PD Iterative Self-learning Control for 3R Plane Robot Trajectory Tracking

Hongwei Gao1,2, Kun Hong2, Jinguo Liu1, Yuquan Leng1, Chuanyin Liu2
1State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, China
2School of Information Science and Engineering, Shenyang Ligong University, China
*Corresponding author, e-mail: ghw1978@sohu.com

Abstract

In order to improve the speed and accuracy of the trajectory tracking control for 3R plane robot, a PD iterative self-learning control algorithm is proposed based on the PD iterative control algorithm. The error of target value and the actual value of this iteration is introduced into the PD learning gain to make the PD learning gain becomes a function of the error and to achieve the effect of self-learning. Simulation analysis of planar 3-link robot trajectory control shows that the proposed algorithm is better than unmodified in speed ability and accuracy.

Keywords: PD iterative control, self-learning, planar 3-link, trajectory tracking

1. Introduction

According to tracking problem, a variety of design methods are provided in control theory and many good results are achieved, but tracking task is achieved asymptotic by the vast majority of them [1-4]. In order to achieve the fully tracking of expect output, Iterative Learning Control (ILC) method was proposed by Arimoto et al. in 1984[5]. Significant achievements are made in the development of ILC, which is applied in a wide range of industries, including robot [6, 7], rapid thermal processing [8, 9], industrial process control, functional neuromuscular stimulation [10]. Objective of ILC is to make the system tracking error gradually decreases after repeated running, and fully track the desired trajectory ultimately. The experience of the previous control of the control system is used. The deviation of actual output information measured and the target track given in advance is used to correct the not ideal control signals. An ideal input is founded by a relatively simple learning algorithm to produce the desired movement of the controlled object [11, 12]. In recent years, significant progresses are made in convergence, robustness and applied research on learning [13, 14].

The PD iterative control is the most common class of iterative learning control algorithm, the error of target value and the actual value of this iteration is introduced into the PD learning gain to make the PD learning gain which is a fixed value in the conventional algorithm becomes a function of the error to achieve the effect of self-learning. Simulation analysis of planar 3-link robot trajectory control shows that the proposed algorithm is better than unmodified in speedability and accuracy.

The paper is organized as follows. PD Iterative Self-learning Control algorithm is given in Section 2. In Section 3, the static analysis of planar 3-link robot is discussed. Finally, simulation analysis of planar 3-link robot trajectory control is discussed in Section 4.

2. Algorithm Description

2.1. PD Iterative Control

D type algorithm is sensitive to the high frequency interference of the error signal due to the differential action. This problem can be resolved by the P type control algorithm. The maximum error can be reduced and the tracking accuracy can be improved by PD type ILC algorithm. The conventional PD type iterative control law:
\[
u_{k+1}(t) = u_k(t) + k_p e_{k+1}(t) + k_d \dot{e}_{k+1}(t)
\]

(1)

Where \(k_p\) and \(k_d\) is the PD type iterative control gain, 
\(e_{k+1}(t) = y_d(t) - y_{k+1}(t)\).

### 2.2. PD Iterative Self-Learning Control

Conventional \(k_p\) and \(k_d\) is replaced of functions of 
\(e_{k+1}(t)\), namely: 
\(k_p = k_p(e_{k+1}(t))\), 
\(k_d = k_d(e_{k+1}(t))\) on the basis of conventional PD type iterative learning control. Therefore, the improved PD-type iterative self-learning control law is:

\[
u_{k+1}(t) = u_k(t) + k_p(e_{k+1}(t)) e_{k+1}(t) + k_d(e_{k+1}(t)) \dot{e}_{k+1}(t)
\]

(2)

### 2.3. Determination of \(k_p(e_{k+1}(t))\) and \(k_d(e_{k+1}(t))\):

Nonlinear PID tuning is analyzed by Yongli Xiao et al [15] by using step response curve of the general system, shown in Figure 1. \(k_p(e_{k+1}(t))\) and \(k_d(e_{k+1}(t))\) are constructed based on this analysis thinking.

