An Inverse Kinematics Solution Based on CA-CMAC and ILC for Trajectory Tracking of Redundant DOF Manipulator

To cite this article: LIU Hua-qing and LV Wang 2017 J. Phys.: Conf. Ser. 806 012018

View the article online for updates and enhancements.
An Inverse Kinematics Solution Based on CA-CMAC and ILC for Trajectory Tracking of Redundant DOF Manipulator

LIU Hua-qing and LV Wang

(1. Shanghai Institute of Satellite Engineering, Shanghai 200240, China; 2. School of Aerospace, Tsinghua University, Beijing 100084, China)

liuhuaqing@zju.edu.cn ; lvwang509@126.com

Abstract. The composite control strategy of CA-CMAC and ILC is adopted to solve the inverse kinematic problem of the redundant DOF manipulator during its real-time and high-precision tracking on the three-dimensional space target trajectory. A direct inverse model control strategy is adopted, in which CA-CMAC takes the current joint angles and the desired position increment of the manipulator as the input, and estimates the expected joint angle increments of the manipulator using the system history control experience. Then the estimated joint angles are taken as the initial value of the ILC module by which the control effect is improved iteratively. Based on the MATLAB, the tracking controls of linear and circular space target trajectories were simulated respectively. The results show that CA-CAMC and ILC composite control has better tracking precision and stability than CMAC control, while keeping the joint angles of the manipulator continuous and smooth during trajectory tracking.

1. Introduction

Inverse kinematics problem is a kind of important problem of the motion control of manipulators. Especially for the redundant DOF manipulator, the inverse kinematics problem often does not have the unique solution. At present, the method of solving the inverse kinematics problem of the manipulator can be classified into the following categories: analytic method [1], geometric method [2], iterative method [3], intelligent algorithm [4,5]. Although the analytic method can give all the roots, but the solution process is cumbersome and computationally large. The geometry method is simple, but it is only valid for some simple arms. Iteration method is suitable for most cases, but cannot provide all solutions, the calculation is huge and there is the possibility of non-convergence. Intelligent algorithms such as genetic algorithms, artificial neural networks, etc., due to real-time, stability and accuracy could not be guaranteed, so is rarely used.

The Cerebellar Model Articulation Controller (CMAC) proposed by Albus JS [6] in 1975 is a neural network model to simulate cerebellar function based on the Eccles cerebellar space-time model. CMAC is a tabular search-based adaptive neural network which can express complex nonlinear functions. It has the inherent capability of local generalization, information classification and storage, and has the advantages of fast learning speed and simple network structure. Because of the superior performance mentioned above, CMAC has better non-linear approximation ability than general neural network. On this basis, Sun [7] proposed the CA-CMAC (Credit assigned CMAC). Through the effective use of historical learning information, the learning speed is further improved.

Iterative Learning Control (ILC) was proposed by the Japanese scholar Arimoto [8] in 1984 for the repetitive motion of manipulator. The basic idea of ILC is to track the trajectory of the manipulator
repeatedly. Based on the previous experience of the control system, the control variables can be modified according to the error between the actual trajectory and the expected trajectory. And gradually find an ideal input curve, making the actual trajectory of the robot arm close to the desired trajectory. ILC is applicable to the controlled object with repetitive motion properties. Theoretically, it can achieve complete tracking, and the computation is small, so it is suitable for the control of fast motion. However, ILC is sensitive to the initial state. Theoretical studies generally assume that the state of the manipulator is exactly equal to the desired state at the initial moment, or that the error is small. However, in practical applications, the initial error may be relatively large, in this case how to improve the ILC so that it can converge to the desired value, is an important issue. Many scholars have studied the ILC initial value problem, and gives the solution. Arimoto [9] studied the effect of initial positioning error on system tracking performance. Porter B [10] introduced the pulse correction to eliminate the initial error, thus achieving a complete tracing of the desired trajectory. Sun [11] proposed a learning algorithm with initial correction and terminal attraction to achieve complete tracking of the desired trajectory on the predetermined interval in the presence of initial error.

Aiming at the real-time, stable and high-precision requirement of inverse kinematics solution in the trajectory tracking control of the redundant manipulator, considering the CMAC control precision is not high enough but it has the advantage of real-time and generalization ability, while the ILC control precision is high but there is an initial value problem, a CA-CMAC and ILC composite controller is designed to realize the real-time and high-precision inverse kinematics tracking control of the redundant DOF manipulator to the target trajectory by complementing each other.

