Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 4, No. 3, 2023

A Gravity Compensation Algorithm of Robot Manipulator Control based on the Trigonometric Function

Yan Wang, Xue Chen

Guizhou Light Industry Technical College, Guiyang 550025, China

Abstract: The conventional gravity compensation algorithm requires precise dynamic parameters and a complex matrix transformation operation, which is difficult in applications to real-time control. In this paper, a simple and practical gravity compensation algorithm is proposed based on the space geometry characteristics of a mechanical arm and the principle of torque balance. This algorithm does not require a complex calculation of space coordinate transformation and does not require obtaining all accurate dynamic models and parameters. It only requires estimating the maximum gravity moment of the mechanical arm and simply calculating the trigonometric function. Thus, this algorithm can be extended to a non-parallel shaft mechanical arm, which is suitable for N joints in space. To verify the control effect after gravity compensation, the most easily comprehensible proportional-derivative controller combined with gravity compensation is used to control two-joint and three-joint mechanical arms for simulation. With the gravity compensation and non-compensation of the mechanical arm and with a comparison with other compensation methods, such as the fixed gravity compensation algorithm, the results show that the gravity compensation algorithm can achieve better trajectory tracking control, higher steady-state precision, an effectively reduced work burden of the controller and improved system stability.

Keywords: Robot Manipulator; Trajectory Tracking Control; Gravity Compensation; Trigonometric Function.

1. Introduction

Service robots have a human-like appearance and structure that mainly relies on a mechanical arm to complete daily operations. These robots require trajectory tracking control to accomplish these actions and tasks. Since most of the driving force of the mechanical arm in space motion works against its own gravity, to reduce gravity disturbance and improve steady-state precision, it is necessary to compensate the gravity of the mechanical arm in the control process.

At present, some progress has been made in the study of gravity compensation, mainly including a mechanical-structure method and a control-analysis method [1]. In the mechanical-structure method, some compensation devices are added to the mechanical structure in Ref. [2-5]; for example, the gravity was compensated by adding springs to each joint. In the control-analysis method, a feedforward or feedback link is added to the control strategy to compensate its own gravity. In this paper, we will study the control-analysis method. In Ref. [6, 7], a fixed position control algorithm of a mechanical arm with fixed gravity compensation was proposed, which is relatively simple and applicable to fixed position control in operating space. However, the error of the trajectory tracking control of this method is relatively large. In Ref. [8, 9], the least squares method and back-propagation neural network were applied to identify dynamic parameters such as the mass and mass centre of a mechanical arm connecting rod and to estimate the gravity term. However, this method is time-consuming and complex and is not easy to control in real time. To realise the trajectory tracking control of a mechanical arm, a gravity compensation algorithm was applied for real-time and accurate estimation of the gravity moment in Ref. [10, 11]. However, in real-life applications, it is difficult to guarantee the accuracy of gravity compensation because various accurate dynamic parameters are difficult to obtain. In Ref. [12, 13], a system dynamics model using Lagrange's equation was built to calculate the torque generated by each part of the centre of gravity in joint space, using the torque motor to obtain the corresponding torque for gravity compensation. One disadvantage of this method is that the calculation of the gravity matrix with multiple degrees of freedom is relatively complex and the exact parameters of the Lagrange equation are not easily obtained. Based on the traditional fixed gravity compensation, the gravity compensation function of each joint was derived in Ref. [14]. A relatively accurate online gravity compensation proportional-derivative (PD) control strategy was proposed for the right arm of a de-icing robot. However, this compensation algorithm requires knowing the weight and mass centre as well as other parameters of the mechanical arm in advance and is only applicable to parallel shaft mechanical arms.

It can be seen from the above review that the conventional gravity compensation algorithm requires precise dynamic parameters and a complex matrix transformation operation, which is difficult in applications in real-time control. Although some improved real-time control algorithms are simple to calculate, they are only applicable to parallel shaft mechanical arms [15]. In view of the shortcomings of the above studies, in this paper, we study a seemingly simpler gravity compensation method. This algorithm does not need complicated calculations, nor does it require obtaining all accurate dynamic models and parameters to achieve a better control effect, and it is also suitable for non-parallel shaft mechanical arms.

2. Gravity Compensation Algorithm of a Mechanical Arm

2.1. Analysis of Gravity Compensation of a Mechanical Arm

According to the Lagrange modelling method, the mechanical arm dynamics equation with *n* degrees of freedom can be expressed as follows:

$$\boldsymbol{M}(q)\ddot{q}+\boldsymbol{C}(q,\dot{q})\dot{q}+\boldsymbol{G}(q)=\boldsymbol{\Gamma} \tag{1}$$

where $q(t)$ represents the angular displacement vector of a joint, which is a bounded differentiable vector; $\boldsymbol{M}(q)$ represents the inertial matrix of the mechanical arm; $\boldsymbol{C}(q,\dot{q})$ represents the Coriolis force and the centrifugal force vector; $\boldsymbol{G}(q)$ represents the gravity vector; and $\boldsymbol{\Gamma}$ represents the input torque. Mechanical arms usually work at a low speed. It can be seen from Eq. (1) that the dynamic performance of the robot is affected by gravity. If gravity is regarded as a disturbance, most of the work done by the input torque will be used to overcome the disturbance of the gravity moment. If gravity can be compensated effectively in the control process, the control efficiency will be greatly improved.

