IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 4, 2020, accepted February 25, 2020, date of publication March 2, 2020, date of current version March 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2977687

Estimation-Based Quadratic Iterative Learning Control for Trajectory Tracking of Robotic Manipulator With Uncertain Parameters

MINFENG ZHU[1], LINGJIAN YE[2], (Member, IEEE), AND XIUSHUI MA[2]
[1]College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China
[2]Ningbo Institute of Technology, Zhejiang University, Ningbo 315100, China

Corresponding author: Lingjian Ye (lingjian.ye@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673349, in part by the Foundation of Key Laboratory of Advanced Process Control for Light Industry (Jiangnan University, APCLI1802), in part by the Ningbo Natural Science Foundation under Grant 2018A610188, and in part by the Talent Project of Zhejiang Association of Science and Technology under Grant 2017YCGC014.

ABSTRACT In this paper, we consider iterative learning control for trajectory tacking of robotic manipulator with uncertainty. An improved quadratic-criterion-based iterative learning control approach (Q-ILC) is proposed to obtain better trajectory tracking performance for the robotic manipulator. Besides of the position error information, which has been used in existing Q-ILC methods for robotic control, the velocity error information is also taken into consideration such that a new norm-optimal objective function is constructed. Convergence and error sensitivity properties for the proposed method are also analyzed. To deal with uncertainty, the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are incorporated for estimation of uncertain parameters by constructing extended system states. The performances between the two filters are also compared. Simulations on a 2DOF Robot manipulator demonstrate that the improved Q-ILC with parameter estimators can achieve faster convergence and better transient performance compared to the original Q-ILC, in the presence of measurement noise and model uncertainty.

INDEX TERMS Iterative learning control, quadratic-performance-criterion, norm-optimal, EKF, UKF, robot manipulator, online dual estimation.

I. INTRODUCTION

Robot manipulators are widely used in the modern manufacturing such as injection molding, automobile assembly industry and spinning, where there are plenty of repetitive operations. This type of complex operation requires the robot to guarantee high performance even in the presence of measurement noise, external disturbances and model uncertainties. The work in [1] and [2] proposes a robotic arm trajectory tracking method based on sliding mode control, which can quickly achieve trajectory tracking error convergence. Although significant developments have been achieved by applying these traditional control methods, some challenges still exist. These challenges can be solved by adopting recursive learning methods to adapt to complex external environments and to deal with unknown mechanism model modeling problems. This efficient method can be characterized as iterative learning control (ILC), experience and data from previous iteration are utilized to improve the trajectory tracking performance of next trial. Since firstly proposed by Arimoto *et al.* [3], ILC has become a popular approach for trajectory tracking, especially in the automatic process with repetitive characteristics, where iterative learning can continuously update the control by using the mechanism of repetition. An extensive overview over the field of ILC is available in [4]–[6]. In general, research in trajectory tracking of robot manipulator has made significant progress over the past period [7]–[12]. For example, researchers proposed to combine existing traditional control methods with ILC, such as the fuzzy adaptive ILC [7], [8], where the robustness of fuzzy control and the high-precision control performance of ILC are absorbed. In order to deal with plant uncertainty, ILC was combined with adaptive control for varying operating conditions [12]. An adaptive ILC (AILC) scheme based on

The associate editor coordinating the review of this manuscript and approving it for publication was Behnam Mohammadi-Ivatloo.

the proportional-derivative feedback structure was proposed in [13]. Another improved adaptive ILC algorithm based on an estimation procedure was proposed in [14]. Two-degree-of-freedom ILC method was used in [15] to compensate the joint flexibility of industrial robot manipulator, which achieved more effective performance compared to the dual-stage ILC that has been proposed previously in [16]. More advanced learning strategies have been issued for robotic trajectory tracking control. The ILC was combined with system identification technique in [17], aiming to model the controlled system using identification methods with a data-driven approach such that the model-based ILC can more efficiently solve tracjectory tracking problem of industrial robots. In [18], a system identification method based on batch adaptive control algorithm [19] was applied to the iterative learning control system for online parameter identification, which is appealing for practical usage. Neural networks and iterative adaptive dynamic programming (ADP) algorithm were applied in [20] to identity the unknown system dynamics, approximate performance function and caculate control inputs, respectively. In [21], the nonlinear parametric time-delay uncertainties and non-zero initial error are considered, an adaptive ILC scheme based on Lyapunov approach is explored to achieve perfect tracking performance and extend the application of ILC in industry.

Among various ILC oriented methods, a class of optimization-based methods emerge as a hot topic, which considers the control task as an optimization problem. Moore [22] issued to update the input signal of the next operation by solving an quadratic norm-optimal problem that constructed by tracking error. This algorithm was later referred as the Q-ILC. A more effective Q-ILC algorithm was proposed in [23], taking the adjacent input variations into account. In article [24], different types of stochastic disturbance are considered in the bias in a linear static model, and Q-ILC algorithms are derived. On the other hand, since uncertainty is widely present in robot system, the Kalman filter was typically considered to enhance the ILC performance [25]–[27]. In these methods, the Kalman filter was used to estimate the system states. The work in [28] gave a comprehensive comparison between the Q-ILC and the estimation-based ILC combined with Kalman filter (K-ILC). The work in [29] expanded on the basis of full-order unknown input observer (FOUIO) and reduced-order unknown input observer (ROUIO), and proposed to use state vectors and input vectors to form novel observers for system state estimation. A new adaptive ILC that incorporate a Recursive Least Squares (RLS) algorithm was proposed in [30], the learning gain tuned iteratively along the learning axis and pointwisely along the time axis, the new AILC achieves pointwise convergence over a finite time interval. In [14], the AILC algorithm based on Kalman filter was applied on industrial robot, demonstrates the appeal of adaptive iterative learning in practical applications.

The approach presented in this paper is composed of two novel schemes that encompass both the estimation and control. In the estimation part, the Unscented Kalman Filter (UKF) is employed for simultaneous estimation for system states and uncertain parameters. Compared with the classical Extended Kalman Filter (EKF), UKF is more efficient for state estimation of highly nonlinear system, as a feedforward signal for iterative learning control, hence speeding up convergence. In the control part, we propose an improved Q-ILC scheme, that is, a new cost function by adding the velocity error information in the optimization formulation. Compared to existing methods that only use the position error information for industrial robots, the new scheme utilizes more available observations such that the control performance can be improved, as verified by both theoretical analysis and simulation studies. In summary, the integration of improved Q-ILC algorithm and the UKF-based dual estimation is able to greatly improve the trajectory tracking performance of uncertain robot systems, also, maintain robustness to noise, disturbance and model errors. The general algorithm framework is given in Fig. 1.

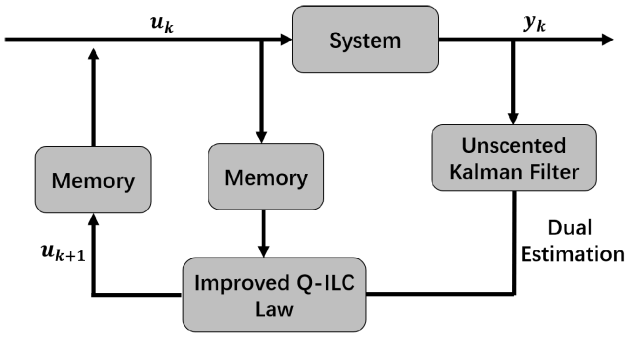


FIGURE 1. The general Q-ILC framework proposed in this paper.

The article is organized as follows: In Section II, the original Q-ILC algorithm is reviewed. Then, in Section III, an improved Q-ILC algorithm is proposed, the convergence properties and sensitivity performance are also derived. Section IV introduces the dual estimation strategy for the uncertain robot system, both the EKF and UKF are investigated. The performances are compared and the results are analyzed with conclusive remarks. Simulation studies for the 2DOF robot are conducted in Section V to show the advantages of the new approach. Finally, Section VI concludes this paper and proposes future perspectives.

II. ORIGINAL Q-ILC ALGORITHM

In this section, the original Q-ILC algorithm will be introduced briefly, consider the following sampled-time system:

$$x(m+1) = Ax(m) + Bu(m),\ x(0) = x_0, \tag{1}$$

$$y(m) = Cx(m). \tag{2}$$

where m denotes the time index, the state matrices A, B, C are assumed to be time-invariant for simplicity. However, it is straightforward for the formulation of a time-varying system by using time-dependent matrices.

