# Data-Based Optimal Control

Wouter Aangenent,[1] Dragan Kostić,[2] Bram de Jager,[3] René van de Molengraft,[4] and Maarten Steinbuch[5]

[1,3,4,5]Department of Mechanical Engineering
Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

[2]Embedded Systems Institute
P.O. Box 513, 5600 MB Eindhoven,
The Netherlands

[1]W.H.T.M.Aangenent@tue.nl, [2]dragan.kostic@esi.nl, [3]A.G.de.Jager@tue.nl,
[4]M.J.G.v.d.Molengraft@tue.nl, and [5]M.Steinbuch@tue.nl

***Abstract*— This paper deals with data-based optimal control. The control algorithm consists of two complementary subsystems, namely a data-based observer and an optimal feedback controller based on the system's Markov parameters. These parameters can be identified on-line using only input/output data. The effectiveness of the resulting controller is evaluated with a regulation and a tracking control experiment, performed on a direct-drive robot of spatial kinematics.**

## I. Introduction

建模困难耗时，引出后面基于数据的控制。

Plant modelling and model-based control design are the conventional steps to obtain feedback controllers. The performance of the resulting controller is strongly influenced by the quality of the underlying model. The derivation of an accurate model using first principles or using system identification can be a time consuming and not a straightforward task. Moreover, even complex models will not cover all system dynamics, and adaptive and robust feedback control methods have not completely overcome the problem of modelling errors. The problems inherent to plant modelling are the motivation of research in the field of data-based control techniques, such as: *unfalsified control* [1], *iterative feedback tuning* [2], *pulse response based control* [3], *Markov data-based LQG control* [4], *data-based predictive control* [5], *virtual reference feedback tuning* [6], *disturbance-based control* [7], and *simultaneous perturbation stochastic approximation* [8]. In the above techniques, the plant modelling step is circumvented, and the control design is only based on input/output data measured from the plant.

This paper investigates data-based optimal control, that is applicable on-line both to regulation and tracking control problems over an arbitrary time interval. A Markov data-based LQG control algorithm is suggested in [4]. This algorithm derives the control input by adequate processing of several measurements and it consists of two parts, namely, a data-based optimal feedback control of the estimated *controller state* and a data-based observer for this state. Both the feedback and observer are based on the system Markov parameters, i.e., on the values of the system's impulse response. A drawback of [4] is that these parameters are supposed to be known a priori. Another limitation of the algorithm is that the state observation and the computation of the control input are performed over a finite horizon. If this horizon consists of $N$ samples, then an equal number of Markov parameters is needed. Consequently, a large horizon requires an extensive number of Markov parameters which gives rise to a large computational burden. If control over an extensive period of time is needed, then one should iteratively compute the control input over a horizon of acceptable length. The algorithm of [4] does not allow iterative computation, since its observer cannot deliver the appropriate initial controller state at the beginning of each iteration. The data-based algorithm formulated in [5] computes feedback gains for a model predictive control over an arbitrary time interval. The restriction, though, is that the gains can be computed off-line only for linear time-invariant plants. These properties prohibit the use of these algorithms for on-line control.

In this paper, the limitations of [4] and [5] are circumvented. The obtained results follow from a combination of [4] and [9]. Such a combination has not yet been proposed and it enables experimental implementation. Therefore no experimental results are available in literature. To be precise, the observer part of [4] will be replaced by a data-based state observer proposed in [9]. The initialization problem is not present in this observer, which facilitates data-based optimal control during an arbitrary time interval. The control input can be generated on-line and can be used for the control of time-varying or nonlinear plants, since the algorithm is able to adapt to the actual dynamics by obtaining a linear model of the system at each sample. This will be demonstrated by experimental results obtained on a highly nonlinear direct-drive robot. The contributions of the paper are: (i) derivation of the data-based optimal control algorithm applicable over an arbitrary time interval; (ii) on-line implementation of the algorithm; (iii) experimental evaluation on a nonlinear plant.

The paper is organized as follows. The data-based optimal feedback controller is derived in the next section. An algorithm to observe the controller state and to identify the Markov parameters is presented in Section III. Section IV explains a setup for on-line implementation and describes the robot used in the experiments. The experimental results are presented in Section V. Conclusions are given at the end.

## II. Optimal control

To facilitate data-based control design, model-based optimal control theory is briefly reviewed in this section. A closed form solution to the difference Riccati equation is given, and the data-based optimal controller is derived.

