d \dot{\varphi} \cos \varphi)^2 + \frac{1}{2} I_{wx} \dot{\phi}^2 + \frac{1}{2} I_{wy} \dot{\theta}_r^2 + \frac{1}{2} I_{wy} \dot{\theta}_l^2 (7)$$

$$K_r = \frac{1}{2} m_w (\dot{x} + b \dot{\varphi} \cos \varphi + d \dot{\varphi} \sin \varphi)^2 + \frac{1}{2} m_w (\dot{y} + b \dot{\varphi} \sin \varphi - d \dot{\varphi} \cos \varphi)^2 + \frac{1}{2} I_{wx} \dot{\phi}^2 + \frac{1}{2} I_{wy} \dot{\theta}_r^2 + \frac{1}{2} I_{wy} \dot{\theta}_l^2 (8)$$

Where, $K_l$ and $K_r$ represent the kinetic energy of the left and right drive wheels.

The potential energy of the car body in the mobile chassis is:$P_b = m_b g h_b$

Where, $h_b$ is the distance from the center of gravity of the moving chassis to the horizontal plane of the coordinate system.

### 3.2. Calculation of total kinetic energy and total potential energy of the robot arm

Use $r_i$ to indicate the point of the i-link coordinate system of the robot arm. The position of this point in the base coordinate system of the robot can be obtained by transforming the matrix, that is:

$$P_i = T_i r_i = \mathbf{0} r_i \quad \text{(9)}$$

The speed at this point is a function of the joint speed $q_1, q_2, ... q_l$ of all the robot arms. Therefore, all joint variables are derived according to Equation (9), that is, the speed of point $r_i$ is obtained, and the kinetic energy expression of the mass unit can be obtained:

$$dK_i = \frac{1}{2} \text{Trace} [\{\sum_{j=1}^l (U_{ip} \frac{d q_j}{d t}) \frac{d q_j}{d t} \} \{\sum_{j=1}^l (U_{ir} \frac{d q_j}{d t}) \frac{d q_j}{d t} \}] \quad \text{(10)}$$

where, $p$ and $r$: the joint numbers, because the speed of point $i$ on the link is affected by other links, the influence can be added to the point on the link $i$. For the integral operation of the upper mass unit, the total kinetic energy expression of the link $i$ can be obtained:

$$K_i = \frac{1}{2} \text{Trace} [\sum_{p=1}^l \sum_{j=1}^l (U_{ip} \int r_i r_i^T dm_i) U_{ir}^T q_p q_r] \quad \text{(11)}$$

Can get matrix expression of pseudo inertia:

$$I_i = \begin{bmatrix}
\theta_1 & I_{lxy} & I_{lxx} & m_i \mathbf{x}_i \\
I_{lxy} & \theta_2 & I_{lyz} & m_i \mathbf{y}_i \\
I_{lxx} & I_{lyz} & \theta_3 & m_i \mathbf{z}_i \\
m_i \mathbf{x}_i & m_i \mathbf{y}_i & m_i \mathbf{z}_i & m_i
\end{bmatrix} \quad \text{(12)}$$

Where: $\theta_1 = \frac{1}{2} (-I_{lxx} + I_{lyy} + I_{lzz}), \quad \theta_2 = \frac{1}{2} (I_{lxx} - I_{lyy} + I_{lzz}), \quad \theta_3 = \frac{1}{2} (I_{lxx} + I_{lyy} - I_{lzz})$

It can be known from the model that (12) is substituted into (11) to get the kinetic energy expression of the manipulator link, so the total kinetic energy of the robot is:
\[ K_L = \frac{1}{2} \sum_{i=1}^{7} Trace [\sum_{p=1}^{7} \sum_{r=1}^{7} U_{ip} l_i U^T_{ir} q_p q_r] \]  \hfill (13)

The total potential energy of the robot arm is: \[ P_L = \sum_{i=1}^{7} P_i = -\sum_{i=1}^{7} m_i g^T T_r i_i \]

Where, \( g^T \) is the gravity acceleration, \( r_i \) is the position of the center of mass of the link in the corresponding link frame.