2. Control object description

The research object of this paper is redundant DOF manipulator such as the industrial manipulator with six degrees of freedom as shown in Fig. 1 (a). To simplify the analysis while retaining the ability of the manipulator to track target trajectory in three-dimensional space, two degrees of freedom of the wrist are neglected, thus, it is abstracted as the four-joint serial manipulator structure shown in Fig. 1 (b). Regardless of the posture constraints of the end-effector, the manipulator has four degrees of freedom and the workspace dimension is three, so it is a redundant manipulator.

As shown in Fig. 1 (b), the coordinate system o-xyz is fixed to the base of the manipulator, and the arm lengths of the four joints are \(l_0, l_1, l_2, l_3\) respectively. The joint \(l_0\) can be rotated around the z-axis, its joint angle is denoted as \(\theta_0\), and the initial position is directed to the x-axis positive direction. The joint \(l_1, l_2, l_3\) respectively, has and only has one degree of freedom, and the three joints always move in the same plane, the corresponding joint angles are denoted as \([\theta_1, \theta_2, \theta_3]\), and the initial position of each points to the extension line direction of the previous joint. Thus, we could obtain the position of end-effector the manipulator:

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \begin{bmatrix}
[l_0 + l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3)] \cos \theta_0 \\
[l_0 + l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3)] \sin \theta_0 \\
l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)
\end{bmatrix}
\]  

(1)
The goal of the trajectory tracking kinematic control of the manipulator is to control the joint angles $[\theta_0, \theta_1, \theta_2, \theta_3]$ so that the position of the end-effector $\mathbf{R}_e$ tracks the target trajectory in three-dimensional space in real time and accurately. Since the four joints are always in the same plane, the joint angle $\theta_0$ has a unique solution:

$$
\begin{bmatrix}
X_d & Y_d
\end{bmatrix} = \begin{bmatrix}
\cos \theta_0 & \sin \theta_0
\end{bmatrix} \sqrt{X_d^2 + Y_d^2}
$$

Thus, the original problem is converted to the inverse kinematics problem of the planar three-joint manipulator, where the coordinate of the target trajectory is changed to $\mathbf{r}_d$:

$$
\mathbf{r}_d = \begin{bmatrix}
x_d \\
y_d
\end{bmatrix} = \begin{bmatrix}
\sqrt{X_d^2 + Y_d^2} - l_0 \\
Z_d
\end{bmatrix}
$$

The control variable is $\theta_1, \theta_2, \theta_3$. The actual position of the end-effector is:

$$
\begin{bmatrix}
x \\
y
\end{bmatrix} = \begin{bmatrix}
l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\
l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)
\end{bmatrix}
$$

The control problem can then be described as follows: by controlling the three joint angles so that the actual position of the end-effector of the manipulator tracks the target trajectory, i.e. the tracking error tends to zero.

### 3. Design of the CA-CMAC and ILC Composite Controller

The composite controller consists of a CA-CMAC module and an ILC module, as shown below.

![Control system block diagram](image)

**Figure 2.** Control system block diagram

The control system works as follows: At control period $t$, the expected position of the end-effector of the manipulator $\mathbf{r}_d(t+1)$ is taken as the input signal of the system. $\mathbf{r}_d(t+1)$ minus the actual position of the end-effector, and the desired position increment $\Delta \mathbf{r}_i = [\Delta x_i, \Delta y_i]$ is obtained. The actual joint angles $\mathbf{\theta}(t) = [\theta_1, \theta_2, \theta_3]$ of the manipulator at period $t$ and $\Delta \mathbf{r}_i = [\Delta x_i, \Delta y_i]$ are merged into a vector $\mathbf{C}$ as the input to the CA-CMAC neural network controller ($p$ is the proportional coefficient):

$$
\mathbf{S} = [\theta_1, \theta_2, \theta_3, p\Delta x_i, p\Delta y_i]
$$

### 3.1. CA-CMAC controller

CA-CMAC is a kind of look-up table adaptive neural network which uses linear structure to express complex nonlinear function. It stores the learning data in the memory and outputs the sum of the weights of the activated units. CA-CMAC network consists of input space $S$, quantization space $M$, virtual memory space $AC$, the actual storage space $AP$ and output layer $Y$. The mapping between the four layers is as follows [6].
3.1.1. **S to M mapping.** The continuous variable $s$ is discretized as:

$$ m_k = \left\lfloor \frac{s_k - s_{k,\text{min}}}{Q_k} \right\rfloor, k = 1, \ldots, 5 $$

where $Q_k$ is the quantization step number; $s_k$ is the $k$th element of the input vector; and \( \left\lfloor \right\rfloor \) is the round-up operator.

