RECURSIVE IDENTIFICATION OF PHYSICAL PARAMETERS IN A FLEXIBLE ROBOT ARM

Måns Östring and Svante Gunnarsson

ABSTRACT

Recursive identification of a physically parameterized model of a single link flexible robot arm is considered. The parameters are estimated using a recursive prediction error method applied to a linear continuous time model structure and discrete time data. The algorithm is applied to real data from an industrial robot, and three important parameters are identified using only measurements of the motor angle. The experiments show that the particular parameter that is estimated will have big influence on the algorithm behavior.

KeyWords: Recursive identification, robotics, physical parameters.

I. INTRODUCTION

This paper deals with recursive identification of the parameters in a physically parameterized model of a flexible robot arm. Recent examples of off-line robot identification in general are given in, e.g., [1], [2], and [3]. Off-line identification of the parameters in the type of physically parameterized robot models studied here is presented in [4].

Off-line identification of physical parameters in continuous time models can be carried out using commercially available software. See, for example, [5]. Recursive identification of parameters in discrete time models of black-box type is also an established area [6]. The topic of this paper, i.e., to recursively estimate the physical parameters in continuous time models, has been considered in the nonlinear case in e.g., [7]. The topic is important in cases like fault isolation and fault identification where the main task is to identify the continuous time parameter values as fast as possible after a fault has occurred. It is in this context the recursive identification of physical parameter values comes in. Examples of the use of the recursively identified parameters for diagnosis purposes are given in [8].

II. THE RECURSIVE IDENTIFICATION ALGORITHM

2.1 Algorithm structure

The identification will be carried out using a RPEM (recursive prediction error) algorithm, see e.g., [6], and the algorithm is studied before presenting the application. One of the fundamental components in system identification is the model structure, and here the focus will be on physically parameterized linear models. This means that the model structure is given by the state space equations

\[ x(t) = A(\theta) x(t) + B(\theta) u(t) \], \[ y(t) = C(\theta) x(t) \], \hfill (1) \]

where the vector \( \theta \) contains the parameters to be identified. Furthermore \( u(t) \) and \( y(t) \) denote the input and output signals, and \( x(t) \) denotes the \( n \)-dimensional state vector. The second fundamental component in identifi-
cation is the set of data collected from the real system. It will here be assumed that the data are collected in real time and that the identification is carried out recursively. The aim is to find the parameter vector \( \theta \) such that the model (1) describes the data as accurately as possible. This is formulated as the problem of recursively minimizing the criterion

\[
V_k(\theta) = \sum_{i=1}^{k} \varepsilon^2(l, \theta), \tag{2}
\]

where \( \varepsilon(l, \theta) \) denotes the prediction error, i.e., difference between the measured output signal and the predicted output signal,

\[
\varepsilon(l, \theta) = y(l) - \hat{y}(l, \theta). \tag{3}
\]

In Eq. (2) and in the sequel the variable \( k \) will be used to denote the discrete time sampling points. In Eq. (3) the variable \( \hat{y}(l, \theta) \) denotes the prediction of the system output at time \( l \) given the parameter vector \( \theta \). A complete derivation of the RPEM algorithm can be found in [6], and the resulting algorithm is given by

\[
\hat{\theta}(k) = \hat{\theta}(k-1) + P(k) \psi(k) \varepsilon(k), \tag{4}
\]

where \( \varepsilon(k) = y(k) - \hat{y}(k, \hat{\theta}(k-1)) \), \( P(k) \) is a symmetric (covariance) matrix, and \( \psi(k) \) denotes the gradient of the prediction error with respect to the unknown parameters. In order to apply the algorithm (4) one has to choose some appropriate method for updating the matrix \( P(k) \), and this issue will be discussed in Section 2.2. Furthermore the prediction \( \hat{y}(k, \hat{\theta}(k-1)) \) and its gradient \( \psi(k) \) have to be computed using the model structure, and this topic is treated in Section 2.3.

2.2 Covariance matrix update equations

When designing an update algorithm for the matrix \( P(k) \) there are two main issues that have to be taken into account. First, when identifying parameters that are subject to changes it is important that the algorithm maintains its tracking ability. Second, it is important to prevent \( P(k) \) from growing during periods of poor excitation. These issues can be dealt with in different ways and the aim here is to point out some possible solutions. The two algorithms that will be studied here are given by

\[
\overline{P}(k) = \frac{1}{\lambda} \left( P(k-1) - \frac{P(k-1)\psi(k)\psi^T(k)P(k-1)}{1 + \psi^T(k)P(k-1)\psi(k)} \right), \tag{5}
\]

and

\[
\overline{P}(k) = P(k-1) - \frac{P(k-1)\psi(k)\psi^T(k)P(k-1)}{1 + \psi^T(k)P(k-1)\psi(k)} + \Delta, \tag{6}
\]

respectively. The first is the well known forgetting factor algorithm, where \( 0 < \lambda \leq 1 \) is the so called forgetting factor. The second algorithm is the covariance matrix modification algorithm, and it is inspired by a Kalman filter interpretation of the identification problem. In (6), \( \Delta \) is a symmetric positive definite matrix. Choosing the design variables \( \lambda \) or \( \Delta \) appropriately ensures that the algorithm maintains its tracking ability.