### A. Model-based optimal control

Consider the linear time-invariant discrete-time system

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k), \end{aligned} \tag{1}$$

where $x(k)$ is the $n$ dimensional state vector, $u(k)$ is the $m$ dimensional control input vector, and $y(k)$ is the $\ell$ dimensional output vector. The objective of the well-known LQ control problem [10] is to find the functional

$$u(k) = f(A, B, C, Q, R, x(k)), \tag{2}$$

such that the quadratic cost

$$J = y(N)^T Q y(N) + \sum_{k=0}^{N-1} \left( y(k)^T Q y(k) + u(k)^T R u(k) \right) \tag{3}$$

is minimized subject to system (1), where $Q$ and $R$ are positive definite symmetric weighting matrices and $N$ denotes the length of the time interval of interest. The optimal input is given by [10]

$$u(k) = -K(k)x(k), \quad k = 0, 1, 2, \ldots, N-1, \tag{4}$$

where

$$K(k) = \left( R + B^T X(k+1) B \right)^{-1} B^T X(k+1) A, \tag{5}$$

and $X(k+1)$ is the solution of the difference Riccati equation

$$\begin{aligned} X(k) &= C^T Q C + A^T X(k+1) A \\ &\qquad - A^T X(k+1) B \left( R + B^T X(k+1) B \right)^{-1} \\ &\qquad \times B^T X(k+1) A, \\ X(N) &= C^T Q C. \end{aligned} \tag{6}$$

In subsection II-C, the data-based counterpart of (4) and (5) will be derived. This will require a closed-form solution to the difference Riccati equation (6) which is presented next.

### B. Closed-form solution to the Riccati equation

Here, a closed-form (batch-form) solution to the Riccati equation will be presented. Consider again system (1) and let the cost-function to be minimized be (3). Following a similar strategy as used in [11], one can determine the closed-form solution to be

$$X(k) = \mathbf{C}(k)^T (\mathbf{Q}(k)^{-1} + \mathbf{S}(k) \mathbf{R}(k)^{-1} \mathbf{S}(k)^T)^{-1} \mathbf{C}(k), \tag{7}$$

where

$$\mathbf{C}(k) = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{N-k} \end{bmatrix},$$

$$\mathbf{S}(k) = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ CB & 0 & \ddots & & \vdots \\ CAB & CB & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ CA^{N-k-1}B & CA^{N-k-2}B & \cdots & CB & 0 \end{bmatrix}$$

$$(\mathbf{S}(N) = 0), \tag{8}$$

$$\mathbf{R}(k) = \text{diag}(R, R, \ldots, R), \quad \mathbf{Q}(k) = \text{diag}(Q, Q, \ldots, Q).$$

Here, $\mathbf{R}(k)$ and $\mathbf{Q}(k)$ each contain $N - k + 1$ diagonal blocks. This solution can be easily verified to satisfy the recursive Riccati equation (6).

### C. Data-based optimal control

This section presents a data-based counterpart to the model based optimal controller given by (4),(5). It is shown in [4] that the optimal control sequence $u(k),\ k = 0, 1, 2, \ldots, N-1$, can be computed using the Markov parameters of system (1) instead of its state-space representation. The Markov parameters equal the impulse response data $M_i = CA^{(i-1)}B$ of the system at the sample times $i = 1, 2, \ldots, N$. To summarize [4], the optimal control sequence is obtained after the closed-form solution to the Riccati equation (7) is substituted in the model-based control law (4),(5), which yields

$$u(k) = \mathbf{G}(k) \mathbf{x}_c(k), \tag{9}$$

where

$$\begin{aligned} \mathbf{G}(k) = &-(R + \mathbf{B}(k+1)^T \mathbf{K}(k+1) \mathbf{B}(k+1))^{-1} \\ &\times \mathbf{B}(k+1)^T \mathbf{K}(k+1), \end{aligned} \tag{10}$$

$$\mathbf{B}(k+1) = \mathbf{C}(k+1) B = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{N-k} \end{bmatrix}, \tag{11}$$

$$\begin{aligned} &\mathbf{K}(k+1) = \\ &\quad (\mathbf{Q}(k+1)^{-1} + \mathbf{S}(k+1) \mathbf{R}(k+1)^{-1} \mathbf{S}(k+1)^T)^{-1}, \end{aligned} \tag{12}$$

$$
\mathbf{S}(k+1) = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ M_1 & 0 & \ddots & & \vdots \\ M_2 & M_1 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ M_{N-k-1} & M_{N-k-2} & \cdots & M_1 & 0 \end{bmatrix}, \tag{13}
$$

$$
(\mathbf{S}_N = 0),
$$

$$
\mathbf{x}_c(k) = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N-k} \end{bmatrix} x(k). \tag{14}
$$

The term $\mathbf{x}_c(k)$ represents the *controller state vector*, while the term $\mathbf{G}(k)$ is the optimal feedback gain. These terms, and the Markov parameters, can be computed using the data-based algorithm presented in the next section.