2.2. Calculation of the Gravity Moment of a Mechanical Arm

According to the Lagrange equation for robots, the gravity term of the ith joint of the mechanical arm with *n* joints can be expressed as follows:

$$g_i=-\sum_{k=i}^{n} m_k g^T \frac{\partial^0\boldsymbol{T}_k}{\partial q_i}{}^k\boldsymbol{p}_k \tag{2}$$

where m_k is the mass of each connecting rod of the robot arm, g is the gravity acceleration vector in the basic coordinate system, ${}^0\boldsymbol{T}_k$ is the homogeneous transformation matrix of the *k*-coordinate relative to the basic coordinate system and ${}^k\boldsymbol{p}_k$ represents the distance vector from the origin of the *k*-coordinate to the mass centre of the connection rod *k*. Therefore, the gravity matrix is given as follows:

$$\boldsymbol{G}(q)=\left[g_1,g_2,g_3,\dots,g_n\right]^{\mathrm{T}} \tag{3}$$

We can apply Eq. (2) to exactly calculate the gravity term at each moment and compensate for it, that is, to make the gravity compensation term

$$\boldsymbol{\Gamma_0}=\boldsymbol{G}(q) \tag{4}$$

Let the input torque be

$$\boldsymbol{\Gamma}=\boldsymbol{\Gamma_0}+\boldsymbol{\Gamma}' \tag{5}$$

Substituting Eqs. (3) and (4) into Eq. (1), we get

$$\boldsymbol{M}(q)\ddot{q}+\boldsymbol{C}(q,\dot{q})\dot{q}=\boldsymbol{\Gamma}' \tag{6}$$

where $\boldsymbol{\Gamma}'$ is the input torque after gravity compensation. Thus, the input torque can completely compensate the gravity term, which means that it can completely eliminate the disturbance of the gravity moment. However, the prerequisite is to know the dynamic parameters, such as the mass and mass centre, of each connecting rod of the robot arm. It can be seen from Eq. (2) that, to calculate the gravity moment of each joint of the mechanical arm, the two parameter values of mass m_k and mass centre distance (${}^k p_k$) must be known in advance. However, it is difficult to obtain relatively accurate values using conventional measurement methods. Moreover, this method requires a large number of calculations on the rotation transformation matrix, which becomes very large when there are more joints, so it is difficult to apply this method in actual applications.

2.3. Gravity Compensation Algorithm of Parallel Shaft Mechanical Arms Perpendicular to Gravity

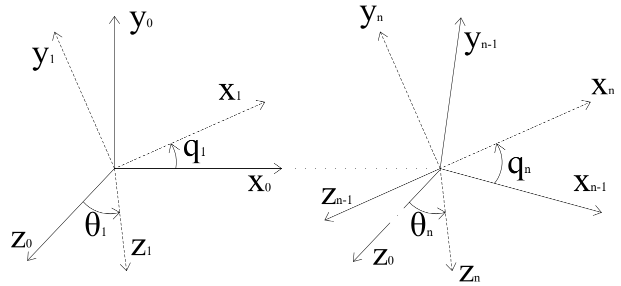


Fig 1. Coordinate of an *N*-joint mechanical arm

Figure 1 shows the coordinate of a mechanical arm with *n* joints as rotating joints, *x* as the connecting rod direction and *z* as the rotation axis direction. In the basic coordinate {0}, the gravity direction is $-y_0$ and the rotation axis direction z_0 is perpendicular to the plane x_0y_0. If not specified, the basic coordinate studied in the following section is the coordinate system shown in Fig. 1. Joint angle q_1 is defined as the angle between x_0 and the connecting rod x_1 in the basic coordinate system and q_{n} is defined as the angle between the connection rods $x_{n\text{-}1}$ and x_n. According to the definition of joint angle and the property of gravity, when the mechanical arm is fully expanded along the direction perpendicular to gravity, in other words, when $q_1=q_2=\cdots\cdots q_{\mathrm{n}}=0$, the maximum gravity moment is received.

If the mechanical arm meets the following two conditions, then the maximum value of the gravity moment at each joint can be derived from Eq. (2):

1) All joints are rotating joints.

2) For all connecting rods, the mass centre of the *i* connecting rod is on the line between the *i* coordinate system and the origin of the *i* +1 coordinate system.

$$\begin{bmatrix}\bar{g}_1\\ \bar{g}_2\\ \bar{g}_3\\ \dots\\ \bar{g}_{n-1}\\ \bar{g}_n\end{bmatrix}=\begin{bmatrix} m_1l_1^c+l_1\sum_{k=2}^{n}m_k\\ m_2l_2^c+l_2\sum_{k=3}^{n}m_k\\ m_3l_3^c+l_3\sum_{k=4}^{n}m_k\\ \dots\\ m_{n-1}l_{n-1}^c+l_{n-1}m_n\\ m_nl_n^c\end{bmatrix}g \tag{7}$$

where $\bar{g}_{\mathrm{n}}$ represents the maximum value of the gravity moment of the *n*th joint, m_{n} is the gravity of the *n*th connecting rod, l_{n} is the length of the nth connecting rod, l_n^c is the distance between the mass centre of the nth connecting rod and the origin of the nth coordinate system and g is the acceleration due to gravity.

Parallel shaft mechanical arm perpendicular to gravity refers to all joint axis angles $\theta_n = k\pi$ (k is an integer); in other words, all joint axes of the mechanical arm are perpendicular to the direction of gravity and parallel to each other. According to the property of torque and the space geometry characteristics of the mechanical arm, we consider the maximum gravity moment and joint angle of each joint as variables. Thus, the online gravity compensation algorithm is designed as follows:

$$\begin{bmatrix} \overline{G}(q_1) \\ \overline{G}(q_2) \\ \overline{G}(q_3) \\ \cdots \\ \overline{G}(q_{n-1}) \\ \overline{G}(q_n) \end{bmatrix} = \begin{bmatrix} (m_1 l_1^c + l_1 \sum_{k=2}^{n} m_k) g \cos(q_1) \\ (m_2 l_2^c + l_2 \sum_{k=3}^{n} m_k) g \cos(q_1 + q_2) \\ (m_3 l_3^c + l_3 \sum_{k=4}^{n} m_k) g \cos(q_1 + q_2 + q_3) \\ \cdots \\ (m_{n-1} l_{n-1}^c + l_{n-1} m_n) g \cos(\sum_{i=1}^{n-1} q_i) \\ m_n l_n^c g \cos(\sum_{i=1}^{n} q_i) \end{bmatrix} \tag{8}$$

where $\overline{G}(q_n)$ represents the torque to be compensated for the nth joint, m_{n} is the gravity of the nth connecting rod, l_{n} is the length of the nth connecting rod, l_n^c is the distance between the mass centre of the nth connecting rod and the origin of the nth coordinate system, g is the acceleration due to gravity and q_{i} is the ith joint angle.