The second issue, i.e., to prevent \( P(k) \) from growing due to poor excitation, can be handled by applying regularization of the information matrix, i.e., the inverse of the covariance matrix. Using the matrix inversion lemma this corresponds to

\[
P(k) = \overline{P}(k)(I + \mu \overline{P}(k))^{-1}, \tag{7}
\]

where \( \mu \) is a positive scalar. For details see [9]. The regularization of the information matrix hence corresponds to a normalization of the covariance matrix. A straightforward use of the normalization defined by Eq. (7) would be computationally demanding. In [9] it is discussed how some of the proposed method for handling the covariance matrix can be seen as approximations of Eq. (7).

The design parameters of the RPEM algorithm are hence \( \lambda \) and \( \Delta \) for the tracking properties and \( \mu \) for the regularization. Different aspects of the choice of these parameters will be discussed below.

2.3 Forming the prediction and its gradient

A key point when applying the RPEM algorithms above is how to determine \( \hat{y}(k, \theta) \) and \( \psi(k) \) for the continuous time model when only discrete time data are available. There are three main alternatives that can be considered. One alternative is to convert the continuous time state space model in Eq. (22) to discrete time form using standard methods assuming zero order hold of the input. The resulting matrices of the discrete time state space model are however complicated functions of the physical parameters, and this makes the differentiation complicated. A second approach is to apply numerical differentiation to the prediction \( \hat{y}(k) \) in order to obtain an estimate of \( \psi(k) \). This approach has been tested in this application, but the result was not satisfactory. The third alternative, which is the approach taken here, is to do the differentiation using the continuous time model and in a second step convert the expression for the gradient to discrete time. Consider therefore a linear state space model, Eq. (1), for which the predictor is
\[ \dot{x}(t) = A(\theta)x(t) + B(\theta)u(t) \quad \dot{y}(t, \theta) = C(\theta)x(t), \quad (8) \]

This means that only the input-output dynamics are estimated, which corresponds to an output error (OE) model structure. In general it is possible to also estimate a disturbance model, but off-line identification has shown, see [4], that the OE-structure gives better results for this application. The equations for the gradient of the prediction with respect to a scalar parameter can be found in, e.g., [10], and they are given by

\[ \psi(t, \theta) = \frac{d}{d\theta} \tilde{y}(t, \theta) = C(\theta)z(t, \theta) + \overline{C}x(t, \theta), \quad (9) \]

where

\[ z(t, \theta) = \frac{d}{dt} x(t, \theta), \quad (10) \]

and

\[ \overline{C}(\theta) = \frac{d}{d\theta} C(\theta). \quad (11) \]

Using (8) and (10) the time derivative of \( z(t) \) is obtained from

\[
\begin{align*}
\frac{d}{dt} z(t, \theta) &= \frac{d}{dt} \frac{d}{d\theta} x(t, \theta) = \frac{d}{d\theta} \frac{d}{dt} x(t, \theta) \\
&= \frac{d}{d\theta} \dot{x}(t) = A(\theta)z(t) + \overline{A}x(t) + \overline{B}u(t),
\end{align*}
\]

where

\[ \overline{A}(\theta) = \frac{d}{d\theta} A(\theta) \quad \overline{B}(\theta) = \frac{d}{d\theta} B(\theta). \quad (12) \]

With the extended state vector

\[ X(t) = \begin{bmatrix} z(t) \\ x(t) \end{bmatrix}, \quad (14) \]

the state space system for generating the prediction and the gradient is given by

\[ \dot{X}(t) = \begin{bmatrix} A(\theta) & \overline{A}(\theta) \\ 0 & A(\theta) \end{bmatrix} X(t) + \begin{bmatrix} \overline{B}(\theta) \\ B(\theta) \end{bmatrix} u(t), \quad (15) \]

\[ \begin{bmatrix} \psi(t) \\ \hat{y}(t) \end{bmatrix} = \begin{bmatrix} C(\theta) & \overline{C}(\theta) \\ 0 & C(\theta) \end{bmatrix} X(t). \quad (16) \]

