Adaptive Control of a Modular Robot System

P. C. Mulders [2], R. J. Voorkamp, J. Jansen, A. C. H. van der Wolf [1]; Technical University Eindhoven/Netherlands
Received on January 16, 1992

Adaptive control is a process of modifying one or more parameters of the controller and these adaptive control algorithms are specially important for flexible manipulators with place- and time dependent parameters, varying during trajectory performance. Here an adaptive controller is described as a combination of the computed torque method and an adaptive PD controller based on the Model Reference Adaptive Control (MRAC) method. It has been applied to a modular robot for loads up to 50 kg consisting of a linear and a rotary actuator showing these parameter variations. Necessary models - extended and reduced - of this modular robot have been made and the proposed controller has been tested in simulations and in the real configuration also with respect to stability, convergence and robustness.

KEYWORDS: Robots, adaptive control.

1. INTRODUCTION

The aim of this work is a study of optimal adaptive control algorithms on systems with place- and time dependent parameters varying during trajectory performance, and to implement this on mechanical manipulators of industrial scale. Experiments on this item already have been done with a linear robot arm. [1]

To test these advanced control systems, a modular robot system - for loads up to 50 kg, consisting of a linear and a rotary actuator, as shown in Fig. 1 - has been constructed.

On the robot system a 3D-force sensor is mounted to perform teach-and replay trajectory operations. After the teach-operation the desired trajectory is again performed eventually with varying parameters, in which the already known motor control signals are updated by the adaptive control algorithm.

Industrial robots are used today for various purposes and until now robots control has been studied mostly under the assumption that actuators are stiff and that the links can be modelled as rigid bodies. Therefore most robots have a very stiff construction in order to avoid deformations and vibrations. For higher operating speeds industrial robots should be lightweight and flexible manipulators with place-control. Hence, more accurate dynamic models should be taken into account to pursue better dynamic performance.

With respect to these developments a number of optimal adaptive trajectory control strategies may be mentioned here e.g.: - the PD method - the computed torque method - the model reference adaptive control (MRAC) method.

All these methods should be considered with regard to convergence, stability and robustness. With the PID controller the deviation from the nominal trajectory is used in proportional, integral and differential form to correct and the PID gain factors are chosen with respect to the system dynamics. For coupled systems with interaction this controller type leads often to instability. By the robust and simple structure, the PID controller is often used as a standard to compare with other controllers.

The linear optimal controller is based on the minimization of a performance criterion-function, which may contain e.g., contributions of the deviations in trajectory positions and velocities, but also the control efforts like the motor control signals. [1] Even the boundaries for the control signals may be taken into account. The feedback control signals are formed by an optimal linear combination of the state variables of the system, which means optimal pole-placement.

The updating of parameters in the algorithm for adaptive control depends on the time to solve the matrix Riccati equation - derived from the performance criterion function - and this is strongly dependent on the number of state-space dimensions of the model. So the optimal trajectory control algorithm is based on a good knowledge of the system, but on the other hand the model should not be too complex, because it might increase the computation time of the optimal control law, so that on-line control becomes impossible.

The two concepts mentioned above are not well applicable to flexible robots with elastic deformations and time varying parameters.

So another approach to improve the behaviour of robots is the computed torque control method, sometimes called the inverse dynamics control. The necessary torques are calculated from the prescribed trajectory and here the control law is designed explicitly on the basis of a model in order to compensate for robot non-linearities. If flexibilities play an important role, it often results in an unacceptable system behaviour.

So the aim is to search for a control law achieving both reasonable trajectory tracking and a certain stabilization of acceptable vibrations.

Adaptive control is a process of modifying one or more parameters of the structure of the control system to force the response of the closed-loop system towards a desired trajectory. Among the various types of adaptive robot control systems the model reference adaptive control (MRAC) systems are important, since they lead to relatively easy to implement systems, with a high speed of adaptation and may be used in a variety of applications. However it is still difficult to derive convergence, stability and robustness conditions.

The applied MRAC - system is described more detailed in Ch.3. Study of the various control methods have been performed both in reality and computer simulations. So from this modular RT robot system an extended model - simulation model - has been made, which is verified by modal analysis techniques. Next the extended model has been reduced to a control model on which the various mentioned control methods are tested.