2.4. General Gravity Compensation Algorithm of a Space Manipulator

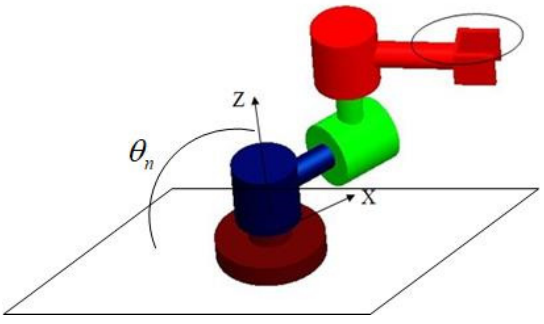


Fig 2. Definition of the joint axis angle

As shown in Fig. 2, the joint axis angle θ_n is defined as the angle between the rotation joint axis z_n and the horizontal plane $z_0 x_0$. Whether and how much the space manipulator requires gravity compensation during the control process are closely related to the joint axis angle. As long as the joint axis angle $\theta_n \neq k\pi/2$ (k is a non-zero integer), the manipulator will be disturbed by the gravity and it will be necessary to compensate the gravity. According to the property of torque and the space geometry characteristics of the mechanical arm, taking the maximum gravity moment, joint axis angle and joint angle of each joint as variables, the online gravity compensation algorithm is designed as follows:

$$\begin{bmatrix} \overline{G}(q_1) \\ \overline{G}(q_2) \\ \overline{G}(q_3) \\ \cdots \\ \overline{G}(q_{n-1}) \\ \overline{G}(q_n) \end{bmatrix} = \begin{bmatrix} (m_1 l_1^c + l_1 \sum_{k=2}^{n} m_k) g \cos(\delta_1 q_1) \\ (m_2 l_2^c + l_2 \sum_{k=3}^{n} m_k) g \delta_2 \cos(\delta_1 q_1 + q_2) \\ (m_3 l_3^c + l_3 \sum_{k=4}^{n} m_k) g \delta_3 \cos(\delta_1 q_1 + \delta_2 q_2 + q_3) \\ \cdots \\ (m_{n-1} l_{n-1}^c + l_{n-1} m_n) g \delta_{n-1} \cos(\sum_{i=1}^{n-2} \delta_i q_i + q_{n-1}) \\ m_n l_n^c g \delta_n \cos(\sum_{i=1}^{n-1} \delta_i q_i + q_n) \end{bmatrix} \tag{9}$$

where $\overline{G}(q_n)$ represents the torque to be compensated for the nth joint, m_{n} is the gravity of the nth connecting rod, l_{n} is the length of the nth connecting rod, l_n^c is the distance between the mass centre of the nth connecting rod and the origin of the nth coordinate system, g is the acceleration due to gravity, q_{i} is the ith joint angle and δ_n is the compensation coefficient.

The general gravity compensation algorithm of a space manipulator [see Eq. (9)] has a compensation coefficient δ_n compared with the gravity compensation algorithm of parallel shaft mechanical arms [see Eq. (8)]. This coefficient is related to the joint axis angle, which is defined as follows:

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \cdots \\ \delta_{n-1} \\ \delta_n \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) \\ \cos(\theta_2) \\ \cos(\theta_3) \\ \cdots\cdots \\ \cos(\theta_{n-1}) \\ \cos(\theta_n) \end{bmatrix} \tag{10}$$

The compensation coefficient is zero when the nth joint axis angle $\theta_n = k\pi/2$ (k is a non-zero integer) because, at this point, the joint axis is parallel to the direction of gravity and gravity has a little influence on the rotation motion of the joint. Therefore, no gravity compensation is required. Most of the previous studies have been conducted on parallel shaft mechanical arms with all joint axis angles of $\theta_n = k\pi$ (k is an integer). In this case, all joint axes are perpendicular to the direction of gravity and the compensation coefficient is 1. The rotation motion of the joint is mostly affected by gravity. When the direction of the joint axis is neither parallel nor perpendicular to the direction of gravity, the compensation coefficient δ_n of each joint is used to indicate the influence of gravity on the rotation motion of the joint. The larger the value is, the larger the disturbance of gravity is, and the corresponding gravity compensation is calculated according to the algorithm. Therefore, the gravity compensation algorithm modified by the compensation coefficient is suitable for non-parallel shaft mechanical arms of any joint axis angle, which extends the application scope of the algorithm in this paper.

3. Control Scheme Design and Stability Analysis

3.1. Control Scheme Design

To verify the feasibility and effectiveness of the compensation algorithm of the mechanical arm trajectory tracking control system, the simplest PD control is adopted for the convenience of analysis. The control process of the dual-joint mechanical arm system is shown as follows.