Equations (11) and (13) are repeated for each parameter that should be identified. This means that for each parameter that is identified the state space model in Eq. (15) is extended with \( n \) more states, where \( n \) is the size of \( x(t) \). Depending on which parameter that is of interest the matrices \( \overline{A} \), \( \overline{B} \) and \( \overline{C} \) will have different properties. Typically the physical parameter \( \theta \) is entering the system matrices only in a few places and hence the matrices \( \overline{A} \), \( \overline{B} \) and \( \overline{C} \) will contain a lot of zeros. In order to generate \( \hat{y}(k) \) and \( \psi(k) \) using discrete time data the state space model in (15) is transformed to its discrete time counterpart in each sampling point, using the current parameter estimate. Before the prediction and the gradient can be computed it is also necessary to check that the system (15) is asymptotically stable when evaluated using the current parameter estimate. It is beyond the scope of this paper to prove stability and convergence of the RPEM algorithm when applied to the physically parameterized model. There are however some actions that can be taken in order to ensure a proper behavior of the algorithm. The stability check of the predictor mentioned above is one such action. Another action is to use some kind of regularization, as discussed in Section 2.2, to handle cases when the excitation is poor. In this particular application the boundedness of the input and output signals is always ensured since the robot is operating in closed loop. It should be noted that since the problem is nonlinear in the parameters the problem of choosing good initial values still remains.

2.4 Some properties of the gradient

An important property when identifying physical parameters is that the character of the gradient \( \psi(k) \) depends on which particular parameter that is identified. Consider the transfer operator of the model converted to discrete time

\[ y(k) = G(q, \theta)u(k) + e(k), \quad (17) \]

where \( e(k) \) is white noise, since, as mentioned above, an output error structure is considered. The prediction is hence given by

\[ \hat{y}(k) = G(q, \theta)u(k). \quad (18) \]

The gradient of the prediction with respect to a scalar \( \theta \) can be expressed as

\[ \psi(k) = G_q(q, \theta)u(k), \quad (19) \]

where \( G_q(q, \theta) \) is the derivative of the transfer operator \( G(q, \theta) \) with respect to \( \theta \). The variance of \( \psi(k) \) will have big a influence on the properties of the estimates, and it can be expressed
The magnitude of the variance hence depends on the character of the input spectrum and the properties of the transfer function $G_\theta$. This will be illustrated in a particular example below.

### III. THE ROBOT SYSTEM

The robot that is studied in this paper is an industrial robot of the type ABB IRB 1400, and it is shown in Fig. 1.

![Fig. 1. ABB IRB 1400.](image)

The dynamics of the robot system when moving around axis one will be approximated by a model consisting of three masses connected via springs and dampers as shown in Fig. 2. In [11] models with two and three masses were compared, and it was found that models with three masses gave considerably better results. Therefore this paper is restricted to three-mass models. The input is the torque $\tau$ generated by the electrical motor, while the output is the motor angle $\phi_m$. The angles of the other masses, $\phi_g$ and $\phi_a$ respectively, are not measurable. The notations are explained in Table 1. Even though this is a considerably simplified description of a real robot arm it is a useful approximation in many situations.

![Fig. 2. Three-mass flexible model.](image)

### Table 1. Notations.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J_m, J_n, J_g$</td>
<td>moments of inertia</td>
</tr>
<tr>
<td>$k_g, k_a$</td>
<td>spring constants</td>
</tr>
<tr>
<td>$f_{mg}, f_{mg}, f_a$</td>
<td>viscous friction coefficients</td>
</tr>
<tr>
<td>$d_g, d_a$</td>
<td>damping coefficients</td>
</tr>
<tr>
<td>$r$</td>
<td>gear box ratio ($r = 1/118$)</td>
</tr>
<tr>
<td>$k_T$</td>
<td>torque constant</td>
</tr>
</tbody>
</table>

Applying torque balances for the three masses and introducing the states

$$x(t) = \begin{bmatrix} r\phi_m(t) - \phi_g(t) \\ \phi_m(t) \\ \phi_g(t) \\ \phi_a(t) \end{bmatrix},$$

the input signal $u(t) = \tau(t)$ and the output signal $y(t) = \dot{\phi}_m(t)$ gives the state space model

$$\dot{x}(t) = Ax(t) + Bu(t) \quad y = Cx(t),$$

where

$$A = \begin{bmatrix} 0 & 0 & r & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ -r \cdot k_g & 0 & -r^2 d_g + f_m & r \cdot d_g & 0 \\ k_g & k_a & r d_g & f_g + d_g + d_a & d_a \\ k_g & k_a & r d_g & f_g + d_g + d_a & d_a \\ 0 & d_a & -d_a + f_a & J_a \end{bmatrix}.$$
In the robot control system the motor angle $\phi_m$ is the only available output signal, but since the measurement noise is fairly small a reasonable estimate of the motor velocity is easily obtained. Therefore the motor velocity is used as output signal in the model above.