2. DESIGN AND MODELLING OF THE MODULAR ROBOT

2.1 The construction of the linear robot arm

Table 1. Design specifications of the linear robot arm.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum velocity</td>
<td>1 m/s</td>
</tr>
<tr>
<td>Maximum acceleration</td>
<td>2 m/s²</td>
</tr>
<tr>
<td>Maximum load</td>
<td>10 kg</td>
</tr>
<tr>
<td>Stroke</td>
<td>1 m</td>
</tr>
<tr>
<td>Position accuracy</td>
<td>0.01 mm</td>
</tr>
<tr>
<td>Power source</td>
<td>Heidenhain LS 513</td>
</tr>
<tr>
<td>Control system</td>
<td>DC-motor: Axem MC 19 PW 26, 1 kW</td>
</tr>
</tbody>
</table>

The mechanical construction is fairly stiff due to the hollow frame construction. The rotation of the motor into a translation of the actuator is converted by a spiral with a ball screw nut. An advantage of this combination is that the backlash can be eliminated by preloading the load. The DC motor is of the disc-armature type. Coupled to the motor shaft is also a tachogenerator and a rotational encoder.

For direct position measurement along the arm an optical linear digital incremental encoder has been mounted. Type Heidenhain LS 513, with a length of 1020 mm and an accuracy of 0.01 mm. The necessary frequency range of the encoder is determined by the speed of the arm and the accuracy of the lineal. The free end of the linear robot arm is extended with a 3D-force sensor, based on the bending principle and measured by strain gauges. The force sensor is used in the TEACH mode.

2.2. The construction of the rotational module.

The mechanical construction is based on a cylinder with slide rails - to minimize the deformation - and is fixed to a ground plate. The transmission from the motor to the turntable consists of a 4 stage toothed wheel combination with divided and preloaded wheels - realized with torsion springs - to eliminate backlash.
The task of the master SBC 186/03 is to control the motion of the robot. The PC also serves as the interface to the user. The optimal control law and the nominal trajectory is calculated off-line by this combination. The parameters of the robot are divided into modules, each one with its own properties like mass, stiffness, etc. The rotation as well as the translation is controlled by its own board h tel SBC 186/03. The rotation module has a motor amplifier, a tachogenerator, and a position measuring system. The position accuracy of the motor is ±0.005 mm. The accuracy may be multiplied by 5 or 10 for the encoder as a linear has<br/>

The rotation and translation are coupled to each other, so the eigenfrequencies of the linear actuator lying in between 0.370 to ± 0.965 m/s and an eventual load m = 0.4 kg. The lowest eigenfrequency \( f_0 \) of the linear actuator is \( f_0 = 110 \ldots 134 \text{ Hz} \). The lowest eigenfrequency \( f_0 \) of the rotational module is \( f_0 = 18 \ldots 20 \text{ Hz} \). The system's coupling is given by the combination of the 5 D.O.F. of the simulation model a compromise is made between the accuracy and the time domain. The model reduction is a compromise between accuracy and efficiency.

The rotation and translation are coupled to each other, so the eigenfrequencies of the linear actuator lying in between 0.370 to ± 0.965 m/s and an eventual load m = 0.4 kg. The lowest eigenfrequency \( f_0 \) of the linear actuator is \( f_0 = 110 \ldots 134 \text{ Hz} \). The lowest eigenfrequency \( f_0 \) of the rotational module is \( f_0 = 18 \ldots 20 \text{ Hz} \). The system's coupling is given by the combination of the 5 D.O.F. of the simulation model a compromise is made between the accuracy and the time domain. The model reduction is a compromise between accuracy and efficiency.

Many distinct components, each one with its own properties like mass, stiffness, etc., combine to a system with divided parameters. An attempt is made to realize a lumped mass model, which describes the behaviour of the robot as good as possible. This approach is a.o. based on previous studies [5] about drives of manipulators. With a complex system description with many parameters, it is necessary to reduce the sensitivity for parameter variations. By applying feedback one may reduce the sensitivity for parameter variations. This leads to higher gain factors, bigger control efforts and increases the possibility of instability.