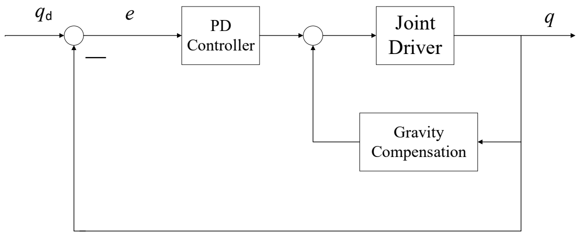


Fig 3. Control flow chart

The expected input of the control system is q_d, the output variable is the real-time joint angle q and the error variable is defined as

$$e = q_d - q \,, \dot{e} = \dot{q}_d - \dot{q} \tag{11}$$

Among these, the PD controller uses input and output deviation for control and its control law is given as

$$\boldsymbol{\Gamma} = \boldsymbol{K}_P \boldsymbol{e} + \boldsymbol{K}_v \dot{\boldsymbol{e}} + \overline{\boldsymbol{G}}(q_d) \tag{12}$$

where K_P and K_v are *n*-dimensional positive definite diagonal constant matrices, and the variables $\bar{G}$ are defined in Eq. (9).

$$\overline{\boldsymbol{G}}(q) = \left[G(q_1), G(q_2), \ldots, G(q_n)\right]^{\mathrm{T}} \tag{13}$$

Using Eqs. (1) and (12), we derive

$$\boldsymbol{M}(q)\ddot{\boldsymbol{q}} + \boldsymbol{C}(q,\dot{q})\dot{\boldsymbol{q}} + \boldsymbol{G}(q) = \boldsymbol{K}_P \boldsymbol{e} + \boldsymbol{K}_v \dot{\boldsymbol{e}} + \overline{\boldsymbol{G}}(q_d) \tag{14}$$

3.2. System Stability Analysis

The closed-loop control equation [see Eq. (14)] and Eq. (11) are rewritten in the state space form as follows:

$$\frac{d}{dt}\begin{bmatrix} \boldsymbol{e} \\ \dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ \boldsymbol{M}(q)^{-1}\left[\boldsymbol{K}_p e + \boldsymbol{K}_v \dot{q} - \boldsymbol{C}(q,\dot{q})\dot{q} + \overline{\boldsymbol{G}}(q_d) - \boldsymbol{G}(q)\right] \end{bmatrix} \tag{15}$$

The Lyapunov function is constructed as follows:

$$V(e,\dot{q}) = \dot{\boldsymbol{q}}^T P(e)\dot{\boldsymbol{q}} + h(e) \tag{16}$$

Where

$$\begin{cases} P(e) = \frac{1}{2}\boldsymbol{M}(q_d - e) \\ h(e) = u(q_d - e) - k_u + \frac{1}{2}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{K}_p \boldsymbol{e} + \boldsymbol{e}^{\mathrm{T}}\overline{\boldsymbol{G}}(q_d) + \frac{1}{2}\overline{\boldsymbol{G}}(q_d)^{\mathrm{T}}\boldsymbol{K}_p^{-1}\overline{\boldsymbol{G}}(q_d) \end{cases} \tag{17}$$

Where $u(q)$ represents the potential energy function of the mechanical arm system.

$$k_u = \min_q \left\{u(q)\right\} \tag{18}$$

Assuming the robot manipulator joints to be all rotary, then

$$u(q) - k_u \geq 0 \tag{19}$$

It can be proved that the Lyapunov function is non-negative definite.

Take the derivative of this function with respect to time:

$$\dot{V}(e,\dot{q}) = \dot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{M}(q)\ddot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^{\mathrm{T}}\dot{\boldsymbol{M}}(q)\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{G}(q) + \boldsymbol{e}^{\mathrm{T}}\boldsymbol{K}_P\dot{\boldsymbol{e}} + \dot{\boldsymbol{e}}^{\mathrm{T}}\overline{\boldsymbol{G}}(q_d) \tag{20}$$

According to the definition of the gravity matrix,

$$\boldsymbol{G}(q) = \frac{\partial}{\partial q}\boldsymbol{u}(q) \tag{21}$$

Using Eq. (15), we can derive

$$\dot{V}(e,\dot{q}) = \dot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{K}_P\boldsymbol{e} - \dot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{K}_v\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^{\mathrm{T}}\overline{\boldsymbol{G}}(q_d) + \boldsymbol{e}^{\mathrm{T}}\boldsymbol{K}_P\dot{\boldsymbol{e}} + \dot{\boldsymbol{e}}^{\mathrm{T}}\overline{\boldsymbol{G}}(q_d) \tag{22}$$

Using the properties of the mechanical arm kinetic equation [16], we obtain

$$\dot{\boldsymbol{q}}^{\mathrm{T}}\left[\frac{1}{2}\dot{\boldsymbol{M}}(q) - \boldsymbol{C}(q,\dot{q})\right]\dot{\boldsymbol{q}} = 0 \tag{23}$$

According to formula (11), we can derive

$$\dot{e} = -\dot{q} \tag{24}$$

Making Eqs. (22), (23) and (24) simultaneously, we can derive

$$\dot{V}(e,\dot{q}) = -\dot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{K}_v\dot{\boldsymbol{q}} \leq 0 \tag{25}$$

We know from Eq. (25) that when $\boldsymbol{K}_v$ is positive definite, $\dot{V}(e,\dot{q})$ is negative semi-definite. According to the Lyapunov stability criterion, the position error e and velocity $\dot{q}(t)$ are bounded under any initial condition. According to La Salle's theorem [15], $[e,\dot{q}] = [0,0]$ is the global asymptotically stable equilibrium point of the system, which proves the stability of the system and meets the basic requirements of the control system.

4. Simulation and Research of Mechanical Arm Gravity Compensation

4.1. Dual-Joint Mechanical Arm Simulation

To verify the effectiveness of the gravity compensation control method, an experimental simulation test was conducted on a dual-joint parallel shaft mechanical arm, shown in Fig. 3. The end motion track of the mechanical arm is designed as a circle, shown in Fig. 4.