IV. EXPERIMENTAL RESULTS

4.1 Identification of nominal model

First, off-line identification is used to get nominal parameter values of the model. The determination of the nominal model is done by identifying a physically parameterized model as described in [4] and [11]. The external excitation that is added to the reference signal of the robot control system is a sum of sinusoids in the range 0-30Hz. The signal is created by setting the discrete Fourier transform of the reference signal to one with random phase and then transforming the Fourier transform to time domain. The sampling frequency of the data is 200Hz. Since the system is operating in closed loop the applied torque signal will be affected by the feedback. The properties of the torque signal are shown in Fig. 3, and it is seen that the input energy is low below 70 rad/s with peaks around 95, 125 and 185 rad/s.

The System Identification Toolbox in Matlab [5] is used in the off-line identification, and the acquired model is shown in Fig. 4. The acquired parameter values from the off-line identification will now be used as nominal parameter values in the recursive identification below.

4.2 Identified parameters

The recursive identification method presented in Section 2 can be used to identify all the physical parameters of the three-mass model shown in Fig. 3. To investigate the properties of the recursive identification method a subset of the physical parameters will be identified recursively. The choice of parameters that are going to be identified recursively originates from which parameter values that are likely to change over time. In these experiments the interest has been concentrated on the three parameters $k_T, J_a^{-1}$ and $f_m$, i.e.,

$$\theta = (k_T, J_a^{-1}, f_m)^T.$$  (25)

An example could be a worn gear box, which indicates an increase in $f_m$. Another situation is a change in the load acting on the arm, which would result in a change in the moment of inertia $J_a$. A further reason for choosing these three parameters is that their estimates will behave differently as will be seen below. There are of course other physical parameters that can be important to identify like, for example, the stiffness and damping of the springs. Off-line identification is used to obtain nominal values of the parameters that are not identified recursively.

4.3 Design variables

The recursive identification will be carried out using both the forgetting factor and the covariance matrix modification versions of the RPEM algorithm. The design parameters to choose are the initial values of $\theta(k)$ and $P(k)$ respectively and also the forgetting factor $\lambda$ and the matrix $\Delta$. The choices of $\lambda$ and $\Delta$ are trade-offs between tracking ability of the algorithm and the variance of the parameter estimates. In this application $\lambda = 0.995$ has been found to be an appropriate value. Since

$$B' = \begin{bmatrix} 0 & 0 & k_T & 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \quad (24)$$
the input can be chosen almost freely it can be designed to give sufficient excitation to avoid windup problems for the matrix $P(k)$. Hence the regularization parameter can be put to zero in this application. The choice of $P(0)$ can be made from different viewpoints. In case the algorithm is going to be used with some kind of change detection it is realistic to assume that $P(0)$ can be chosen to give a fast convergence after sudden changes in the true parameters. Without change detection the tracking properties will be determined by the current values of $P$, which depend on the choice of $\lambda$ and the properties of the input signal via $\psi(k)$. Assuming $\psi(k)$ to be quasi-stationary, see [12], and $\lambda$ to be close to one the matrix $P(k)$ can in steady state, see [13], be approximated by

$$
P = (1 - \lambda) Q^{-1},$$

and

$$
P Q P = \Delta,$$

respectively, where

$$
Q = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \psi(k) \psi^T(k),
$$

and $\mu$ has been put to zero. The components of $\psi(k)$ will be of different magnitude and hence will also the elements of $P$ be of different magnitude. This will then give different tracking and variance properties of the different parameter estimates. Using forgetting factor there is only one design variable available to affect the trade off between tracking and variance. Using the covariance modification the matrix $\Delta$ offers more freedom for dealing with this problem. The aim here is to show the algorithm properties without any change detection involved. In order to select an initial value of $P(k)$ that represents the steady state behavior the update equation for $P(k)$ is first run using the nominal parameter values in the computation of $\psi(k)$. The mean value of the diagonal elements in $P(k)$ are then used to form the initial value $P(0)$ in the actual identification. The initial value of the parameters in $\theta(k)$ is set to 20% above the nominal values of the parameters to see how well the algorithm adapts.