Robotic manipulator operates repeatedly over a finite time span $[0, N]$. Define the stacked outputs and inputs as

$$y_k^T = [y_k^T(1)\, y_k^T(2)\, \ldots\, y_k^T(N)], \tag{3}$$
$$u_k^T = [u_k^T(0)\, u_k^T(1)\, \ldots\, u_k^T(N-1)]. \tag{4}$$

where k represents the iteration index. Denoting the mapping between the input and output as:

$$y_k = Gu_k + G_d d_k + b \tag{5}$$

where

$$G \triangleq \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & \cdots & CB \end{bmatrix}$$

is the sensitivity between the stacked vector u_k and y_k, calculated by linearizing the robot model along the current trajectory. G_d and d represent the disturbance matrix and disturbance signals respectively, b represents a constant vector.

In general, the control objective of a robot is to design a control law that asymptotically converges the actual position trajectory (y_k) to the desired trajectory (y_d). The error expression has been derived in [4]:

$$\bar{e}_{k+1} = \bar{e}_k - G\Delta u_{k+1} + v_k, \tag{6}$$
$$e_k = \bar{e}_k + n_k. \tag{7}$$

where e_k represents the tracking error, v_k and n_k are zero-mean (batch-index-wise) independent identically distributed random vector sequences. The zero tracking error is unable to achieve when external disturbances exist, a reasonable idea is to limit the final error to an acceptable range. Moore [22] proposed to minimize the output tracking error by solving an optimization problem, where the objective function is defined as the norm of output errors. That is, solving the subproblem

$$\min_{u_{k+1}} \|e_{k+1}\|_2$$

whose solution is the optimal control input for the $(k+1)$th trial. Alternatively, Lee *et al.* [4] proposed a more complete optimality criteria that considers both output trajectory errors and adjacent input changes,

$$\min_{\Delta u_{k+1}} \frac{1}{2}\{e_{k+1}^T Q e_{k+1} + \Delta u_{k+1}^T R \Delta u_{k+1}\}$$

where $e_{k+1} = y_d - y_{k+1}$ and Δu_{k+1} are output trajectory errors and input change for the $(k+1)$th iteration, respectively. The weight matrices Q and R are generally positive-definite.

Lee *et al.* [4] derived the Direct-error-based Q-ILC algorithm as follows:

$$u_{k+1} = u_k + H^Q e_k \tag{8}$$
$$H^Q = (G^T Q G + R)^{-1} G^T Q \tag{9}$$

where H^Q is the closed-form solution of the original Q-ILC optimization function.

III. IMPROVED Q-ILC DESIGN FOR TRAJECTORY TRACKING

A. IMPROVED Q-ILC

It is noted that existing Q-ILC methods for trajectory tracking of robots only use position error [23]. Generally, better performance can be achieved by taking full advantage of the known information. Therefore, this paper proposes to add the velocity information, which is often measured in the robot system. The following new quadratic criterion based optimization problem is considered:

$$\min_{\Delta u_{k+1}} J = \frac{1}{2}\{e_{\theta,k+1}^T Q e_{\theta,k+1} + \Delta u_{k+1}^T R \Delta u_{k+1} + e_{v,k+1}^T W e_{v,k+1}^T\} \tag{10}$$

where $e_{\theta,k+1}$ and $e_{v,k+1}$ represent angle error and angular velocity error at $(k+1)$th iteration, respectively. W denotes the weight matrix for the velocity.

To solve (10), consider a local linear model, $y_k = G_k u_k$, where y_k and u_k represent the actual joint angle position and the control torques, respectively, then

$$e_k = [e_{\theta,k}; e_{v,k}], \tag{11}$$
$$e_k = y_d - V_k u_k, \tag{12}$$
$$e_k = e_{k-1} - V_k(u_k - u_{k-1}) = e_{k-1} - V_k \Delta u_k, . \tag{13}$$

where

$$V_k = \begin{bmatrix} G_k & \\ & H_k \end{bmatrix},$$

H_k represents mapping matrix between control input and output velocity state, e_k denotes the combined representation of errors. To be more specific, it has

$$e_{\theta,k} = y_{\theta,d} - G_k u_k, \tag{14}$$
$$e_{v,k} = y_{v,d} - H_k u_k. \tag{15}$$

where $y_{\theta,d}$ and $y_{v,d}$ represent the desired joint angular trajectory and the desired joint angular velocity trajectory, respectively. Assuming that G_{k+1} and H_{k+1} are same to G_k and H_k, then

$$e_{\theta,k+1} = y_{\theta,d} - G_k(u_k + \Delta u_{k+1}), \tag{16}$$
$$e_{v,k+1} = y_{v,d} - H_k(u_k + \Delta u_{k+1}). \tag{17}$$

Combining (14) and (15),

$$e_{\theta,k+1} = e_{\theta,k} - G_k \Delta u_{k+1}, \tag{18}$$
$$e_{v,k+1} = e_{v,k} - H_k \Delta u_{k+1}. \tag{19}$$

Differentiating the objective function J:

$$\frac{\partial J}{\partial \Delta u_{k+1}} = -2{G_k}^T Q e_{\theta,k+1} + 2R\Delta u_{k+1} - 2H_k^T W e_{v,k+1} \tag{20}$$

Let $\frac{\partial J}{\partial \Delta u_{k+1}} = 0$,

$$-2{G_k}^T Q e_{\theta,k+1} + 2R\Delta u_{k+1} - 2H_k^T W e_{v,k+1} = 0, \tag{21}$$

$$\Downarrow$$

$$(G_k^T Q G_k + R + H_k^T W H_k)\Delta u_{k+1} = G_k^T Q e_{\theta,k+1} + H_k^T W e_{v,k+1}, \tag{22}$$

$$\Delta u_{k+1} = (G_k^T Q G_k + R + H_k^T W H_k)^{-1} G_k^T Q e_{\theta,k+1} + (G_k^T Q G_k + R + H_k^T W H_k)^{-1} H_k^T W e_{v,k+1}. \tag{23}$$

The closed-form solution Δu_{k+1} is derived.

To sum up, the improved quadratic-criterion-based iterative learning control law is obtained as:

$$u_{k+1} = u_k + H^{Q^*} e_k, \tag{24}$$

$$H^{Q^*} = \begin{bmatrix} H_1^{Q^*} & \\ & H_2^{Q^*} \end{bmatrix}, \tag{25}$$

$$H_1^{Q^*} = (G^T Q G + H^T W H + R)^{-1} G^T Q, \tag{26}$$

$$H_2^{Q^*} = (G^T Q G + H^T W H + R)^{-1} H^T W. \tag{27}$$

In the presence of process disturbances and measurement errors, the direct use of (12) may be inefficient. To deal with this issue, we will consider an observer-based method, as

$$u_{k+1} = u_k + H^{Q^*} \hat{e}_{k|k} \tag{28}$$

where $\hat{e}_{k|k}$ is the optimal estimation of the true trajectory tracking error, which can be obtained from filters, such as the UKF.

B. CONVERGENCE ANALYSIS

1) CONVERGENCE PROPERTY OF ERROR SEQUENCE

Since system (13) is completely observable, we infer that the observer can be designed as a stable system through Modern control theory. The convergence property is proved as follow:

From observer system model (13), we can infer

$$e_{k+1} = (I - L)e_k, \tag{29}$$

where $L \triangleq VH^{Q^*}$

$$\begin{bmatrix} V_c \\ 0 \end{bmatrix} \triangleq U^T Q^{1/2} V = \begin{bmatrix} U_c^T \\ U_{uc}^T \end{bmatrix} Q^{1/2} V, \ with\ U^{-1} = U^T. \tag{30}$$

Similarly, define

$$\begin{bmatrix} e_c \\ e_{uc} \end{bmatrix} \triangleq U^T Q^{1/2} e. \tag{31}$$

where subscript c stands for controllable, subscript uc stands for uncontrollable, U is orthogonal projection matrix that extract the controllable subspace, trajectory tracking error (29) is devided as controllable part and uncontrollable part. The following convergence property of controllable output has been proposed by Amann *et al.* [23].

Theorem 1: Under the condition that $v_k = n_k = 0\ \forall k$, the controllable part of $\bar{e}_k$ from system (7) converges exponentially to origin under the observer-based Q-ILC of (28). Furthermore, $\Delta u_k \to 0$ as $k \to \infty$.