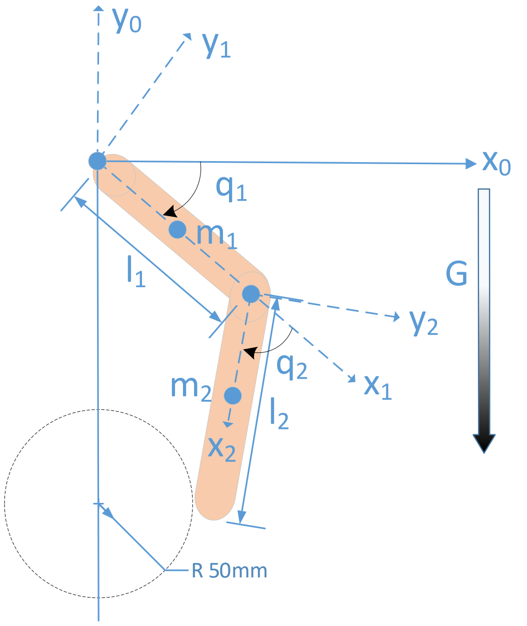


Fig 4. Configuration and motion track of the dual-joint mechanical arm

The ADAMS software was used to model the dual-joint mechanical arm, and MATLAB Simulink is used to design the PD gravity compensation controller. The simulation test scheme is as follows.

1) Setting up the dual-joint parallel shaft mechanical arm model as shown in Fig. 4, the parameters are shown as Table 1.

Table 1. Parameters of the dual-joint mechanical arm

Parameter/ (unit)	l_1,l_2/(mm)	l_1^c, l_1^c/(mm)	m_1,m_2/(kg)
Value	250	125	0.88

2) To make the end motion track of the mechanical arm follow the specified trajectory, it is necessary to calculate the joint variables of the two joints through the trajectory equation and then use these two-joint variables as the input of the control system model. Take q_1 and q_2 as the angular displacement of joint 1 and joint 2, respectively, with l_1 and l_2 being the respective lengths of the two joints. The non-linear equations are

$$\begin{cases} x = l_1 * \cos(q_1) + l_2 * \cos(q_2) \\ y = l_1 * \sin(q_1) + l_2 * \sin(q_2) \end{cases} \tag{26}$$

The time interval t is set as 0–6.28 s with 400 points in the middle, and the joint angles at each time point are solved to obtain the joint space trajectory over the entire time interval.

3) Using the algorithm shown in Eq. (9) and the control process shown in Fig. 2, gravity compensation is carried out using the joint angle from the feedback, and the gravity compensation algorithm of the dual-joint parallel shaft mechanical arm is obtained as follows:

$$\begin{cases} \overline{G}(q_1) = [(m_1 l_1^c + m_2 l_1)g + m_2 l_2^c g]\cos(q_1) \\ \overline{G}(q_2) = m_2 l_2^c g \cos(q_1 + q_2) \end{cases} \tag{27}$$

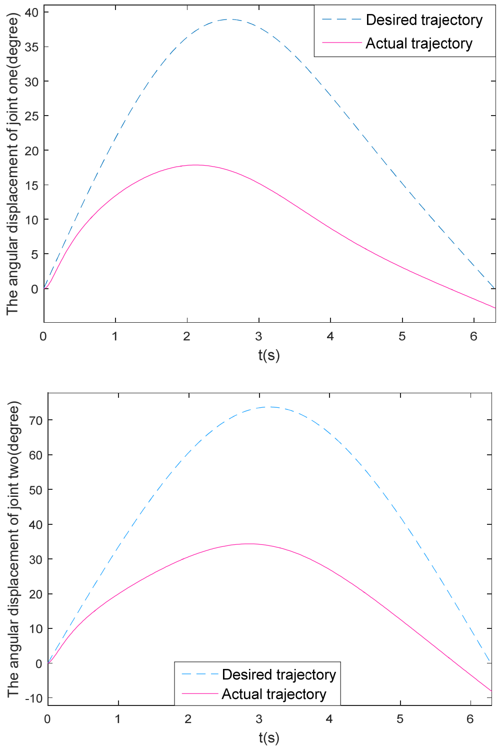


Fig 5. Trajectory tracking curve without weight compensation

The parameters of the PD controller are selected as $\boldsymbol{K}_p = \boldsymbol{K}_v = \begin{bmatrix} 40 & 0 \\ 0 & 20 \end{bmatrix}$. First, we look at the simulation results without gravity compensation.

As can be seen from Fig. 5, the PD controller only uses deviations to carry out the test. If the control does not conduct gravity compensation, the disturbance of gravity will not be able to be eliminated, so the error will be relatively large and it will be difficult to conduct trajectory tracking control. To verify the performance of the gravity compensation algorithm in this paper, the fixed gravity compensation algorithm proposed in [6, 7] was adopted for comparison.

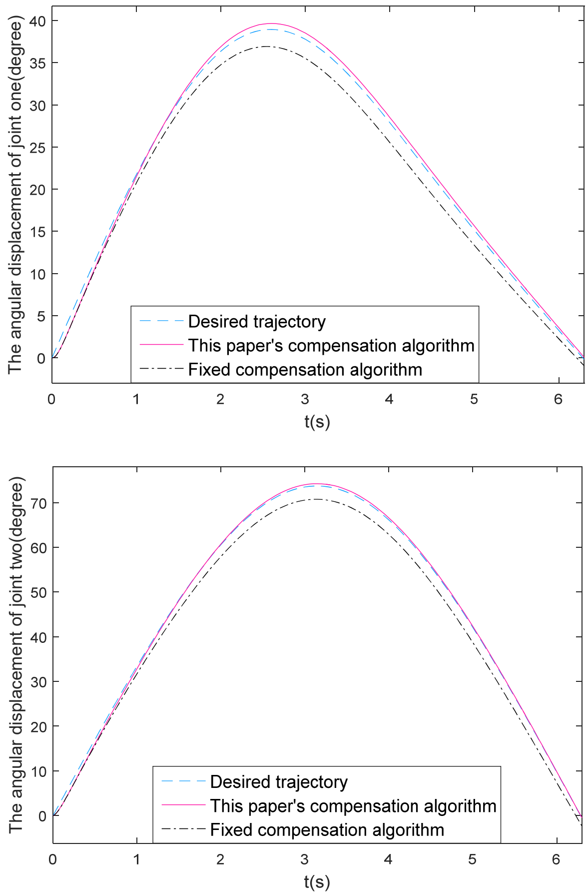


Fig 6. Trajectory tracking curve with weight compensation.