3.3 The hierarchical control structure.

The on-line computer capacity of one controller is often too small or not fast enough to implement an advanced control algorithm in real time. With the combination of the 5 D.O.F. as a simulation model and a controller based on the 5 D.O.F. control model rather good results have been obtained.

Fig. 2. The hierarchical controller system.

The controller system consists of (Fig. 2) consists of 4 Intel single board computers working in parallel and 1 RAM board.

The rotation as well as the translation is controlled by its own board Intel SBC 186/03, coupled to its own input (the measuring system) and its own output (the motor amplifier) by the Intel SBC-interfaces. The task of each SBC 186/03 is:

1. to calculate - according to a control algorithm - the motor voltages
2. to read the position of each module
3. to store these data - motor voltages and positions - into the RAM board.
4. to communicate directly via the multibus system. Data however have to be transported to each other via the common R.4.M-board 028.

A PC 80386 is coupled to the master. In this PC software may be developed and tested. By a data switch it can be used as a terminal for each SBC.

The optimal control law and the nominal trajectory is calculated off-line by this combination. The rotation and translation are coupled to each other, so the eigenfrequencies of the linear actuator lying in between 0.370 to ± 0.965 m/s and an eventual load m = 0.4 kg. The lowest eigenfrequency \( f_0 \) of the linear actuator is \( f_0 = 110 \ldots 134 \text{ Hz} \). The lowest eigenfrequency \( f_0 \) of the rotational module is \( f_0 = 18 \ldots 20 \text{ Hz} \). The system's coupling is given by the combination of the 5 D.O.F. of the simulation model a compromise is made between the accuracy and the time domain. The model reduction is a compromise between accuracy and efficiency.

In this chapter an adaptive controller is proposed, which is a combination of the computed torque method for the main control input and an adaptive PD controller acting on the deviation of the desired trajectory (Fig. 3). The computed torque signal is derived directly from the equations of a control model. An adaptation algorithm based on the Model Reference Adaptive Control (MRAC) method adapts the PD gain-factors on-line for the additional loop.

The model reduction is arbitrary and some methods are known e.g. Guyan. In general the lowest eigenfrequency has most influence on the system dynamics and this one should be present in the reduced model.

With the combination of the 5 D.O.F. as a simulation model and a controller based on the 5 D.O.F. control model rather good results have been obtained. This controller has been implemented also with good result in the R-T module.

3.1 The non-adaptive controller.

As before this controller consists of two parts, a model dependent feedforward controller - also called the computed torque part - and a PD controller.

The computed torque controller computes by a control model the nominal control efforts - the torques - to perform a desired trajectory. Next the PD-controller acts on and compensates for the occurring trajectory error. The control model has to be a representative reduced model of the complete system. It may not be too complex, because otherwise, the computation time of this part of the control signal becomes too big. From the RT-configuration the 5 D.O.F. model (R2 T1) has been applied her.

So the control signal consists of \( u = u_{\text{model}} + u_{\text{PD}} \).

The control model has three degrees of freedom:

Rotation of the rotation module:

Rotation of the linear arm:

Translation of the linear arm:

Then the non-linear differential equations of movement are obtained.

Rotation:

\[
\begin{bmatrix}
\phi
\end{bmatrix} = J_{\text{R}} \phi_0 + \begin{bmatrix}
\mu_R
\phi_0
\end{bmatrix} + \begin{bmatrix}
\mu_R
\phi_0
\end{bmatrix} \frac{K_{\text{R}}}{J_{\text{R}}} \phi_0 = \begin{bmatrix}
\frac{K_{\text{R}}}{J_{\text{R}}}
\frac{K_{\text{R}}}{J_{\text{R}}}
\end{bmatrix} \phi_0
\]

Translation:

\[
\begin{bmatrix}
\phi_1
\end{bmatrix} = J_{\text{T}} \phi_1 + \begin{bmatrix}
\mu_T
\phi_1
\end{bmatrix} + \begin{bmatrix}
\mu_T
\phi_1
\end{bmatrix} \frac{K_{\text{T}}}{J_{\text{T}}} \phi_1 = \begin{bmatrix}
\frac{K_{\text{T}}}{J_{\text{T}}}
\frac{K_{\text{T}}}{J_{\text{T}}}
\end{bmatrix} \phi_1
\]