### 4.4 Results

The parameter adaptation for the forgetting factor case is shown in Fig. 5, and it can be seen that all parameters converge to their nominal values. The convergence rate is approximately the same for the estimates of $J_a$ and $f_m$ respectively, while it is somewhat higher for the estimate of $k_T$. In Fig. 6 the diagonal elements of $P(k)$ are shown.

![Fig. 5. Parameter estimates using the forgetting factor algorithm.](image)

![Fig. 6. Diagonal elements of $P(k)$. Solid: $P_{1,1}(k)$. Dashed: $P_{2,2}(k)$. Dotted: $P_{3,3}(k)$.](image)

Figure 6 shows a big difference in magnitude of the elements of $P(k)$, and this is caused by the big difference in the magnitude of the elements in $\psi(k)$. The energy of the elements of $\psi(k)$ is, according to Eq. (20), determined by the magnitude of $|G_\theta(e^{j\omega_T}, \theta)|$ and the magnitude of the input spectrum. Figure 7 shows the squared amplitude curve of the transfer operators $G_\theta(q, \theta)$ for the three parameters considered here, when the transfer operators are evaluated using the nominal parameter values.

![Fig. 7. Plot of $|G_\theta(e^{j\omega_T}, \theta)|^2$ for the three parameters. Solid line: $k_T$. Dashed line: $J_a^{-1}$. Dash-dotted line: $f_m$.](image)
The curve in Fig. 7 corresponding to $f_m$ (dash-dotted) lies above the two other for the main part of the frequency range. Multiplying this curve by the input spectrum and integrating then leads to a higher energy in the component of $\psi(k)$ corresponding to $f_m$. Equation (26) then implies that the corresponding element of $P$ will be smaller. Analogously it can be found that the component of $\psi(k)$ corresponding to $J_a^{-1}$ will have larger energy compared to the component corresponding to $k_T$. This is also clearly illustrated in Fig. 6.

An identification experiment is also carried out where the covariance matrix modification is used. The aim in this experiment is only to illustrate that this freedom is available. How to use it in a suitable way is left for further work. One possibility could be to choose the tracking and disturbance rejection trade-off individually for the estimated parameters. This is a main difference compared to the forgetting factor algorithm where all parameter estimates are affected by the choice of forgetting factor. For example, assume that it is desirable that the diagonal elements of $P$ are of the same order of magnitude. Due to the difference in magnitude in the elements of $\psi(k)$ it is then necessary to let the elements of $\Delta$ be of different magnitude. Let therefore

$$\Delta = \rho \begin{bmatrix} 5 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 500 \end{bmatrix},$$

where $\rho = 10^{-8}$. Figures 8 and 9 show the results from this experiment. Figure 9 shows that the diagonal elements of $P(k)$ now have the same magnitude. The changes in algorithm properties are illustrated in Fig. 8. The convergence rate for $k_T$ is lower and the variance is lower than for the forgetting factor case. The variance of the estimate of $f_m$ is much higher due to the increase in the corresponding element in $P(k)$. The results indicate that it is possible to handle the tracking and variance trade off more or less individually for the different parameters by suitable choices of the elements in $\Delta$.

In these experiments the excitation could be chosen almost freely. In practice it can not be expected that the input signal is sufficiently exciting all the time during normal robot operation. How to deal with this problem using the regularization procedure presented above is left for further work.

V. CONCLUSIONS

Recursive identification of continuous time parameters of a flexible robot arm has been considered. It has been illustrated how physical parameters of continuous time models can be recursively identified using discrete time data. Some aspects of the choice of method for obtaining tracking abilities of the algorithm have been illustrated.

ACKNOWLEDGMENT

This work was supported by ISIS and CENIT at Linköping University.

REFERENCES

4. Östring, M., S. Gunnarsson, and M. Norrlöf, “Closed Loop Identification of an Industrial Robot Containing


---

**Svante Gunnarsson** was born in Tranås, Sweden, 1959. He received the M.Sc. Degree in Applied Physics and Electrical Engineering from Linköping University, Sweden, in 1983. He received his Ph.D. in Automatic Control from Linköping University, Sweden, in 1988, and is currently working as an assistant professor in the Control and Communications group at the Dept. of E.E., Linköping University, Sweden. His research interests are in the areas of system identification, iterative learning control, and robot control.

---

**Måns Östring** was born in Stockholm, Sweden in 1974. He received the M.Sc. Degree in Applied Physics and Electrical Engineering in 1998 and his licentiate degree in Automatic Control in 2002, both from Linköping University, Sweden. Currently he is working as a lecturer in automatic control in the Physics and Electronics group, Department of Science and Technology, Linköping University, Norrköping, Sweden. His research topics include identification, fault detection and fault isolation, with focus on industrial robots.