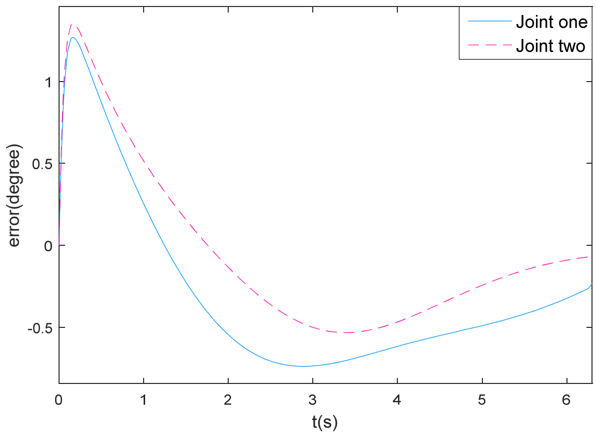


Fig 7. Tracking error curve of the compensation algorithm in this paper

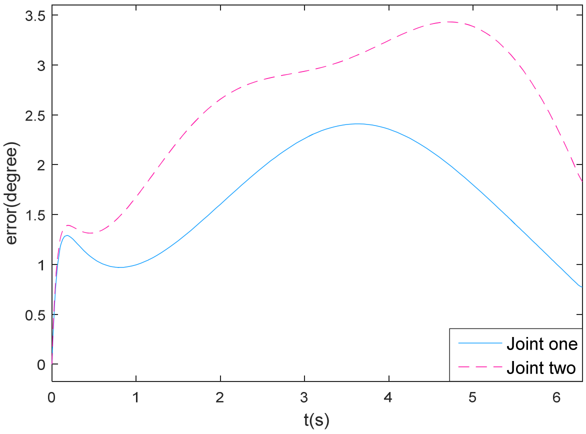


Fig 8. Tracking error curve of a fixed compensation algorithm

It can be seen from Fig. 6 that the trajectory tracking control can be well realized after the adoption of gravity compensation. According to Fig. 7, the gravity compensation algorithm adopted in this paper can cause the tracking error $|e| < 1.5^o$ and steady-state precision to be high. In Fig. 8, the tracking error of the fixed compensation algorithm exceeds 3^o. Therefore, the gravity compensation algorithm in this paper is better in terms of error control.

Next, an input sinusoidal signal is used to test the dynamic tracking performance of the system. It is assumed that the input signal of joints 1 and 2 is

$$\begin{cases} \varphi_1 = 70\sin(t) \\ \varphi_2 = -70\sin(t) \end{cases} \tag{28}$$

The tracking effect of the dual-joint mechanical arm under different sinusoidal signals is as follows.

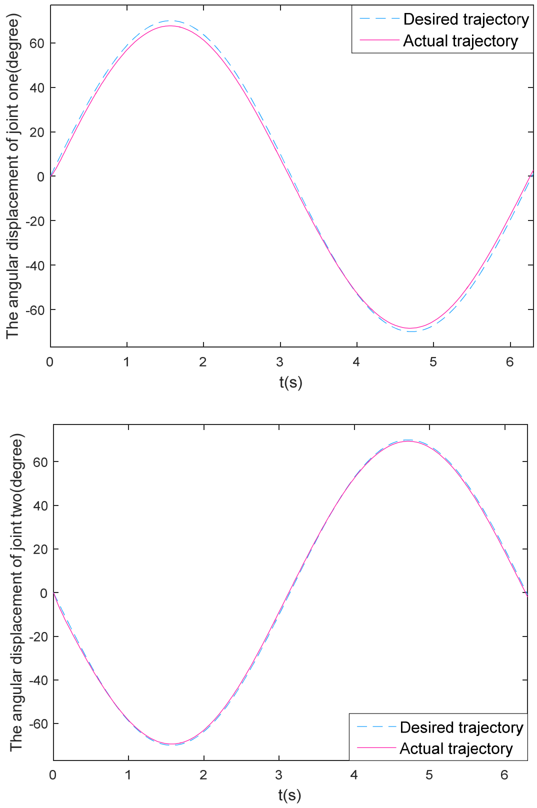


Fig 9. Trajectory tracking curve under a sinusoidal signal

It can be seen from Fig. 9 that the gravity compensation algorithm has a better tracking performance for general sinusoidal waves and the tracking error is $|e| < 2^o$, thus achieving better trajectory tracking control.

4.2. Three-joint Mechanical Arm Simulation

To further verify the effectiveness of the gravity compensation control algorithm, a gravity compensation simulation was performed on the three-joint mechanical arm shown in Fig. 10. The gravity parameters are shown in Table 2.