Substituting the desired trajectory in equation (2) to (4) delivers the nominal control torques, from which the nominal control voltages for the motors may be derived.
The real trajectory is compared with the desired trajectory and the deviation and PD control effort is obtained:

\[ \varepsilon = q - \hat{q} \]

\[ u_{PD} = -K_p \varepsilon - K_d \dot{\varepsilon} \]

The assumption is made that deviations in the rotation or translation only lead to a control effort in that degree of freedom. This means that \( K_p \) in \( K_d \) are of the following structure:

\[ K_p = \begin{bmatrix} K_{p1} & 0 & 0 \\ 0 & K_{p2} & 0 \\ 0 & 0 & K_{p3} \end{bmatrix} \]

\[ K_d = \begin{bmatrix} K_{d1} & 0 & 0 \\ 0 & K_{d2} & 0 \\ 0 & 0 & K_{d3} \end{bmatrix} \]

The feedback gains are determined such that the real system is stable with poles in the left half of the s-plane.

3.2 The adaptive controller.

Adaptive control is a special kind of feedback, in which the states of the process are divided in two categories, characterized by the difference in speed. The signposts to the degrees of freedom are quickly changing states, while the modelparameters are slowly changing.

The fast control loop is the PD-controller with the modeldependent feedforward (computed torque) pan. The systemparameters and subsequently the control parameters modelparameters and feedback gains are not constant, but they are updated in a slower control loop as an answer to the change in the dynamics of the process and its disturbances.

Combining (16) and (17) results in a differential equation for the deviation.

\[ \dot{A}q + (B + K_d)Aq + (C + K_d)\dot{q} = (-P - F) + (A^{-1}K_p(\dot{q}) + A^{-1}B(\dot{q}) + C(\dot{q}) + F) \]

The desired trajectory \( \dot{q} \) and \( \ddot{q} \) are at the right hand side of the equation (17). If the control parameters are constant, then the deviation \( \varepsilon \) asymptotically not become zero, but depend on \( \dot{q} \) and \( \ddot{q} \). Therefore \( A^{-1}K_p \) and \( A^{-1}B \) have to be adapted such that the right hand side of (18) becomes zero.

The feedback gains \( K_p \) and \( K_d \) are also adapted to get stability of the closed loop at the desired performance.

If the position - velocity error is defined as \( z(t) = (x(t) - \dot{x}(t)) \).

This transforms (18) into the adaptive system:

\[ \dot{z}(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{z}(t) + \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} v(t) \]

With \( K_d \) and \( K_p \) the relative dampingfactor and the undamped eigenfrequency.

The reference model is stable. Therefore \( \dot{z}(t) \) and it is assumed that the error of each DOF is decoupled and may be described as a second order differential equation.

\[ \dot{z}(t) = \Delta z(t) + \Delta \dot{z}(t) \]

Where \( \Delta z(t) \) and \( \Delta \dot{z}(t) \) are at the left of each equation and \( \Delta \) is defined by the designer.

The reference model is stable, so there exist a symmetric positive definite matrix \( P \), which obeys the Lyapunov equation (23).

\[ P = \begin{bmatrix} P_1 & P_2 \\ P_2^T & P_3 \end{bmatrix} \]

\[ PD = D^TP + Q \]

In which \( D \) is the 6 x 6 system- and \( Q \) is a symmetric constant 6 x 6 matrix.

From this the adaptation algorithm are derived so that for a trajectory the state of the adaptive system converges to the reference model. The unknown robotparameters \( A^*, B^*, C^* \) and \( F^* \) are slowly time dependent compared with the adaptation.