3.1.2. **M to AC mapping.** The addresses of the activated units in the AC are coded as follows:

$$ ac_i = \sum_{k=1}^{i} a^i_k + i \cdot Q^i, \quad i = 1, \ldots, C $$

among them:

$$ a^i_k = \left\lfloor \frac{m_k}{\Delta_k} + \text{ofs}(i) \right\rfloor Q^{i-1} $$

$$ \text{ofs}(i) = (i-1) \cdot \frac{\Delta_k}{\bar{C}}, \quad i = 1, \ldots, C $$

The meaning of each symbol in the above formula is:
\( \Delta_k \) : represents the quantization width of the $k$-th dimension input component $s_k$;
\( \backslash \) : indicates the remainder operator;
\( Q \) : quantification level.

3.1.3. **AC to AP mapping.** Since the virtual storage space AC requires a large number of storage units, which is physically unrealistic, the storage addresses are compressed by the mapping. Using the hash coding technology to encode the unit address within the AP space (\( ap_i, i = 1, \ldots, C \) represents the address of the $C$ activated C cells in AP; \( h \) is the hash coefficient):

$$ ap_i = \left\lfloor \ln((ac + 1) \cdot h) \backslash N_{AP} \right\rfloor, i = 1, \ldots, C $$

Since the CA-CMAC output is the angular increment of the three joints, the weight matrix is designed as a three-column matrix. After the space compression, the actual storage space has only $3N_{AP}$ storage units.

3.1.4. **AP to Y mapping.** According to the unit address, read the network weights in the activated unit of the AP space, add and get the output:

$$ \Delta \theta_j = \sum_{i=1}^{C} w(ap_i, j), \quad j = 1, 2, 3 $$

Where $w(ap_i, j)$ denotes the $ap_i$-th row and the $j$-th column of the weight matrix $w$.

3.2. **ILC controller**

To further improve trajectory tracking accuracy, an ILC controller is cascaded behind the CA-CMAC controller. At each control period $t$, the output value $\Delta \theta(t+1)$ is obtained by a series of mappings within the CA-CMAC controller. The value is then entered into the ILC controller as its initial value. The role of the ILC controller is to optimize iteratively around the initial value. The optimization objective function is defined as follows:
\[
\min V(\Delta \theta(t+1)) = \mathbf{e}'e/2
\]

In this optimal target sense, the optimal joint angle increments are the set of values that minimize the cost function, denoted as \( \Delta \theta = [\Delta \theta_1, \Delta \theta_2, \Delta \theta_3]^T \). The steepest gradient descent method is used to seek the optimal joint angle increments by a number of iterations:

\[
\Delta \theta^{k+1}(t+1) = \Delta \theta^k(t+1) - \eta \left( \frac{\partial V}{\partial \Delta \theta^k(t+1)} \right)^T = \Delta \theta^k(t+1) + \eta [\mathbf{e}'(t+1)]^T \mathbf{J} \tag{13}
\]

It can be seen that the above equation is a P-type learning control law. Where \( k = 0,1,2... \) is the number of iterations; \( \eta' \) is the learning rate of ILC, taken as a positive value, to prevent divergence, the learning rate should be less than one; \( \mathbf{J} \) is the Jacobian matrix defined as \( \mathbf{J} = \partial \mathbf{r} / \partial \mathbf{\theta} \). Theoretically, after a sufficient iteration, \( \Delta \theta^\ast \) will completely approach the ideal value, but in view of the real-time control, only a limited number of iterations are implemented.

3.3. CA-CMAC weight adjustment algorithm

At the end of each control period, the weights of the CA-CMAC network should be adjusted. The purpose of weight adjustment is to improve the control precision, and finally to make the optimization objective function based on the trajectory tracking error converge to the minimum. The manipulator end-effector position corresponding to the joint angle increment \( \Delta \theta \) is \( \mathbf{r}(t+1) \), and the error of \( \mathbf{r}(t+1) \) with respect to the target track position \( \mathbf{r}_d(t+1) = \mathbf{e}(t+1) \). The CA-CMAC network weight is adjusted according to the tracking error. The adjustment algorithm is as follows: firstly, we use the sum of squares of errors to define the index function \( V = \mathbf{e}'\mathbf{e}(t+1)/2 \), thus the deviation of the index function to the network weights is:

\[
\frac{\partial V}{\partial \mathbf{w}} = \mathbf{e}'\frac{\partial \mathbf{e}(t+1)}{\partial \mathbf{\theta}} = \mathbf{e}'\frac{\partial \mathbf{e}(t+1)}{\partial \mathbf{\theta}} \frac{\partial \mathbf{\theta}(t+1)}{\partial \mathbf{w}} = -\mathbf{e}'\mathbf{J}
\]