From the derivation in last subsection, the optimal control input has been obtained:

$$u_{k+1} = u_k + H^{Q^*} e_k \tag{32}$$

$$\Delta u_{k+1} = H^{Q^*} e_k \tag{33}$$

The tracking error propagation relation is given by using $e = r - Vu$:

$$e_{k+1} = e_k - V\Delta u_{k+1} \tag{34}$$

$$= e_k - VH^{Q^*} e_k \tag{35}$$

$$= (I - VH^{Q^*})e_k\ \forall k \geq 0 \tag{36}$$

Naturally, the following property is derived:

$$e_k = (I - VH^{Q^*})^k e_0 \tag{37}$$

Similarly, the input propagation equation is derived recursively:

$$u_{k+1} = u_k + H^{Q^*} e_k \tag{38}$$

$$= u_k + H^{Q^*}(r - Vu_k) \tag{39}$$

$$= (I - H^{Q^*} V)u_k + H^{Q^*} r\ \forall k \geq 0 \tag{40}$$

which is similar to Levenberg-Marquardt (LM), the most interesting property regarding the tracking errors is derived as follows:

Converting the control law into another expression and introducing some other symbols to form a more compact representation:

$$u_{k+1} = u_k + {P_k}^{-1} {V_k}^T F e_k \tag{41}$$

where

$$P_k = \begin{bmatrix} G_k^T Q G + H^T W H + R & \\ & G_k^T Q G + H^T W H + R \end{bmatrix},$$

$$F = \begin{bmatrix} Q & \\ & W \end{bmatrix}.$$

Theorem 2: If either $ker({V_k}^T) = 0$ (the null space of ${V_k}^T$) or $r \in rangeV_k$ (the set of vectors formed by the transformation matrix of all the vectors in a space), then the norm of trajectory tracking error e_k will eventually converges to 0.

Proof: It is shown that $\Delta u_{k+1} \to 0$ as $k \to \infty$:

$$0 = \lim_{k\to\infty}(u_{k+1} - u_k) = \lim_{k\to\infty} {P_k}^{-1} {V_k}^T F e_k$$

1) If $ker({V_k}^T) = 0$, then there is no such e to satisfy the relationship of $Ve = 0$, and the tracking error will converge eventually to zero.
2) For the second case, if $ker({V_k}^T) \neq 0$ but $r \in rangeV_k$, considering the monotonicity which implies as follow:

$$\lim_{k\to\infty} \|e_k\|^2 = \lim_{k\to\infty} J_k(u_k) := J_\infty \geq 0 \tag{42}$$

From the above property, the following formula can be inferred [23]:

$$\lim_{k\to\infty} \|u_{k+1} - u_k\|^2 = 0 \tag{43}$$

Considering the improved Q-ILC law given above, the expression of J can be written as (the detailed derivation process is given in Appendix A):

$$J_k = e_k^T F e_k + e_{k-1}^T F V_{k-1} (P_{k-1})^{-1} V_{k-1}^T F e_{k-1} \quad (44)$$
$$= < e_{k-1}, (I + \Phi) e_{k-1} > \quad (45)$$

With the tracking error propagation formulation (37), the above optimization function can be further deduced to:

$$\begin{aligned} J_k &= < (I - VH^{Q^*})^{(k-1)} e_0, \\ &\quad \times (I + \Phi)(I - VH^{Q^*})^{-(k-1)} e_{2k-2} > \\ &= < e_0, (I + \Phi) e_{2k-2} > \end{aligned} \quad (46)$$

Then, the final result is obtained:

$$\begin{aligned} \lim_{k\to\infty} J_k &= \lim_{k\to\infty} (V(u^* - u_0))^T F (I + \Phi) e_{2k-2} \\ &= \lim_{k\to\infty} (V(u^* - u_0))^T F (I + \Phi) H^{Q^*-1} \\ &\quad \times (u_{2k-1} - u_{2k-2}) \end{aligned} \quad (47)$$

□

2) CONVERGENCE PROPERTY OF INPUT SEQUENCE

Theorem 3: The algorithm has the following property:

$$\lim_{k\to\infty} \left\| H^{Q^*} (r - V u_k) \right\| = 0. \quad (48)$$

Proof: Refer to the above inference, the property

$$\lim_{k\to\infty} \| u_{k+1} - u_k \|^2 = 0$$

holds, it extends that:

$$u_{k+1} - u_k = H^{Q^*} e_k \quad (49)$$
$$= H^{Q^*} (r - V u_k) \quad (50)$$

Theorem 3 is proved by substituting equation (50) into above property. □

Theorem 4: If $H^{Q^*} V$ has a inverse with norm $1/\sigma^2$, then the following theorem is established:

$$\| Vu \|^2 \geq \sigma^2 \| u \|^2 \quad (51)$$

then, the convergence property of input is as follows:

$$\| u_{k+1} - u_\infty \| \leq (1 - \sigma^2) \| u_k - u_\infty \|^2 \quad (52)$$

where $u_\infty = (H^{Q^*} V)^{-1} H^{Q^*} r$

The specific proof results can refer to Appendix B.

C. SENSITIVITY TO HIGH-FREQUENCY ERRORS

Sensitivity analysis can be derived from improved Q-ILC learning filter matrix H^{Q^*}.

$$\left\| H^{Q^*} \right\|_\infty = \left\| (G^T Q G + H^T W H + R)^{-1} (G^T Q + H^T W) \right\|_\infty \quad (53)$$

$$\leq \frac{\sigma_{max}(G)\sigma_{max}(Q) + \sigma_{max}(H)\sigma_{max}(W)}{\sigma_{min}(G^T Q G + H^T W H + R)} \quad (54)$$

$$\leq \frac{\sigma_{max}(G)\sigma_{max}(Q) + \sigma_{max}(H)\sigma_{max}(W)}{\sigma_{min}(R)} \quad (55)$$

where σ_{max} and σ_{min} represent maximum and minimum sigular value, respectively, the upper bound of learning filter matrix H^{Q^*} remain constant value when matrices Q, W and R are selected and guaranteed to be non-singular. The upper bound value is independent of sampling period, which indicates that the improved Q-ILC is insensitive to high-frequency errors.

IV. DUAL ESTIMATION

As mentioned in Section III, the observer-based method need to be combined with filtering algorithms. In practice, robot system suffers from various uncertainties, such as the measurement noise, process noise and disturbances. Furthermore, the robot system is highly nonlinear and highly coupling, which complicates the control task. The measurement noise is caused by sensor error and external signal interference, which make the measured robot states inaccurate to describe the actual states. Previously, the extended Kalman filter (EKF) has been widely used to filter the sensor noise [28], [31]. For common nonlinear models, consider the following:

$$x_k = f(x_{k-1}, u_k) + v_k, \quad (56)$$
$$y_k = h(x_k) + n_k. \quad (57)$$

where x_k is the system state, y_k represents the observed noisy signal. u_k denotes the control input, v_k and n_k represent the process noise and measurement noise respectively. The specific prediction and update process of the EKF are described in Algorithm 1.

where

$$F_k = \begin{bmatrix} \frac{\partial f1}{\partial x_1} & \cdots & \frac{\partial f1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial fm}{\partial x_1} & \cdots & \frac{\partial fm}{\partial x_n} \end{bmatrix}$$

$$H_k = \begin{bmatrix} \frac{\partial h1}{\partial x_1} & \cdots & \frac{\partial h1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial hm}{\partial x_1} & \cdots & \frac{\partial hm}{\partial x_n} \end{bmatrix}$$

The above Jacobian matrices are obtained by local linearization, i.e. the first-order approximation term is used as the approximate expression of the original state equation and the measurement equation. Linearization also assumes that the linearized state still obeys the Gaussian distribution, and then, the standard Kalman filter can be used to obtain the optimal state estimation. Using the local linearization technique, the local optimal solution of the problem can be obtained, but whether it can converge to the global optimal solution depends on the nonlinear strength of the system model and the choice of the expansion point. During the update of the Jacobian matrix, the high-order terms of Taylor expansion of the nonlinear function are ignored, only first-order term is used, however, the robotic manipulator is generally a highly nonlinear system. Local linearization will bring a large error,

Algorithm 1 Extended Kalman filter

Input: Measurements z_k and input u_k;
Output: Optimal state estimation $\hat{x}_{k|k}$;
1: Calculate the prediction $\hat{x}_{k|k-1}$ and covariance matrix $P_{k|k-1}$:
2: $\hat{x}_{k|k-1} = f(x_{k-1}, u_k, 0)$
3: $P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$
4: Calculate the residual $\widetilde{y}_k$ between measurement and observation that obtained by linearized measurement model h:
5: $\widetilde{y}_k = z_k - h(\hat{x}_{k|k-1}, 0)$
6: Calculate the auxiliary matrix S_k and Kalman Gain K_k:
7: $S_k = H_k P_{k|k-1} H_k^T + R_k$
8: $K_k = P_{k|k-1} H_k^T S_k^{-1}$
9: Correct nominal state $\hat{x}_{k|k}$:
10: $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \widetilde{y}_k$
11: Correct state covariance $P_{k|k}$:
12: $P_{k|k} = (I - K_k H_k) P_{k|k-1}$

which affects the system state estimation effect. Therefore, only when the system's state equation and observation equation are nearly linear and continuous, the results of EKF filter come close to reality.