For \( \dot{R}(t) = P(t)\dot{z}(t) \) with matrices \( A, B, C \) and \( F \), representing the positionadaptationgain - defined by the designer - the controlparameters may be calculated from:

\[ P(t) = P(0) + \int_0^t \dot{R}(t)\dot{z}(t)dt \]

\[ C(t) = C(0) + \int_0^t \dot{R}(t)\dot{z}(t)dt \]

So summarizing the main properties of the adaptive control concept are:

1. Two control loops, a fast loop for the degrees of freedom and a slow loop to adapt the control parameters.
2. The controlparameters are adapted on-line.
3. Feedback takes place from the performance of the fast loop.
4. MEASUREMENT RESULTS
4.1 Simulations.

The simulations have been performed with the package PC-Matlab. It has a number of standard routines o.s. the Runge-Kutta difference routine to calculate the response of the system via a simulation model. For this the model with 3 DOF (R3T2) has been chosen. With the control model 3 DOF (RZT1) the computed torque part of the input signal on the desired trajectory is calculated.

The desired trajectory is a sinus sine wave in both directions shown in Fig. 5.
The minimal sample-time is 7 ms, derived from the implementation and also applied in the simulations. In Fig. 7 the results of the non-adaptive controller are shown with and without the computed torque part (feed forward control of the desired trajectory).

The controller rotation and by a factor 6 for translation compared with the non-adaptive gains are not chosen properly, unsmble.

4.2 Robustness and Adaptation Speed.

It is clear that the use of the feedforward control improves the control performance considerably. If an extra load is applied to the robot an additional uncertainty is introduced to the control law. In Fig. 8 the position-errors are shown with a load of 50 kg and a load of 0 kg for the non-adaptive controller.

The control performance (Fig. 9) becomes less but no instabilities occur. The controllers (with a load of 0 kg) are reduced by a factor 2 for position and by a factor 3 for translation compared with the non-adaptive controller.

The control performance (Fig. 10) becomes less but no instabilities occur. So the controller is rather robust.

A comparison between the performances of the adaptive and the non-adaptive controllers (with a load of 0 kg) is shown in Fig. 9.

The initial conditions of the control parameters are at the start of the trajectory the same for both the adaptive as the non-adaptive controller. The adaptation mechanism adapts the control parameters during the trajectory. The adaptive controller performs better than the non-adaptive controller.

The performance of the adaptive controller on different loads is shown in Fig. 10.

The position errors (Fig. 10 and Fig. 8) are reduced by a factor 2 for rotation and by a factor 6 for translation compared with the non-adaptive controller.

The adaptive controller however will try to stabilize this effect. This is called the robustness of the adaptive controller, where the adaptation mechanism is able to stabilize an initial unstable controller. Also the adaptation mechanism restricts the feedback gains to become negative. In the case that the control parameters are updated only every 20 samples a difference in the realised errors could hardly be noticed.

4.3 Implementation on the RT robot.

With the real R-T robot the same experiments have been performed as described in the simulations. It may be concluded that the use of the computed torque part improves the performance considerably.

In Fig. 11 a comparison is made between the adaptive and the non-adaptive controller. The adaptive controller needs a sufficient long trajectory to estimate the control parameters well. So the nominal trajectory here consists of four skew sine waves. The RT robot is rather stiff, so small variations in the load are easily compensated by the PD controller. With a load of 20 kg the adaptive controller tends to perform better. If the control model is chosen such that the parameter values are 30% lower than the real RT configuration, then the adaptation mechanism updates the control parameters such that the control performance rather quickly becomes better as shown in Fig. 11, which means a good robustness.

The experiments described above have been done with maximum adaptation speed. If this speed is reduced by a factor twenty, there is nearly no difference in the position error.

**CONCLUSIONS**

The application of feedforward (computed torque) control derived via a control model calculated from the desired trajectory as a nominal control effort improves the control performance considerably. The non-adaptive controller is sensitive to load variations, so a load of 50 kg makes the control performance worse.

The adaptive controller is preferable if the robot dynamics are poorly known. In that case a non-adaptive controller will give a bad control performance and possibly lead to instability.

The adaptation mechanism estimates the best control parameters and is an improvement compared to the non-adaptive controller. The adaptive controller is also rather robust. An initial deviation of the parameter values of the control model with 300% causes the adaptation mechanism to update the controlparameters quickly and results again in a good control performance.

**LITERATURE**