In summary, the Lagrangian function is: \( L = K - P = K_b + K_r + K_l - P_b - P_l \)

The Lagrangian function can be used to describe the dynamic equations of the system:

\[ T_i = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \]  \hfill (14)

\( T_i \) is the generalized driving torque acting on the \( i \)-th joint. As in Equation (14), we can first find the \( \frac{\partial L}{\partial q_i} \) term and then the \( \frac{\partial L}{\partial \dot{q}_i} \) term in the dynamic equation. In summary, the dynamic equation of the service robot is:

\[ T_i = \sum_{j=1}^{7} \sum_{k=1}^{7} \frac{\partial T_r}{\partial q_j} \frac{\partial T_r}{\partial q_k} q_j q_k + \sum_{j=1}^{7} \sum_{k=1}^{7} \sum_{i=1}^{7} \frac{\partial T_r}{\partial q_i} l_i \frac{\partial T_r}{\partial q_j} q_j q_k - \frac{1}{2} \sum_{j=1}^{7} \sum_{k=1}^{7} \sum_{i=1}^{7} \frac{\partial T_r}{\partial q_i} l_i \frac{\partial T_r}{\partial q_j} q_j q_k + \sum_{j=1}^{7} \sum_{k=1}^{7} \sum_{i=1}^{7} m_i g^T T_r i_i \]

4. Joint space trajectory planning of object

This section studies the use of controlled parameters to plan the motion of the robot in the joint space. The object is now placed on the plane of the coordinates \((1, 1)\) of the world coordinate system. The equation of kinematics can be obtained by using MATLAB:

\[ sT_7 = \begin{bmatrix} -0.2289 & 0.5477 & 0.8046 & 138.68 \\ -0.2562 & -0.8312 & 0.4929 & -8.897 \\ 0.9391 & -0.0933 & 0.3307 & 495.14 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]  \hfill (15)

From Chapter 2, \( \cos \theta_1 = 0.574 \) can be obtained, and \( \theta_1 = 55.17^\circ \) can be known. Similarly, \( \theta_2 = 8.17^\circ, \theta_3 = 35.44^\circ, \theta_4 = 7.15^\circ, \theta_5 = 28.56^\circ, \theta_6 = 10.77^\circ \) can be obtained.

Assuming that the first joint in the system moves from the initial angle to the terminal angle within 5 seconds, the angle of the joint at the 1st, 2nd, 3rd, and 4th seconds is calculated using the cubic polynomial. The initial and end conditions are known to be:

\[ \theta_{(t=0)} = 55^\circ, \dot{\theta}_{(t=0)} = 0, \ddot{\theta}_{(t=5)} = 12^\circ, \dot{\theta}_{(t=5)} = 0 \]  \hfill (16)

Four unknowns in the following cubic polynomial equations can be solved:

\[ \theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 \]  \hfill (17)

Get the first order for (17): \( \dot{\theta}(t) = c_1 + 2c_2 t + 3c_3 t^2 \)

Combine the above three formulas to get: \( c_0 = 55, c_1 = 0, c_2 = -5.16, c_3 = 0.688 \). This gives a polynomial equation for position, velocity and acceleration:

\[ \theta(t) = 55 - 5.16t^2 + 0.688t^3, \dot{\theta}(t) = -10.32t + 2.064t^2, \ddot{\theta}(t) = -10.32 + 4.128t \]  \hfill (18)

Substitute time to get: \( \theta(1) = 50.53^\circ, \dot{\theta}(2) = 34.86^\circ, \theta(3) = 27.13^\circ, \theta(4) = 16.47^\circ \)

Assume that the second joint in the system moves from the initial angle to the terminal angle within 5 seconds. Similarly, the positional polynomial equations for the second to sixth joints are as follows:

\[ \theta(t) = 8 - 0.19t^2 + 0.026t^3, \theta(t) = 35 - 2.88t^2 + 0.38t^3, \theta(t) = 7 - 0.36t^2 + 0.048, \theta(t) = 29 - 2.52t^2 + 0.336t^3, \theta(t) = 11 - 0.08t^2 + 0.01t^3 \]  \hfill (19)

5. Overall control scheme design and implementation

The overall mechanical structure is divided into four basic parts: mobile chassis unit, robot arm action unit, main controller unit and handle operation unit. The power supply unit supplies power for each part, which is added as an independent part to form a complete control system. The main controller unit is Raspberry PI, which controls four motors of the mobile chassis through four PWM signals, and
communicates with C8051F040 development board placed in the manipulator through CAN bus. The manipulator realizes open-loop control by micro stepping motor. In the handle operation unit, the wireless receiver is connected to the Raspberry PI to realize serial communication between the handle and the Raspberry PI. Block diagram of overall system structure design is shown in Figure 3.