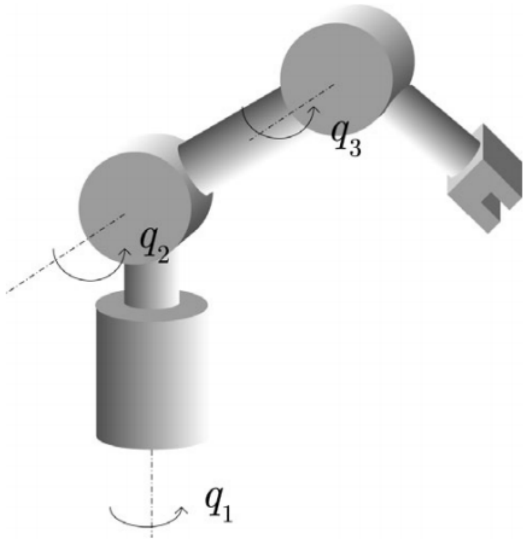


Fig 10. Structure and model of the three-joint mechanical arm

Table 2. Parameters of the three-joint mechanical arm

Parameter	Rod length/(mm)				Mass/(kg)		
	L_1	L_2	L_3		M_1	M_1	M_3
Value	100	125	125		1	0.88	0.88

Since the terminal attitude is not considered, only the first three joints controlling the terminal position are used for the trajectory tracking control experiment. The trajectory motion curves of the three-joint angles are shown in the following:

$$\begin{cases} \varphi_1 = \pi/3\sin(0.5t+\pi) \\ \varphi_2 = \pi/3\sin(0.5t) \\ \varphi_3 = \pi/3\sin(0.5t) - 0.3556\pi \end{cases} \tag{29}$$

In addition, the gravity compensation algorithm based on the trigonometric function shown in Eq. (12) is utilised, and the parameters are set as follows:

$$\boldsymbol{K}_p = \begin{bmatrix} 1000 & & \\ & 500 & \\ & & 500 \end{bmatrix} \qquad \boldsymbol{K}_v = \begin{bmatrix} 10 & & \\ & 100 & \\ & & 100 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 & & \\ & 94\cos(q_2) & \\ & & 18\cos(q_2+q_3) \end{bmatrix}$$

According to Eq. (10), since the rotation axis of the first joint is parallel to the direction of gravity, its gravity compensation coefficients are set as $\delta_1 = 0 \Rightarrow G_1 = 0$; thus, no gravity compensation is required. The experimental results are shown as follows. Figures 11 and 12 represent the angular displacement and tracking error curves of each joint, respectively.

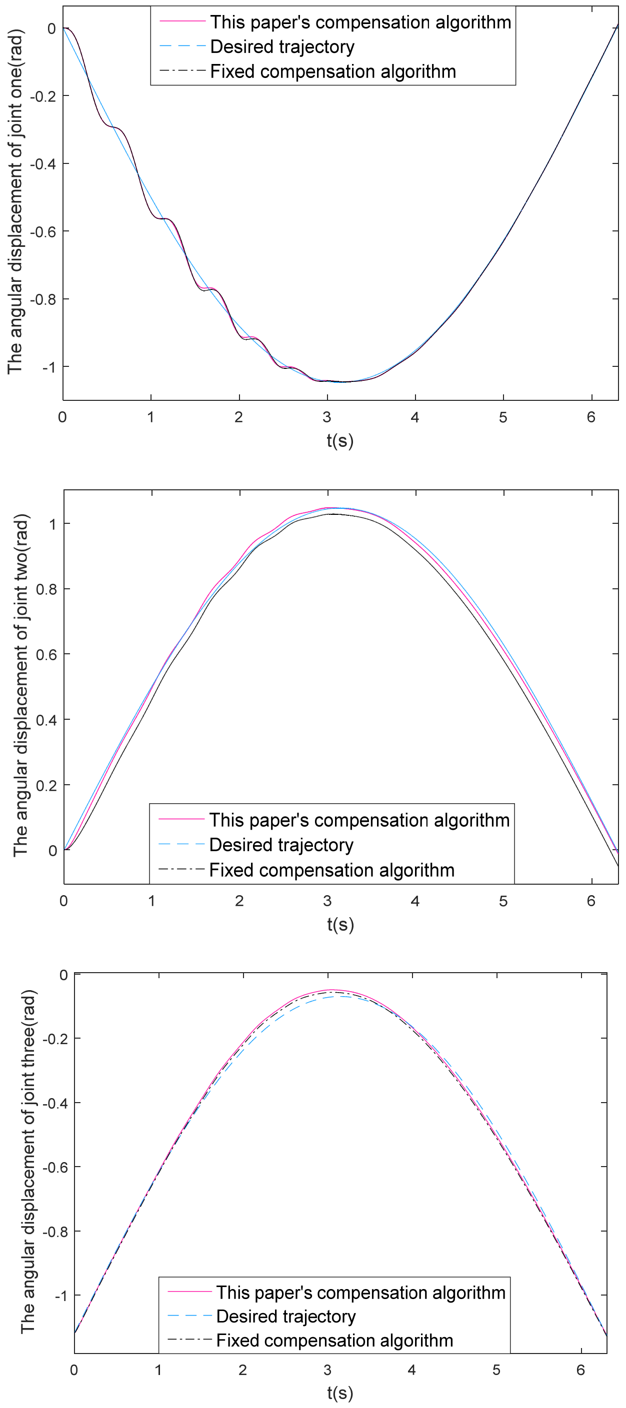
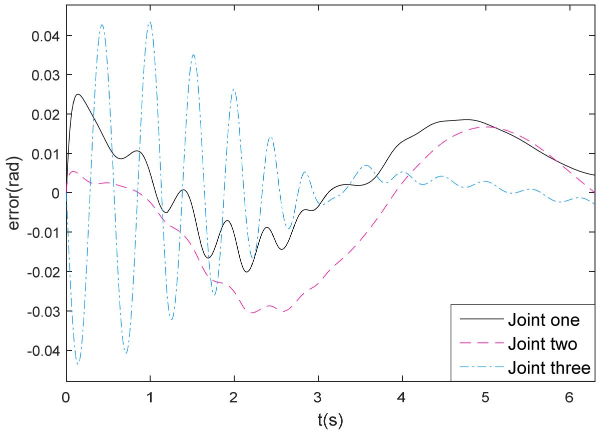