The conventional weight adjustment algorithm distributes the correction values evenly to the activated physical storage cells so that the activated cells that do not need to adjust the weights excessively are adjusted repeatedly due to the principle of average distribution, while those that require multiple adjustments has not been fully adjusted, resulting in low learning efficiency. In order to improve the learning efficiency, the CA-CMAC was proposed [7]. The historical information of the physical storage unit determines the magnitude of its weight adjustment: for any unit, the more it is activated in the past, the higher its credibility and should be assigned a small correction value; on the contrary, the less it is activated in the past, the less reliable it is, and should be assigned a larger correction value. In this paper, the steepest descent method based on credibility assignment is used to adjust the weights, where the learning rate \( \eta \) is positive [7]:

\[
w_{ij,i+1} = w_{ij,i} - \eta \sum_{j=1}^{C} \frac{C [f(i)+1]}{\sum_{j=1}^{C} [f(j)+1]} \frac{\partial V}{\partial \mathbf{w}} = w_{ij,i} + \eta \sum_{j=1}^{C} \frac{C [f(i)+1]}{\sum_{j=1}^{C} [f(j)+1]} \mathbf{e}(t+1)^T \mathbf{J} , i=1,...,C \tag{15}
\]

Among then, \( I \) represents the set of \( C \) physical memory cells being activated and \( f(i) \) represents the number of history activations of the \( i \)th activated physical memory cell.
4. Simulation verification

The arm lengths of the four joints are taken as $l_0 = 0.1\text{m}$, $l_1 = 0.5\text{m}$, $l_2 = 0.5\text{m}$, $l_3 = 0.1\text{m}$, respectively. The generalization factor of the CA-CMAC $C = 10$, the total number of cells in memory $N_{\text{AP}} = 3000$, the hash coefficient $h = 12345$. The quantization levels of each component of input variable is taken as 360, and the learning rate $\eta = 0.1$, and the learning rate of the ILC $\eta’ = 0.5$. Base on MATLAB platform, the tracking of linear and circular trajectories of the redundant DOF manipulator under the composite control of CA-CMAC and ILC is simulated. The control precision of a complete trajectory tracking control process is defined as the root mean square of the tracking error $e$, noted as $e$. Let the four joint angles be zero at the initial moment, and after 1000 training periods, we get the simulation results as shown below.

![Figure 3. Linear trajectory tracking](image)

![Figure 4. Circular trajectory tracking](image)

![Figure 5. The optimization process of the linear trajectory tracking accuracy](image)

![Figure 6. The optimization process of the circular trajectory tracking accuracy](image)

![Figure 7. After training, the time-history curves of each joint angle in the tracking process of the linear trajectory (angle unit: °; time unit: dimensionless)](image)

![Figure 8. After training, the time-history curves of each joint angle in the tracking process of the circular trajectory (angle unit: °; time unit: dimensionless)](image)

It can be seen from the Figs.5 and 6 that with the increase of the number of training periods, the tracking error of the manipulator to the target trajectory can be reduced gradually and stabilized to a certain level, which indicates that the composite controller has a certain convergence. Comparing the
red and blue curves shown in Figs. 5 and 6, it can be seen that the accuracy of the composite control is more than one order of magnitude higher than that of the CMAC control, and the stability is also stronger. This is because: 1, the weight assignment algorithm based on the Credibility Assignment is adopted; 2, the addition of the ILC controller not only enhances the control algorithm stability, but also improves the trajectory tracking accuracy. It can be seen from Fig.7 and Fig.8 that the time-history curves of the four joint angles in the two examples are all smooth, which indicates that the manipulator moves smoothly during trajectory tracking and the energy consumption is small. Once the weight adjustment is complete, when one input the target trajectory, the controller only needs to look up the table to get the appropriate control input of the manipulator, so it has good real-time performance.

5. Conclusion
The output of the CA-CMAC provides an initial iteration value for the ILC, which solves the problem of the ILC initial value. In addition, the cascade of the ILC to the CA-CMAC not only restrains the divergence phenomenon caused by over-learning of the CA-CMAC, enhances the stability of control, but also improves the control precision and realizes the complementary effect of the CA-CMAC and the ILC. Considering that the composite controller is simple and easy to implement, it has certain application value.

References