![Figure 1. Step response curve of the general system](image)

**a. The proportional gain \(k_p\)**

To guarantee a faster response speed, \(k_p\) should initially be larger when the response time is 
\(0 \leq t \leq t_1\). In order to reduce overshoot, \(k_p\) is desired to decreases gradually as 
\(e_{k+1}(t)\) gradually decreases, which makes the system inertia gradually weakened and will not have a big overshoot. \(k_p\) is desirable to gradually increase in order to increase the role of reverse control and reduce the overshoot between \(t_1\) and \(t_2\). \(k_p\) is expected to gradually decrease in order to make the system return to the steady state point as soon as possible, and no longer generate a large inertia between \(t_2\) and \(t_3\). \(k_p\) is expected to gradually increase between \(t_3\) and \(t_4\), the same as between \(t_1\) and \(t_2\). Obviously, according to the above variation, changing of \(k_p\) as 
\(e_{k+1}(t)\) changes is shown in Figure 2. The nonlinear function can be constructed as follows:

\[
k_p(e_{k+1}(t)) = a_p + b_p \left(1 - \sec h(c_p e_{k+1}(t))\right)
\]

(3)

Where \(a_p, b_p, c_p\) are positive real constants. Maximum value of \(k_p\) is \(a_p + b_p\) when, 
minimum value of \(k_p\) is \(a_p\) when \(e_{k+1}(t) = 0\). The rate of change of \(k_p\) can be adjusted by adjusting the size of \(c_p\).

**b. The differential gain \(k_d\)**

In order to inhibit the generation of overshoot and does not affect the speed of response, \(k_d\) should gradually increase when the response time is 
\(0 \leq t \leq t_1\). \(k_d\) is desirable to continually increase in order to increase the role of reverse control and reduce the overshoot between \(t_1\) and \(t_2\). \(k_d\) should gradually decreases at \(t_2\). In order to inhibit the generation of
overshoot, $k_d$ should gradually increases again between $t_2$ and $t_3$. The symbol of error change rate should be considered when nonlinear function of $k_d$ is constructed according to requirements of changings of $k_d$. Changing of $k_d$ is shown in Figure 3 and the nonlinear function structured is:

$$k_d(e_{k+1}(t)) = a_d + b_d \frac{e_{k+1}(t)}{1 + c_d \exp(d_d \cdot e_{k+1}(t))}$$ (4)

Where $a_d, b_d, c_d, d_d$ are positive real constants. $a_d$ is the minimum value of $k_d$, $a_d + b_d$ is the maximum value of $k_d$. $k_d = a_d + b_d/(1+c_d)$ when $e_{k+1}(t)=0$. Rate of change of $k_d$ can be adjusted as $a_d$ is adjusted [16].

So the law of improved PD iterative self-learning control is:

$$u_{k+1}(t) = u_k(t) + k_p \cdot e_{k+1}(t) \cdot e_{k+1}(t) + k_d \cdot e_{k+1}(t) \cdot \dot{e}_{k+1}(t)$$

$$= u_k(t) + (a_d + b_d(1 - \text{sech}(c_d \cdot e_{k+1}(t)))) \cdot e_{k+1}(t) + (a_d + b_d/(1 + c_d \exp(d_d \cdot e_{k+1}(t)))) \cdot \dot{e}_{k+1}(t)$$ (5)

3. Static Analysis of Planar 3-link Robot

A plane three-link is regarded as the simulation object, organizational chart of which is shown in Figure 4 [17].
3.1. Kinematics Equation

Ector equation of plane 3R robot can be seen as follows:

\[ R_{pl} = R_1 + R_2 + R_3 \]  \hspace{1cm} (6)

Where \( R_{pl} \) is the straight line distance of the end of the robot relative to the origin, \( R_1 \) is the length of the connecting rod 1, \( R_2 \) is the length of the connecting rod 2, \( R_3 \) is the length of the connecting rod 3, scalar equation that \( R_{pl} \) corresponds to x and y axis is:

\[ x_{pl} = r_i \cos \theta_1 + r_2 \cos (\theta_1 + \theta_2) + r_3 \cos (\theta_1 + \theta_2 + \theta_3) \]  \hspace{1cm} (7)

\[ y_{pl} = r_i \sin \theta_1 + r_2 \sin (\theta_1 + \theta_2) + r_3 \sin (\theta_1 + \theta_2 + \theta_3) \]  \hspace{1cm} (8)

Where \( x_{pl} \) is the x value of the end of rod, \( y_{pl} \) is the y value of the end of rod, \( \theta_i \) is the angle of the rod i relative to the rod (i -1), \( r_i \) is the length of rod i.

Equation (7) and Equation (8) are second derivative to get the equation of velocity.