## III. Data-based Markov parameter and controller state estimation

This section provides a method to obtain the Markov parameters and the controller state vector based on the algorithm derived in [9]. Again, consider system (1). If the states in this system are repeatedly substituted, then for some $p \geq 0$,

$$
\begin{aligned} x(k+p) &= A^p x(k) + \mathbf{B}_p \mathbf{u}_p(k), \\ \mathbf{y}_p(k) &= \mathbf{O}_p x(k) + \mathbf{T}_p \mathbf{u}_p(k), \end{aligned} \tag{15}
$$

where $\mathbf{u}_p(k)$ and $\mathbf{y}_p(k)$ are vectors containing $p$ sets of input and output data starting with $u(k)$ and $y(k)$,

$$
\mathbf{u}_p(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+p-1) \end{bmatrix}, \quad \mathbf{y}_p(k) = \begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+p-1) \end{bmatrix}. \tag{16}
$$

The matrix $\mathbf{B}_p$ in equation (15) is an $n \times pm$ controllability matrix, $\mathbf{O}_p$ a $p\ell \times n$ observability matrix and $\mathbf{T}_p$ a $p\ell \times pm$ Toeplitz matrix of the system Markov parameters,

$$
\mathbf{B}_p = \begin{bmatrix} A^{p-1}B & \cdots & AB & B \end{bmatrix}, \quad \mathbf{O}_p = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{p-1} \end{bmatrix},
$$

$$
\mathbf{T}_p = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ CB & 0 & \ddots & \ddots & \vdots \\ CAB & CB & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ CA^{p-2}B & \cdots & \cdots & CB & 0 \end{bmatrix}. \tag{17}
$$

As long as $p\ell \geq n$, it is guaranteed for an observable system that there exists a matrix $\mathbf{M}$ such that [9]

$$
A^p + \mathbf{M}\mathbf{O}_p = 0. \tag{18}
$$

The existence of $\mathbf{M}$ ensures that, for $k \geq 0$, an expression exists where the state variable is eliminated from (15):

$$
\begin{aligned} x(k+p) &= (\mathbf{B}_p + \mathbf{M}\mathbf{T}_p)\mathbf{u}_p(k) - \mathbf{M}\mathbf{y}_p(k), \\ \mathbf{y}_p(k) &= \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{T}_p \end{bmatrix} \begin{bmatrix} \mathbf{u}_p(k-p) \\ \mathbf{y}_p(k-p) \\ \mathbf{u}_p(k) \end{bmatrix}, \end{aligned} \tag{19}
$$

where

$$
\mathbf{P}_1 = \mathbf{O}_p(\mathbf{B}_p + \mathbf{M}\mathbf{T}_p), \quad \mathbf{P}_2 = -\mathbf{O}_p\mathbf{M}.
$$

The parameters $\mathbf{P}_1$, $\mathbf{P}_2$, and the matrix $\mathbf{T}_p$ can directly be identified from a set of input/output data by solving the linear system (using a generalized inverse)

$$
\begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{T}_p \end{bmatrix} \mathbf{V} = \mathbf{Y}, \tag{20}
$$

where the data matrices are

$$
\mathbf{Y} = \begin{bmatrix} \mathbf{y}_p(k+p) & \mathbf{y}_p(k+p+1) & \cdots & \mathbf{y}_p(k+p+L) \end{bmatrix} \tag{21}
$$

$$
\mathbf{V} = \begin{bmatrix} \mathbf{u}_p(k) & \cdots & \mathbf{u}_p(k+L) \\ \mathbf{y}_p(k) & \cdots & \mathbf{y}_p(k+L) \\ \mathbf{u}_p(k+p)) & \cdots & \mathbf{u}_p(k+p+L) \end{bmatrix} \tag{22}
$$