5.1. CAN bus communication debugging
The Raspberry PI implements CAN bus communication through peripheral CAN module control. Run the command: dmesg | grep -i "(can|spi)", you will see the message of Figure 4, indicating that the peripheral CAN module is successfully initialized:

![Figure 4. CAN module initialization.](image)

After the CAN module is successfully initialized, the function of transmitting data is further debugged. Steps are as follows: (1) Open socket: s = socket (PF_CAN, SOCK_RAW, CAN_RAW); If it fails, return -1; (2) Specify device can0: strcpy(ifr.ifr_name, "can0"); ret = ioctl(s, SIOCGIFINDEX, &ifr); (3) Bind the socket to the CAN interface: addr.can_family = AF_CAN; addr.can_ifindex = ifr.ifr_ifindex; ret = bind(s, (struct sockaddr *)&addr, sizeof(addr)); (4) Set rules, only send packets: setsockopt(s, SOL_CAN_RAW, CAN_RAW_FILTER, NULL, 0); (5) Set the data to be sent: struct can_frame frame; frame.can_id = 0x123; frame.can_dlc = 8; frame.data[0] = 1 to frame.data[7] = 8; (6) Send data nbytes = write(s, &frame, sizeof(frame)); Write the data sent to the socket through the write () function; If it fails, return -1; (7) Close sockets and CAN devices close (s): system("sudo ifconfig can0 down"). Through the above steps, as shown in the Figure 5, CAN communication between the Raspberry Pi and the C8051F040 development board is realized.

5.2. The overall realization of the remote operation service robot
Press the "start" key of the handle to initialize, and send instructions to the Raspberry Pi through the handle. If it is a movement instruction, Raspberry Pi will output four PWM programs to the motor of the mobile site; If it is a grab command, calculate the motion parameters of each joint of the robot arm and send a command to the C8051F040 development board through the CAN bus. The robot arm can

![Figure 3. Block diagram of overall system structure design.](image)
grasp the object according to the instruction. The remote control service robot grabs objects as shown in Figure 6.

![Raspberry Pi CAN communication](image1)

**Figure 5.** Raspberry Pi CAN communication.

![Remote operation robot grabs objects](image2)

**Figure 6.** Remote operation robot grabs objects.

## 6. Conclusions

In this paper, the overall design of the remote control service robot includes four control units, the main controller is Raspberry PI, which controls the mobile chassis, and communicates with C8051F040 development board through CAN bus. The kinematics model of the robot is built, and the mobile chassis and body are analyzed one by one; the dynamics analysis is carried out, and the torque required by each motor is obtained; the inverse kinematics of the manipulator is solved to obtain the joint parameters of the manipulator, which lays a theoretical foundation for the actual debugging. This system adds CAN module to Raspberry PI. Can bus technology is applied to the development of remote control service robot, which can greatly reduce the number of lines needed in the robot arm, and the robot arm can be expanded to increase the freedom of the robot arm to meet different needs.

## Acknowledgments

This work was supported by the Open Foundation of China under Grant DZGC201612 and the Key Laboratory of Electronic Engineering.

## References

[1] Lian Shiqi 2017 Dadu: Accelerating Inverse Kinematics for high-DOF robots[C]. 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC) 1-6

[2] O Hock and J Sedo 2018 Inverse kinematics using transposition method for robotic arm [C]. 2018 ELEKTRO 1-5

[3] W A Wolovich and H Elliott 1984 A computational technique for inverse kinematics [C]. IEEE Conference on Decision and Control 1359-1363

[4] K Belda and O Rovny 2017 Predictive control of 5 DOF robot arm of autonomous mobile robotic system motion control employing mathematical model of the robot arm dynamics[C]. 2017 21st International Conference on Process Control (PC) 339-344

[5] R M Teodorescu, I B Cioc, B A Vochin and A I Litav 2016 LabVIEW application used for remote control of a mobile robot with xbee communication[C]. 2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI) 1-4

[6] Sun W, Liu C and Zhu J 2018 A Remote Controlled Mobile Robot Based on Wireless Transmission[C]. 2018 2nd IEEE Advanced Management Information, Communicates, Electronic and Automation Control Conference (IMCEC) 2173-2176