3.2. Dynamics Equation

Diagram of rod 1 is shown in Figure 5. Dynamic of the rod 1 is analyzed:

![Diagram of rod 1](image)

Figure 5. Diagram of rod 1

Three dynamic equations of linkage 1 can be deduced from Figure 5, and dynamic equations of other two rods are similar to rod 1:

\[
\begin{align*}
F_{01,x} + F_{21,x} &= m_1 a_{c1,x} \\
F_{01,y} + F_{21,y} - m_1 g &= m_1 a_{c1,y} \\
M_1 - M_2 &= F_{21,x} r_{c1} + F_{21,y} r_{c1} - m_1 g r_{c1} = I_1 a_1
\end{align*}
\]  \hspace{1cm} (9)

Equations of payload force are

\[
\begin{align*}
m_{pl} \ddot{x}_{pl} &= -F_{43,x} \\
m_{pl} \ddot{y}_{pl} &= -F_{43,y} - m_{pl} g
\end{align*}
\]  \hspace{1cm} (10)

where \( m_i \) is the quality of the rod i, \( m_{pl} \) is the quality of the payload, \( g \) is the acceleration due to gravity, \( I_1 \) is the moment of inertia of rod i, \( a_{c1,x} \) is the angular acceleration of center of gravity of rod i in the x direction, \( a_{c1,y} \) is the angular acceleration of center of gravity of rod i in
the y direction, $F_{ij,x}$ is force of rod i relative to rod j in the x direction, $F_{ij,y}$ is force of rod i relative to rod j in the y direction, $M_i$ is torque of revolute pair i.

4. Simulation Analysis

A plane three-link is regarded as the simulation object, parameter of the robot mechanical system is assumed as $m_1=4.0kg$, $m_2=3.0kg$, $m_3=3.0kg$, $r_1=r_2=1m$, $r_3=0.8m$, $r_{c1}=r_{c2}=0.5m$, $r_{c3}=0.4m$, $l_1=0.08kg*m^2$, $l_2=0.2kg*m^2$, $l_3=0.25kg*m^2$, $g=9.8067m/s^2$. In the simulation, rotation angle trajectory of the three joint is $q_{d1} = \sin (\pi * t / 6)$; $q_{d2} = \sin (t)$; $q_{d3} = (\pi / 8) * t$, the initial position is $x_0 = 2.8m$, $y_0 = 0m$. Simulation platform is shown in Figure 6, which is composited by the input unit, control unit, control unit and output unit. A detail description of a single joint iteration unit is shown in Figure 7.

![Figure 6. Simulation diagram](image)

![Figure 7. Iterative part](image)

Tracking effect in 1 iteration, 5 iterations, 10 iterations and 30 iterations respectively of three joints are shown in Figure 8. It can be clearly seen that tracking is poor in 1 iteration and the effect gets better and better through multiple iterations. It tracks on a given trajectory basically when the number of iterations reaches to 30 times by a number of simulations.

Change graph of kp and kd of three joints in the 30th iteration is shown in Figure 8. It can obviously be seen that kp and kd change in the tracking process. It can be seen from Figure 9(a) that variation of kp is relatively large at the beginning, because the e is relatively large and gradually stabilizes.
The path tracking effect diagram after 30 iterations is shown in Figure 10. It can be seen that effect in PD iterative self-learning control proposed is better than PD iterative control, and it can tracks the given target trajectory more accurately.

![Diagram of Joint 1 Tracking](image1)

(a) joint 1

![Diagram of Joint 2 Tracking](image2)

(b) joint 2

![Diagram of Joint 3 Tracking](image3)

(c) joint 3

Figure 8. Tracking effect diagram of each joint in the different number of iterations
Figure 9. $k_p$ and $k_d$ of three joints in the 30th iteration

Figure 10. Path tracking effect diagram
Error convergence process in 30 iterations is shown in Figure 11. It can be seen that convergence rate in PD iterative self-learning control proposed is better than PD iterative control whether in the X direction or Y direction, as well as the final error value is smaller.

![Graph of error convergence process in X direction](image1)

![Graph of error convergence process in Y direction](image2)

Figure 11. Error convergence process in 30 iterations process

5. Conclusion

Iterative control of three-link robot is researched in depth. The basic principle of the PD type iterative learning control is understood firstly. Then a PD type iterative self-learning control algorithm is proposed based on the PD type iterative learning control algorithm. The error $e_{k+1}(t)$ of target value and the actual value of this iteration is introduced into the PD learning gain to make the PD learning gain $k_p$ and $k_d$ become error functions $k_p(e_{k+1}(t))$ and $k_d(e_{k+1}(t))$ and to achieve the effect of self-learning. Simulation analysis of planar 3-link robot trajectory control shows that the proposed algorithm is better than unmodified in speedability and accuracy.

Acknowledgements

This work is partly supported by the National Natural Science Foundation of China (Grant Nos. 51175494, 61128008), the State Key Laboratory of Robotics Foundation (Grant No. O8A120S), Liaoning Province Innovation Team Project (Grant No. LT2012005), and Liaoning Province Educational Office Foundation of China (Grant No. L2011038).
References