In order to improve the filtering effect for nonlinear systems, the UKF algorithm based on unscented transformation was proposed [32]. Unlike EKF, UKF does not approximate the nonlinear model, but approximates the posterior probability density to obtain sub-optimal filtering performance. In this paper, we further consider robots with unknown parameters. Let the system state and the unknown parameters w be concatenated into an extended system state vector X_k, where $X_k = [x_k^T \ w_k^T]^T$, then, the state space in (56) and (57) can be rewritten as:

$$\begin{bmatrix} x_k \\ w_k \end{bmatrix} = \begin{bmatrix} f(x_{k-1}, u_k, w_{k-1}) \\ I \cdot w_{k-1} \end{bmatrix} + \begin{bmatrix} \upsilon_k \\ \upsilon_k \end{bmatrix} \tag{58}$$

$$y_k = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_k \\ w_k \end{bmatrix} + n_k \tag{59}$$

The UKF explicitly extracts the so-called sigma sample points from Gauss, and then transform them through a nonlinear function. Usually, these sigma points are located at the mean value and symmetrically distributed at the covariance of the main axis. For an n-dimensional Gaussian distribution with mean and variance, the result 2n+1 sigma point $\chi^{[i]}$ is selected according to the following rules:

$$\chi^{[0]} = \mu \tag{60}$$

$$\chi^{[i]} = \mu + (\sqrt{(n+\lambda)\Sigma})_i \ i = 1, \cdots, n \tag{61}$$

$$\chi^{[i]} = \mu - (\sqrt{(n+\lambda)\Sigma})_{i-n} \ i = n+1, \cdots, 2n \tag{62}$$

where $\lambda = \alpha^2(n+\kappa) - n$, α and κ are proportional parameters that determine how far the sigma points are distributed over the mean. Each sigma point has two weights associated with it. A weight $\omega_m^{[i]}$ is used for calculating the mean, and another weight $\omega_c^{[i]}$ is used when calculating the covariance of Gaussian:

$$\omega_m^{[0]} = \frac{\lambda}{n+\lambda} \tag{63}$$

$$\omega_c^{[0]} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) \tag{64}$$

$$\omega_m^{[i]} = \omega_c^{[i]} = \frac{1}{2(n+\lambda)} \ i = 1, \cdots, 2n \tag{65}$$

For Gaussian distribution, $\beta = 2$ is the best choice. These sigma vectors are propagated through nonlinear functions $\psi(x)$:

$$y^{[i]} = \psi(\chi^{[i]}) \tag{66}$$

As a result, the mean (μ') and covariance of the Gaussian distribution (Σ') are captured by weighted sample mean and covariance of the sigma points,

$$\mu' = \sum_{i=0}^{2n} \omega_m^{[i]} y^{[i]} \tag{67}$$

$$\Sigma' = \sum_{i=0}^{2n} \omega_c^{[i]} (y^{[i]} - \mu')(y^{[i]} - \mu')^T \tag{68}$$

The specific UKF equation can refer to Algorithm 2, where γ is a constant parameter, R_t and Q_t denote additive measurement noise and additive process noise, respectively.

Algorithm 2 Unscented Kalman filter

Input: Mean μ_{t-1}, variance Σ_{t-1}, measurements z_k and input u_t;
Output: Mean μ_t, variance Σ_t;
1: Choosing sigma points χ_{t-1}:
2: $\chi_{t-1} = (\mu_{t-1} \ \mu_{t-1} + \gamma\sqrt{\Sigma_{t-1}} \ \mu_{t-1} - \gamma\sqrt{\Sigma_{t-1}})$
3: Noiseless predicted state propagation through nonlinear function ψ:
4: $\bar{\chi}_t^* = \psi(u_t, \chi_{t-1})$
5: Calculate weighted mean $\bar{\mu}_t$ and variance $\bar{\Sigma}_t$:
6: $\bar{\mu}_t = \sum_{i=0}^{2n} \omega_m^{[i]} \bar{\chi}_t^{*[i]}$
7: $\bar{\Sigma}_t = \sum_{i=0}^{2n} \omega_c^{[i]} (\bar{\chi}_t^{*[i]} - \bar{\mu}_t)(\bar{\chi}_t^{*[i]} - \bar{\mu}_t)^T + R_t$
8: Get new sigma points $\bar{\chi}_t$:
9: $\bar{\chi}_t = (\bar{\mu}_t \ \bar{\mu}_t + \gamma\sqrt{\bar{\Sigma}_t} \ \bar{\mu}_t - \gamma\sqrt{\bar{\Sigma}_t})$
10: Calculate predicted observations $\bar{Z}_t$ by observation model h:
11: $\bar{Z}_t = h(\bar{\chi}_t)$
12: Get weighted observation $\hat{z}_t$:
13: $\hat{z}_t = \sum_{i=0}^{2n} \omega_m^{[i]} \bar{Z}_t^{[i]}$
14: Calculate uncertainty S_t:
15: $S_t = \sum_{t=0}^{2n} \omega_c^{[i]} (\bar{Z}_t^{[i]} - \hat{z}_t)(\bar{Z}_t^{[i]} - \hat{z}_t)^T + Q_t$
16: Obtain the cross-covariance between state and observation:
17: $\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} \omega_c^{[i]} (\bar{\chi}_t^{[i]} - \bar{\mu}_t)(\bar{Z}_t^{[i]} - \hat{z}_t)^T$
18: Calculate Kalman Gain K_t:
19: $K_t = \bar{\Sigma}_t^{x,z} S_t^{-1}$
20: Update estimation for μ_t and Σ_t:
21: $\mu_t = \bar{\mu} + K_t(z_t - \hat{z}_t)$
22: $\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T$

FIGURE 2. 2 DOF Quanser robot plant [33].

Compare to EKF, UKF is more accurate and its accuracy is equivalent to the second-order Taylor expansion. Another great advantage of UKF is that there is no need to calculate the Jacobian matrix that is sometimes not available. The comparisons of estimation performance between EKF and UKF are given in Section V.

V. SIMULATION

A. THE TESTBED

Fig.2 shows the 2 DOF Quanser robot plant, and the corresponding schematic diagram pictured in Fig.3 shows the robot manipulator. The 2 DOF Robot module is connected to two Rotary Servo Base Units, which are mounted at a fixed distance. Two servomotors on the Rotary Servo Base Units are mounted at a fixed distance and control a 4-bar linkage system: two powered arms coupled through two non-powered arms. The system is planar and has two actuated and three unactuated revolute joints.

The goal of the 2 DOF Robot experiment is to manipulate the X-Y position of a four-bar linkage end effector(joint E). Such a system is similar to the kinematic problems encountered in the control of other parallel mechanisms that have singularities. The actuated joint A and B are control by two Quanser Rotary servo motors respectively. q_1 and q_2 represent the output shaft angle of SRV02 A and SRV02 B.