Fig 11. Trajectory tracking curve of the three-joint mechanical arm

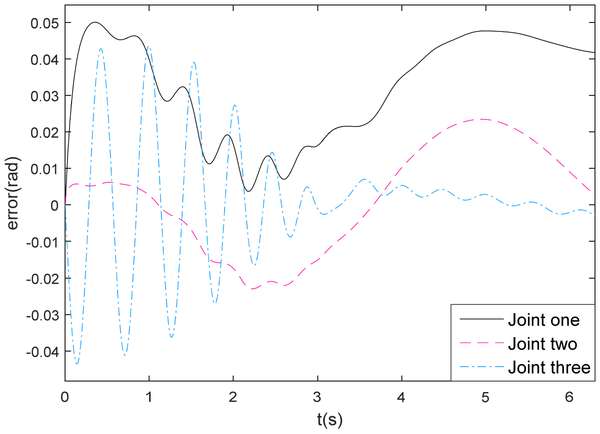
As can be seen from Figs. 11 and 12, the error of the gravity compensation algorithm in this paper is no more than 0.045 rad, whereas the error of the fixed compensation algorithm is more than 0.05 rad. Therefore, the compensation algorithm in this paper has a better compensation effect and can realise better trajectory tracking control.

5. Conclusion

In this paper, a gravity compensation method with an easy algorithm and implementation was proposed. Furthermore, the concept of a ‘compensation coefficient’ was introduced and applied to gravity compensation based on the space geometry characteristics of a mechanical arm and the principle of torque balance.



(a) The algorithm in this paper



(b) Fixed compensation algorithm

Fig 12. Tracking error curve of the three-joint mechanical arm.

This algorithm does not require calculating the gravity compensation through complex matrix space transformation; it only requires knowing the mass parameters and it can compensate the gravity online through simple trigonometric function transformation, which can be extended and applied to multi-joint non-parallel shaft mechanical arms.

Through simulation, experimental research and analysis, this compensation algorithm is simple to apply and can compensate the gravity well in the trajectory tracking control of a robotic manipulator.

Acknowledgments

Sponsored by the "2022 Doctoral Research Start-up Funding Project of Guizhou Polytechnic of Light Industry" (NO. 22QYBS10).

References

[1] F.A. Machorro-Fernandez, V. Parra-Vega, E. Olguin-Diaz. Active Mechanical Compensation to Obtain Gravity-Free Robots: Modeling, Control, Design and Preliminary Experimental Results[J]. Ieee Int Conf Robot, 2009: 4346-4351.

[2] C. Cho, S. Kang. Design of a Static Balancing Mechanism with Unit Gravity Compensators[J]. Ieee Int C Int Robot, 2011.

[3] A. Mashayekhi, A. Nahvi, M. Yazdani, M. Mohammadi Moghadam, M. Arbabtafti, M. Norouzi. VirSense: a novel haptic device with fixed-base motors and a gravity compensation system[J]. Industrial Robot: An International Journal, 2014, 41(1): 37-49.

[4] S.-H. Kim, C.-H. Cho. Incomplete gravity compensator f or a 4-DOF manipulator[J]. Journal of Mechanical Science and Technology, 2015, 29(10): 4417-4426.

[5] C. Cho, S. Kang. Design of a Static Balancing Mechanism for a Serial Manipulator with an Unconstrained Joint Space Using One-DOF Gravity Compensators[J]. Ieee T Robot, 2014, 30(2): 421-431.

[6] YANG Y G, PENG Z, WANG T, WANG Y F. Application of fuzzy immune PD controller with fixed gravity compensation to displacement control for manipulator actuated by pneumatic muscle actuator[J]. Robot, 2008, (03): 259-263.

[7] WANG Z Z, WANG W L, JIN C S, SUN S H. Research of Robot Controlling with Fixed Gravity Compensation[J]. Machinery Design and Manufacture, 2015, (03): 166-169.

[8] XIE G H, ZHANG J C. Experimental Research on Gravity Compensation for Humanoid Robot Arm[J]. Journal of Mechanical Transmission, 2010, (07): 26-28.

[9] XIE G H, JIN M N, ZHANG J C, WU D M. Research on Gravity Compensation Using BP- NN Algorithm for Robot Arm[J]. Process Automation Instrumentation, 2012, (02): 22-24.

[10] ZHAO J, REN S J, YU Z Y. Study of Manipulator PD Control System Based on Gravity Compensation[J]. Modern Scientific Instruments, 2012, (05): 58-59.

[11] Y.C. Liu, N. Chopra. Gravity-Compensation-Driven Position Regulation for Robotic Systems Under Input/Output Delays [J]. IEEE Transactions on Control Systems Technology, 2014, 22(3): 995-1005.

[12] WANG H M, DU Z J, YAN Z Y, LIU R Q. Gravity Compensation Algorithm for Hybrid Master Manipulator[J]. Robot, 2014, (01): 111-116+128.

[13] E.-J. Kim, K. Seki, M. Iwasaki. Motion control of industrial robots by considering serial two-link robot arm model with joint nonlinearities[J]. Journal of Mechanical Science and Technology, 2014, 28(4): 1519-1527.

[14] CHEN Y J, WANG Y N. Tracking Control of Deicing Robot on Voltage Transmission Lines Based on Online Gravity Compensation[J]. Information and Control, 2013, (04): 401-406.

[15] Emmerich C, Nordmann A, Swadzba A, et al. Assisted Gravity Compensation to cope with the complexity of kinesthetic teaching on redundant robots[C]// IEEE International Conference on Robotics and Automation. IEEE, 2013:4322-4328.

[16] R. Kelly, V. Santibáñez, A. Loría. Control of robot manipulators in joint space[M]. London: Springer, 2005.