The choice of the length $L$ of data is up to the designer but it should be in accordance with the rule stated in [12]: "To identify the Markov parameters uniquely, the data set must be sufficiently long and rich so that the rows of $\mathbf{V}$ associated with input data are linearly independent." Let $p = N+1$, then the Markov parameters $M_i = CA^{i-1}B$, $i = 1, 2, \ldots, N$ can be extracted from $\mathbf{T}_p$ in various ways, e.g. by direct selection from the first column. An estimate of the controller state vector (14) can be obtained via (19) and (15)

$$
\begin{aligned} \hat{\mathbf{x}}_c(k) &= \begin{bmatrix} \mathbf{0}_\ell & \mathbf{I}_{(N-k)\ell} \end{bmatrix} \mathbf{O}_p \hat{x}(k) \\ &= \begin{bmatrix} \mathbf{0}_\ell & \mathbf{I}_{(N-k)\ell} \end{bmatrix} (\mathbf{P}_1 \mathbf{u}_p(k-p) + \mathbf{P}_2 \mathbf{y}_p(k-p)), \\ p &= N-k+1. \end{aligned} \tag{23}
$$

The presented algorithm can be used both in open-loop stable and in closed-loop systems. The estimated controller state can be used in the control algorithm in Section II-C.

## IV. Implementation and experimental setup

In this section, the implementation of the data-based controller and the experimental setup are discussed.

### A. Implementation

The control setup is depicted in Fig. 1. In theory our approach aims at keeping the error $e = r - y$ at zero. However, since this is not feasible in practice, we are satisfied with keeping the error in a small band around zero. Thus, the control algorithm presented in Section II-C is based on $u$ and $e$ as the control input and the system output, respectively. These signals are used in the controller $\mathbf{C}$ as depicted in Fig. 1. To ensure that the data in $\mathbf{V}$ in (22) is rich enough, noise $n$ is added to the output of the controller. The control algorithm becomes effective after enough data is collected to completely fill $\mathbf{Y}$ and $\mathbf{V}$ defined in (21) and (22), respectively. During the transient period, i.e., while the necessary amount of data is acquired, the output of $\mathbf{C}$ is disabled and therefore, the system is operated in open-loop. Since this presents a potential danger, in the future other ways of obtaining this required data have to be sought. After $0.1$ second, sufficient data is collected, so $\mathbf{C}$ is enabled and its output is applied to the plant. The feedback law is defined by (9). The optimal feedback gain $\mathbf{G}(k)$, (10), and an estimate for the controller state vector, (14), are obtained each sampling instant by solving the linear system (20) after $\mathbf{Y}$ and $\mathbf{V}$ are constructed based on $u$ and $e$. This solution requires a high computational effort and therefore, in real-time implementation, the length of the horizon $N$ and the sample frequency are restricted. Every horizon, the optimal control input is computed, and at the end of each horizon the control algorithm is restarted with the last set of estimated states as initial condition. It should be noted that this is not the same as receding horizon control. A flow chart of the control algorithm is depicted in Fig. 2. This algorithm is able to adapt to the actual dynamics by virtue of the on-line implementation, which makes this controller also useful for systems with slowly time-varying dynamics or even for nonlinear systems, since at every sample a linearized model of the system is obtained from the measured data.

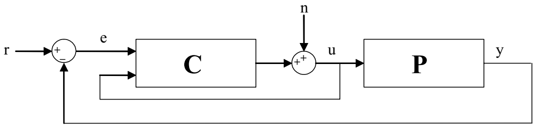


Fig. 1. Implementation scheme of the controller, $\mathbf{C}$ is the data-based controller and $\mathbf{P}$ is the controlled plant

### B. Experimental setup

The robotic arm, shown in Fig. 3, is the subject of our case study. It is an experimental facility for research in motion control [14], [13]. The photo and kinematic parameterizations according to the well-known Denavit-Hartenberg's (DH) notation [15], reveal three revolute degrees of freedom (d.o.f.), which makes such a kinematic structure referred to as RRR. Each d.o.f. is actuated by a gearless brushless DC direct-drive motor and has an infinite range of motions, thanks to the use of sliprings for the transfer of power and sensor signals. An RRR robot has an anthropomorphic structure featuring the most dexterous and versatile motions, leading to highly nonlinear kinematics and dynamic behavior. Direct drive actuation makes the control problem even harder, since there are no gear-heads to reduce the effects of nonlinear dynamics. The actuators are Dynaserv DM-series servos with nominal torques of 60, 30, and 15 [Nm], respectively. The servos are driven by power amplifiers with built in current controllers. Joint motions are measured using incremental optical encoders, with a resolution of approximately $10^{-5}$ [rad].