Let m_i and l_i, $i = 1, 2, 3, 4$ be the mass and length of the bars, and $l_{si} = \frac{l_i}{2}, i = 1, 2, 3, 4$ be the mass center positions of the bars. The dynamics of the plane robot are given in [34]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D(q) = \tau \tag{69}$$

$$M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}; \; C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}; \; D = \begin{bmatrix} d_1 & d_2 \end{bmatrix}^T. \tag{70}$$

$$\begin{aligned}
m_{11} &= m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos q_2(t)) + I_1 + I_2, \\
m_{12} &= m_{21} = m_2(l_{c2}^2 + l_1 l_{c2}\cos q_2(t)) + l_2, \\
m_{22} &= m_2 l_{c2}^2 + I_2, \\
c_{11} &= h\dot{q}_2(t), \\
c_{12} &= h\dot{q}_1(t) + h\dot{q}_2(t), \\
c_{21} &= -h\dot{q}_1(t), c_{22} = 0, \\
h &= -m_2 l_1 l_{c2}\sin q_2(t), \\
d_1 &= (m_1 l_{c1} + m_2 l_1)g\cos q_1(t) + m_2 l_{c2} g\cos(q_1(t) + q_2(t)), \\
d_2 &= m_2 l_{c2} g\cos(q_1(t) + q_2(t)).
\end{aligned}$$

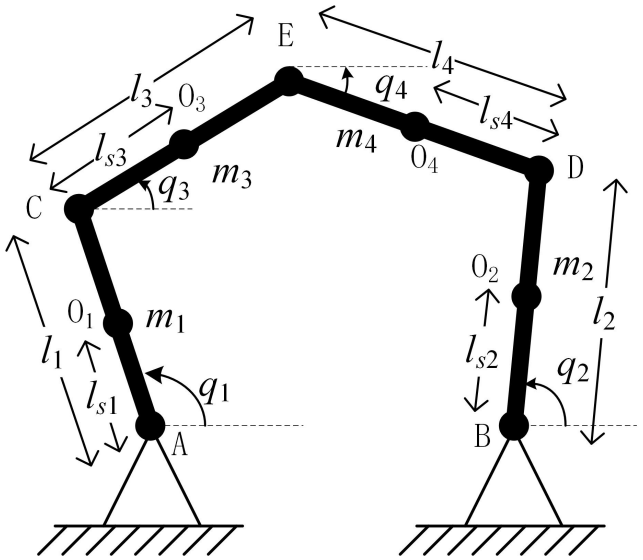


FIGURE 3. Schematic diagram of the 2DOF robot manipulator.

TABLE 1. Parameter list.

Symbol	Specification	value	units
m_1	mass	10	kg
m_2	mass	5	kg
1_1	length	1	m
l_2	length	0.5	m
l_{s1}	length	0.5	m
l_{s2}	length	0.25	m
I_1	Inertia constant	0.83	
I_2	Inertia constant	0.3	
g	Gravity constant	9.81	N/kg

where q denotes the joint angle position, defined by $q = [q_1\ q_2]$, q_1 and q_2 for the $1st$ and $2nd$ joint, respectively. $M(q) \in R^{n\times n}$ is the inertia matrix, $C(q, \dot{q}) \in R^{n\times n}$ denotes the Centrifugal and Coriolis forces, and $D(q) \in R^n$ represents the gravitational force, $\tau \in R^n$ is the vector of torques which operates at the joints.

In simulation, the robot system parameters are listed in Table.1.

B. SIMULATION 1: COMPARISON OF ESTIMATION PERFORMANCE

In this simulation, the estimation performance between EKF and UKF are compared. The input torque is set to a fixed value:

$$\tau = [1, 1]^T,$$

the covariance of measurement noise is selected as:

$$R = diag([1, 1, 4, 4])^2$$

and the unknown parameters I_1 and I_2 are all set to a constant initial value:

$$[I_1, I_2] = [1, 1].$$

The results of parameter estimation by UKF and EKF are shown in Fig.4. Fig.5 is a partial enlarged view of Fig.4, it shows that the state estimation result of UKF are very close

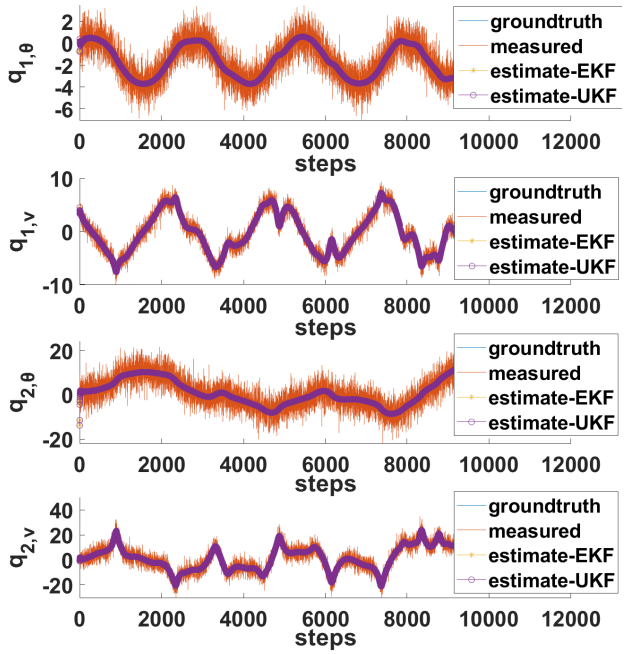


FIGURE 4. Comparison of state estimation performance between UKF and EKF.

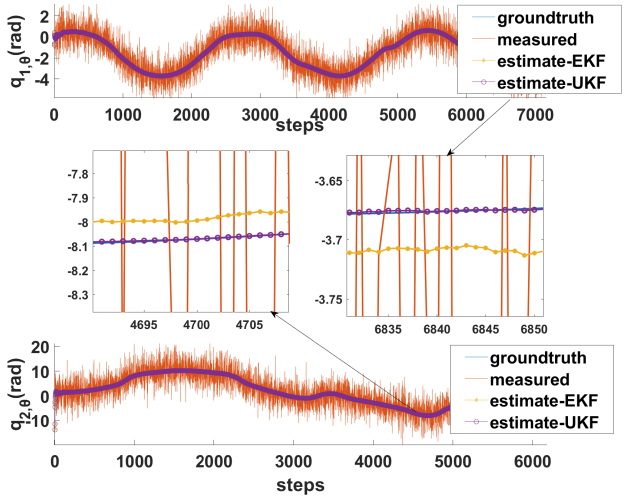


FIGURE 5. The partial enlarged view of Fig.4 (The line indicating the UKF estimation result almost coincides with the line indicating the actual state).

to the actual state, this is mainly because the application of EKF in such a highly nonlinear system will bring a large linearization error. Similarly, form Fig.6, it shows that UKF achieved more stable and accurate parameter estimation performance. All these experiments illustrate the applicability of UKF in robot state estimation compared to EKF.

C. SIMULATION 2: COMPARISON OF CONVERGENCE

To control the joint E to have desired shape movement, this can be done by giving a reference signal at each channel (Corresponding to q_1 and q_2 respectively), here select the sinusoidal signal. In the absence of measurement noise,

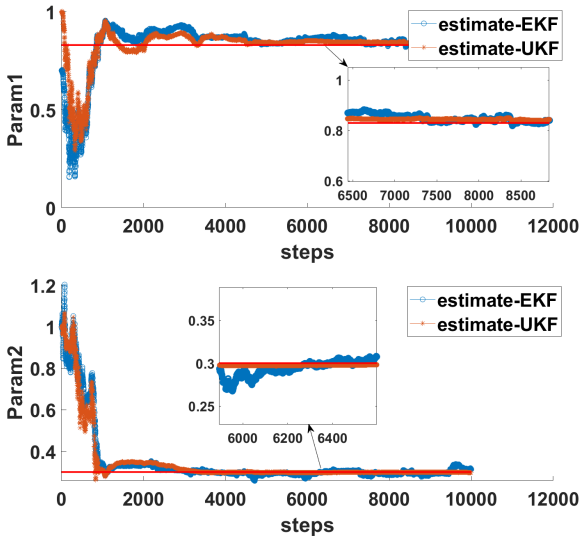


FIGURE 6. Estimation of uncertain parameters of EKF and UKF.

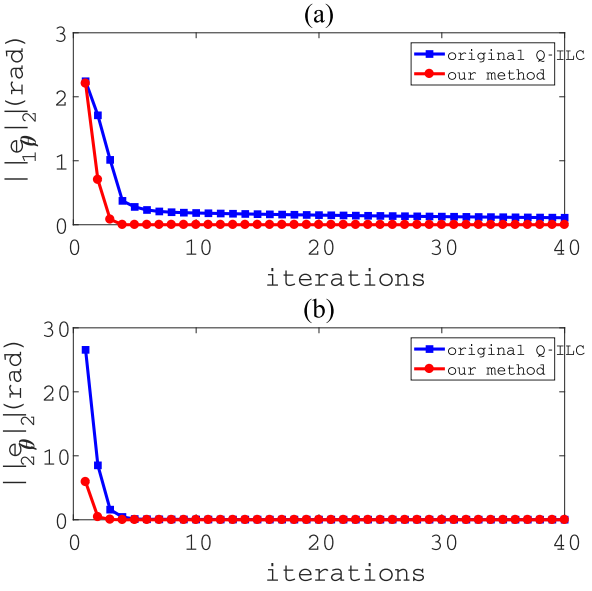


FIGURE 7. The angle position tracking error of 2DOF planar robot.

the tracking performance of original Q-ILC and the improved Q-ILC issued in this paper are shown in Fig.7. The $e_{1,\theta}$ and $e_{2,\theta}$ represent two joint angle position error separately (the error among q_1, q_2 and their corresponding expected value). From Fig.7, it indicates that our improved Q-ILC algorithm achieves faster convergence and better transient performance compared to original Q-ILC, also, the results indicate that with the same convergence criteria our algorithm provides better final results. Thus for same quality of convergence it can be used with less strict convergence criteria.

The method used in this paper requires the updates of sensitivities G_k and H_k (iterative linearization). Generally, linearization using finite difference is time consuming that perhaps influence the robot real-time control. To this end, we apply Automatic Differentiation (AD) that improves the computational efficiency immensely.

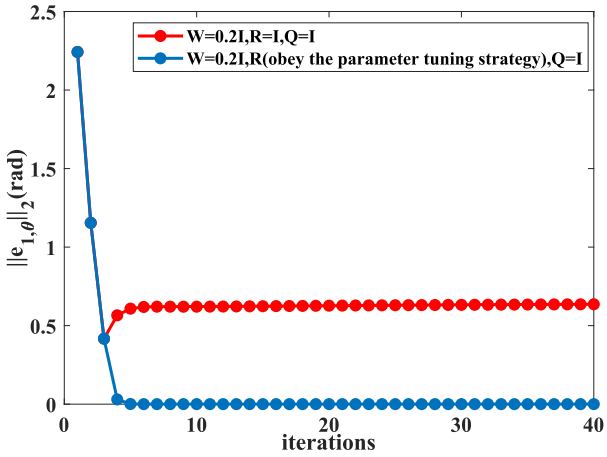


FIGURE 8. Comparison experiment of weight matrix R.

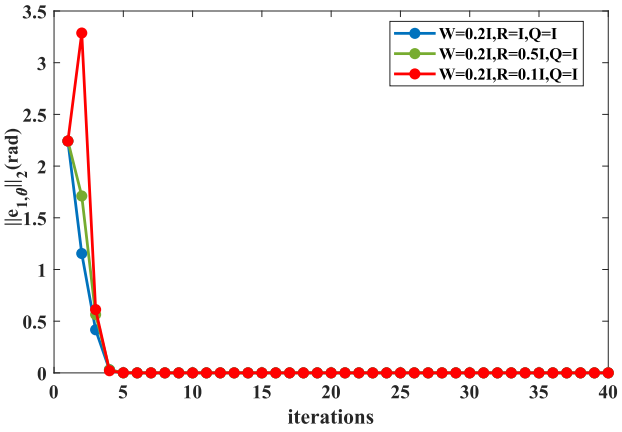


FIGURE 9. The influence of weight matrice R on error convergence.

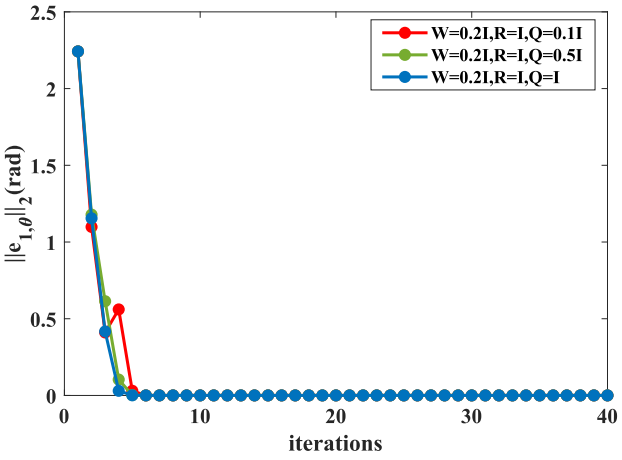


FIGURE 10. The influence of weight matrice Q on error convergence.

D. SIMULATION 3: INFLUENCE OF PARAMETERS ON CONTROL PERFORMANCE

In this subsection, influences of weight matrices on control performance will be disscussed. Since Q and W have similar effects, only the effect of Q is considered here. Furthermore, for simplicity, only the convergence result for the first joint is analyzed.

Firstly, consider the two following parameter setting:

1) W=0.2I, R=I, Q=I;
2) W=0.2I, R is adjusted according to the parameter tuning strategy (as described later), Q=I.

The experimental results are shown in Fig.8, it can be found that for the first case the tracking errors cannot be further reduced after 5*th* iteration, this is mainly because when the tracking error converges to the neighborhood of the origin, the control input will not change significantly. In this case, a large weight for the control input increment discourages the movement of inputs for the next batch, thus reducing the trajectory tracking accuracy. In the second experiment, a simple heuristic parameter selection strategy was considered for R: when the trajectory tracking error was lower than a certain threshold, R was adjusted to 0.01I. The result in Fig.8 proves the effectiveness of parameter adjustment strategy.

In the next, we further analyze the general influence of R on the control performance. To be straightforward for demonstration, the following parameters are selected for R:

1) R=0.1I;
2) R=0.5I;
3) R=I.

The results are given in Fig.9. Bascially, faster convergence can be achieved when choose for a larger R. The results confirm the usefulness of a switching strategy for R. That is, a large R should be used in the early stage to speed up the convergence and then a small R to improve the tracking accuracy.

In order to analyze the influence of Q, the parameters are selected as follows(in this case, the switching strategy for R is followed):

1) W=0.2I, R=I, Q=0.1I;
2) W=0.2I, R=I, Q=0.5I;
3) W=0.2I, R=I, Q=I.

The results are shown in Fig.10. We observe that with the increase of Q, the convergence speed is mildly accelerated, however, this phenomenon is not significant, because the weight W has a major impact on the speed error, according to the theoretical analysis. In practice, Q can be selected as $0.5I \sim I$.

E. SIMULATION 4: CONVERGENCE AND ROBUSTNESS

In this simulation, convergence and robustness in the case of noise and uncertainty parameters are studied. The Gaussian white noise is added to the measured data, and the uncertain parameters are selected as I_1 and I_2, the initial values of these uncertain parameters are set to 0.1 and 1 respectively.

The robot state estimation and uncertain parameter estimation results are given in Fig.11 and Fig.12, respectively. From these figures, we can see that the dual estimation of UKF

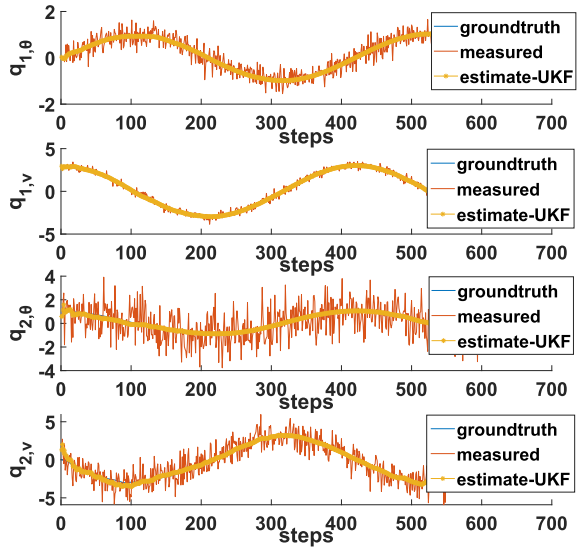


FIGURE 11. The estimated system states by UKF in one trial.

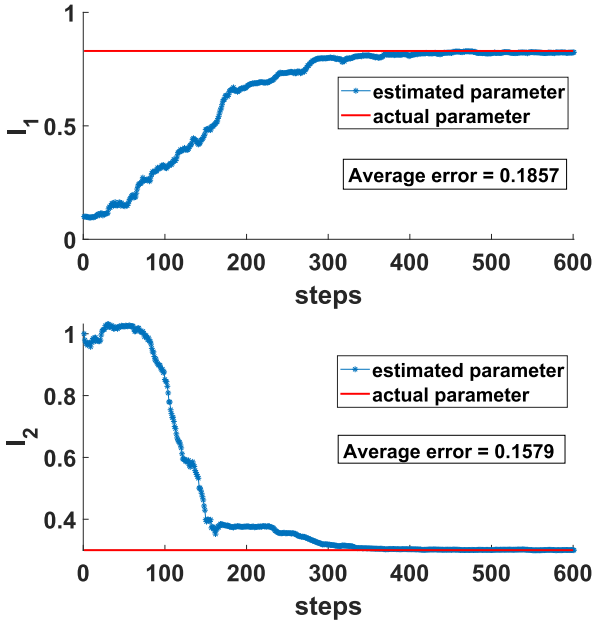


FIGURE 12. The estimated uncertain parameters by UKF.

enables to capture the actual states and model parameters, which benefits a lot to the convergence rate of ILC. The result of "improved Q-ILC+UKF" algorithm in the presence of measurement noise and uncertain parameters is shown in Fig.13, fluctuations of the error convergence curve can be found, this is mainly due to the inaccurate estimation by UKF in the first few batches. On the other hand, the selection strategy for R can reduce such fluctuation, as can be seen, the final error curve is gradually stable. Although the transient performance is influenced by measurement noise and model errors, the combination of improved Q-ILC and UKF achieves a satisfactory performance, the errors of q_1 and q_2 converge to zero at approximately the 10-th and 5-th iterations, respectively.

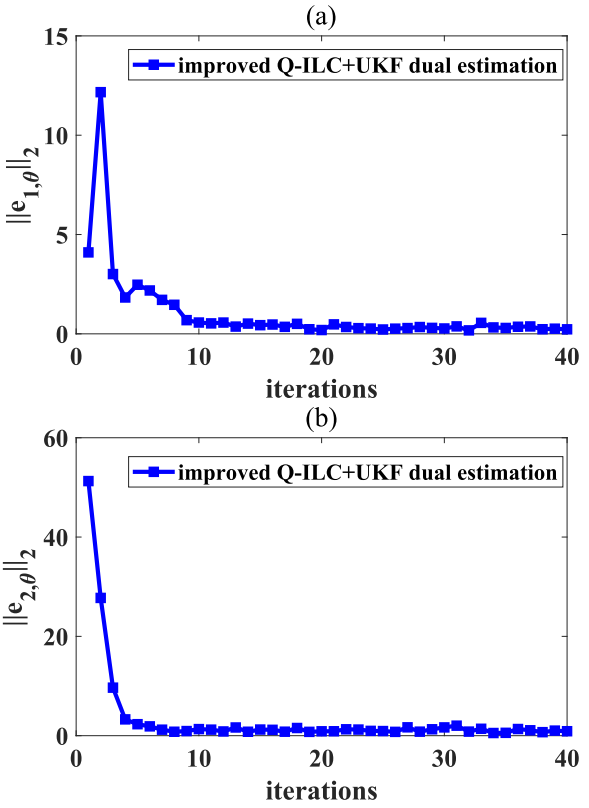


FIGURE 13. The tracking error norm of 2DOF planar robot in the presence of measurement noise and uncertain parameters.

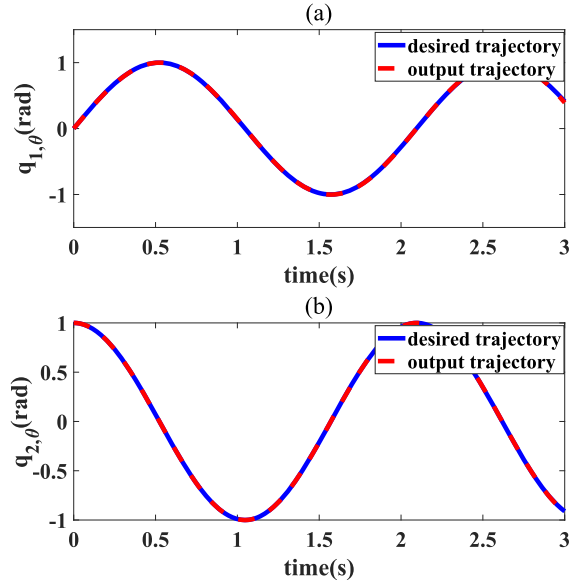


FIGURE 14. The output trajectory of 15-th iteration when measurement noise added.

Finally, the trajectory tracking performance is given in Fig.14, which show that the output trajectory converges to the given trajectory at 15-th iteration. The above simulation results and theoretical analysis in this paper show that compared with the original Q-ILC algorithm, the combination of improved Q-ILC and UKF's online dual estimation achieves faster error convergence.

VI. CONCLUSION

This paper proposed an improved Q-ILC algorithm for trajectory tracking of uncertain robot manipulators, the velocity information is considered compared to the original Q-ILC. The closed-form solution of the new QILC method is derived, together with convergence and robust analysis. Basically, the trajectory tracking error of the previous batch is more fruitfully used as the feedforward signal of the Q-ILC algorithm, which is helpful for improving the control performance. In this work, we presented a dual estimation strategy using the UKF algorithm, where the unknown parameters and systems states are simultaneously estimated. In the simulation study of a 2DOF plane robot, the output trajectory converged to the desired trajectory within a few iterations, the combination of the improved Q-ILC and UKF achieves faster error convergence and is more robust to noise and model errors. The improved Q-ILC algorithm is suitable for manipulator systems that demand fast tracking for high production efficiency. However, since industrial robots also suffer from real-time disturbances that cannot be handled by iterative learning, how to incorporate feedback control into the framework of this paper for robot control is important. Moreover, iterative learning control under completely unknown dynamics is also a subject that will be studied in the future.

APPENDIXES

APPENDIX A

$$J_k = e_k^T F e_k + e_{k-1}^T F V_{k-1} P_{k-1}^{-1} V_{k-1}^T F e_{k-1} \tag{71}$$

Using the tracking error propagation (34), J_k can be converted to:

$$\begin{aligned} J_k &= e_k^T F e_k + e_{k-1}^T F V_{k-1} P_{k-1}^{-1} V_{k-1}^T F e_{k-1} \\ &= e_{k-1}^T (I - V_{k-1} P_{k-1}^{-1} V_{k-1}^T F)^T F (I - V_{k-1} P_{k-1}^{-1} V_{k-1}^T F) \\ &\quad \times e_{k-1} + e_{k-1}^T F V_{k-1} P_{k-1}^{-1} V_{k-1}^T F e_{k-1} \\ &= e_{k-1}^T F e_{k-1} + e_{k-1}^T F V_{k-1} P_{k-1}^{-1} V_{k-1}^T F V_{k-1} P_{k-1}^{-1} V_{k-1}^T \\ &\quad \times F e_{k-1} + e_{k-1}^T F V_{k-1} P_{k-1}^{-1} V_{k-1}^T F e_{k-1} \\ &= e_{k-1}^T F e_{k-1} + e_{k-1}^T F (V_{k-1} H_{k-1}^{Q^*})^2 e_{k-1} \\ &\quad + e_{k-1}^T F V_{k-1} H_{k-1}^{Q^*} e_{k-1} \\ &= < e_{k-1}, (I + \Phi) e_{k-1} > \end{aligned} \tag{72}$$

where

$$\Phi = V_{k-1} H_{k-1}^{Q^*} + (V_{k-1} H_{k-1}^{Q^*})^2 \tag{73}$$

APPENDIX B

If $H^{Q^*} V$ has a inverse, then, the formulation

$$(H^{Q^*} V)^{-1} (u_{k+1} - u_k)$$

will also converge to zero as $k \to \infty$, it implies that

$$\begin{aligned} &\lim_{k \to \infty} (H^{Q^*} V)^{-1} (u_{k+1} - u_k) \\ &\quad = \lim_{k \to \infty} (H^{Q^*} V)^{-1} [H^{Q^*} (r - V u_k)] \\ &\quad = (H^{Q^*} V)^{-1} H^{Q^*} r - u_k \end{aligned} \tag{74}$$

so, the limit of u_k is $u_\infty = (H^{Q^*} V)^{-1} H^{Q^*} r$, based on the above derivation, it follows that:

$$\begin{aligned} u_{k+1} - u_\infty &= (I - H^{Q^*} V) u_k + H^{Q^*} r - u_\infty \\ &= (I - H^{Q^*} V) u_k + H^{Q^*} V u_\infty - u_\infty \\ &= (I - H^{Q^*} V)(u_k - u_\infty) \end{aligned} \tag{75}$$

According to the derivation given in [23], the following inequality holds:

$$\|u_{k+1} - u_\infty\| \leq (1 - \sigma^2) \|u_k - u_\infty\|^2 \tag{76}$$

REFERENCES

[1] S. Mobayen, F. Tchier, and L. Ragoub, "Design of an adaptive tracker for n-link rigid robotic manipulators based on super-twisting global nonlinear sliding mode control," *Int. J. Syst. Sci.*, vol. 48, no. 9, pp. 1990–2002, Mar. 2017.

[2] S. Amirkhani, S. Mobayen, N. Iliaee, O. Boubaker, and S. H. Hosseinnia, "Fast terminal sliding mode tracking control of nonlinear uncertain mass–spring system with experimental verifications," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 1, Feb. 2019, Art. no. 172988141982817.

[3] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. Robotic Syst.*, vol. 1, no. 2, pp. 123–140, 1984.

[4] J. H. Lee, K. S. Lee, and W. C. Kim, "Model-based iterative learning control with a quadratic criterion for time-varying linear systems," *Automatica*, vol. 36, no. 5, pp. 641–657, May 2000.

[5] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.

[6] K. L. Moore, Y. Chen, and H.-S. Ahn, "Iterative learning control: A tutorial and big picture view," in *Proc. 45th IEEE Conf. Decis. Control*, Dec. 2006, pp. 2352–2357.

[7] C.-J. Chien, "A combined adaptive law for fuzzy iterative learning control of nonlinear systems with varying control tasks," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 1, pp. 40–51, Feb. 2008.

[8] L. P. Zhang and F. Yang, "Fuzzy iterative learning control design for output tracking of discrete-time fuzzy systems," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Aug. 2004, pp. 678–682.

[9] J. Shi, F. Gao, and T.-J. Wu, "Single-cycle and multi-cycle generalized 2D model predictive iterative learning control (2D-GPILC) schemes for batch processes," *J. Process Control*, vol. 17, no. 9, pp. 715–727, Oct. 2007.

[10] W. Chen and Y.-Q. Chen, "Robust iterative learning control for output tracking via second-order sliding mode technique," in *Proc. Amer. Control Conf.*, Jun. 2010, pp. 2051–2056.

[11] C. Yin, J.-X. Xu, and Z. Hou, "A high-order internal model based iterative learning control scheme for nonlinear systems with Time-Iteration-Varying parameters," *IEEE Trans. Autom. Control*, vol. 55, no. 11, pp. 2665–2670, Nov. 2010.

[12] S. Zamiri, A. Madadi, and H.-R. Reza-Alikhani, "Adaptive PID type iterative learning control," *Int. J. Electr. Comput. Eng.*, vol. 4, no. 6, pp. 37–42, Dec. 2014.

[13] A. Tayebi, "Adaptive iterative learning control for robot manipulators," *Automatica*, vol. 40, no. 7, pp. 1195–1203, Jul. 2004.

[14] M. Norrlof, "An adaptive iterative learning control algorithm with experiments on an industrial robot," *IEEE Trans. Robot. Autom.*, vol. 18, no. 2, pp. 245–251, Apr. 2002.

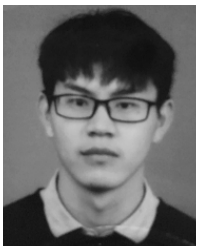
[15] C. Wang, M. Zheng, Z. Wang, and M. Tomizuka, "Robust two-degree-of-freedom iterative learning control for flexibility compensation of industrial robot manipulators," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 2381–2386.

[16] W. Chen and M. Tomizuka, "Dual-stage iterative learning control for MIMO mismatched system with application to robots with joint elasticity," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 4, pp. 1350–1361, Jul. 2014.

[17] P. C. Marchal, O. Sörnmo, B. Olofsson, A. Robertsson, J. G. Ortega, and R. Johansson, "Iterative learning control for machining with industrial robots," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 9327–9333, 2014.

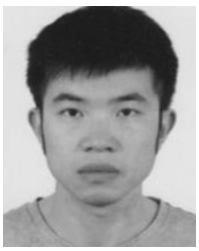
[18] B. Bukkems, D. Kostic, B. de Jager, and M. Steinbuch, "Learning-based identification and iterative learning control of direct-drive robots," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 537–549, Jul. 2005.

[19] K. Hamamoto and T. Sugie, "Iterative learning control for robot manipulators using the finite dimensional input subspace," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 632–635, Aug. 2002.
[20] X. Li, L. Xue, and C. Sun, "Linear quadratic tracking control of unknown discrete-time systems using value iteration algorithm," *Neurocomputing*, vol. 314, pp. 86–93, Nov. 2018.
[21] Q. Yan, J. Cai, L. Wu, and Q. Zhou, "Error-tracking iterative learning control for nonlinearly parametric time-delay systems with initial state errors," *IEEE Access*, vol. 6, pp. 12167–12174, 2018.
[22] G. Honderd, "Iterative learning control for deterministic systems," *Automatica*, vol. 32, no. 6, pp. 948–949, Jun. 1996.
[23] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control for discrete-time systems with exponential rate of convergence," *IEE Proc.-Control Theory Appl.*, vol. 143, no. 2, pp. 217–224, Mar. 1996.
[24] H. Kim, J. H. Park, and K. S. Lee, "Methods and properties of quadratic iterative learning control for semi-conductor processes under different perturbations," in *Proc. 13th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2013, pp. 570–573.
[25] F. L. Mueller, A. P. Schoellig, and R. D'Andrea, "Iterative learning of feedforward corrections for high-performance tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 3276–3281.
[26] A. P. Schoellig, F. L. Mueller, and R. D'Andrea, "Optimization-based iterative learning for precise quadrocopter trajectory tracking," *Auto. Robots*, vol. 33, nos. 1–2, pp. 103–127, Apr. 2012.
[27] A. Schollig and R. D'Andrea, "Optimization-based iterative learning control for trajectory tracking," in *Proc. Eur. Control Conf. (ECC)*, Aug. 2009, pp. 1505–1510.
[28] N. Degen and A. P. Schoellig, "Design of norm-optimal iterative learning controllers: The effect of an iteration-domain Kalman filter for disturbance estimation," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 3590–3596.
[29] S. Ben Warrad, O. Boubaker, M. Lungu, and S. Mobayen, "Full and reduced-order unknown input observer design for linear time-delay systems with multiple delays," *Math. Problems Eng.*, vol. 2018, pp. 1–13, Nov. 2018.
[30] R. Chi, Z. Hou, and J. Xu, "Adaptive ILC for a class of discrete-time systems with iteration-varying trajectory and random initial condition," *Automatica*, vol. 44, no. 8, pp. 2207–2213, Aug. 2008.
[31] J. Wallen, S. Gunnarsson, R. Henriksson, S. Moberg, and M. Norrlof, "ILC applied to a flexible two-link robot model using sensor-fusion-based estimates," in *Proc. 48th IEEE Conf. Decis. Control (CDC) Held Jointly 28th Chin. Control Conf.*, Dec. 2009, pp. 458–463.
[32] E. A. Wan and R. V. D. Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adapt. Syst. Signal Process., Commun., Control Symp.*, Oct. 2000, pp. 153–158.
[33] *2 DOF Robot*. Accessed: Apr. 18, 2017. [Online]. Available: https://www.quanser.com/products/2-dof-robot/
[34] C. C. de Wit, B. Siciliano, and G. Bastin, *Theory of Robot Control*. London, U.K.: Springer, 2012.



MINFENG ZHU received the B.S. degree in electrical engineering and automation from Ningbo University, Ningbo, China, in 2017. He is currently pursuing the M.S. degree in control science and engineering with Zhejiang University, Hangzhou, China.

His research interests include iterative learning control, simultaneous localization and mapping, and deep learning.



LINGJIAN YE received the B.Eng. degree in biochemical engineering and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2006 and 2011, respectively.

Since 2011, he has been with the School of Information Science and Engineering, Ningbo Institute of Technology, Zhejiang University, Ningbo, China, where he is currently a Professor. His research interests include control structure design, plant-wide control, real-time optimization, and their applications in process systems.

XIUSHUI MA received the B.S. degree in industrial electrical automation from the Anhui College of Mechanical and Electrical Engineering, Wuhu, China, in 1985, and the M.S. degree in automation engineering and the Ph.D. degree in precision machinery and instruments from the Hefei University of Technology, Hefei, China, in 1995 and 2002, respectively.

Since 2011, he has been with the School of Information Science and Engineering, Ningbo Institute of Technology, Zhejiang University, Ningbo, China, where he is currently a Professor. His research interests include detection technology and automation equipment, and precision instruments and machinery.