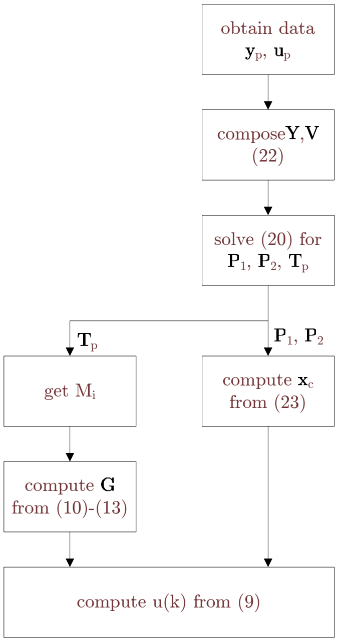

obtain data
$\mathbf{y}_\mathrm{p}$, $\mathbf{u}_\mathrm{p}$
compose$\mathbf{Y}$,$\mathbf{V}$
(22)
solve (20) for
$\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{T}_\mathrm{p}$
$\mathbf{T}_\mathrm{p}$
$\mathbf{P}_1$, $\mathbf{P}_2$
get $\mathrm{M}_\mathrm{i}$
compute $\mathbf{x}_\mathrm{c}$
from (23)
compute $\mathbf{G}$
from (10)-(13)
compute u(k) from (9)

Fig. 2. Flow chart of the control algorithm, executed at each sampling instant.

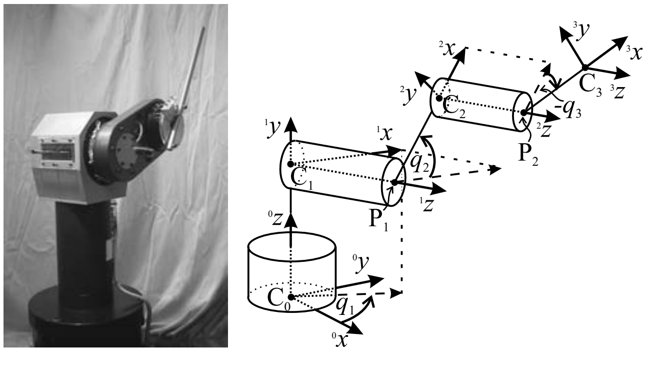
Both amplifiers and encoders are connected to a PC-based control system. This systems consists of the MultiQ I/O board from Quanser Consulting (8 × 13 bits ADC, 8 × 12 bits DAC, 8 digital I/O, 6 encoder inputs, and 3 hardware timers), combined with a real-time controller for Matlab/Simulink (Wincon). This facilitates the design of controllers in Simulink and their real-time implementation. The control system features a time delayed joint angular response to the given control input of two samples. The rigid body dynamics of the robot can be represented using a standard Euler-Lagrange formalism [16] by

$$\begin{aligned}\boldsymbol{\tau}(t) = \mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{C}\left(\mathbf{q}(t), \dot{\mathbf{q}}(t)\right)\dot{\mathbf{q}}(t) + \mathbf{g}(\mathbf{q}(t))\\ + \boldsymbol{\tau_f}\left(\mathbf{q}(t), \dot{\mathbf{q}}(t)\right)\end{aligned} \tag{24}$$

where $\boldsymbol{\tau}$ is a $3 \times 1$ vector of control torques, $\mathbf{M}$ is a $3 \times 3$ inertia matrix, $\mathbf{q}$, $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ are $3 \times 1$ vectors of joint motions, velocities, and accelerations, respectively, and $\mathbf{C}\dot{\mathbf{q}}$, $\mathbf{g}$, and $\boldsymbol{\tau_f}$ are the $3 \times 1$ vectors of Coriolis/centripetal, gravitational, and friction forces. The robot is a highly coupled system, for most configurations the mass matrix is non-diagonal which implies that a torque applied by one motor will result in movement of all three joints.

## V. Experimental results

In this section we present experimental results in controlling the motion $q_1$ of the RRR robot. This is the motion of the robot's first joint. The other two joints are not controlled and they are allowed to move freely. Because of the nonlinear couplings in the robot dynamics, the actuated motion influences the unactuated ones and vice versa. No compensation for the nonlinear dynamics is present. The motion $q_1$ corresponds to the output of the plant $\mathbf{P}$ shown



(a) (b)

Fig. 3. The RRR-robotic arm: (a) Photo, (b) Kinematic parameterizations according to the DH notation

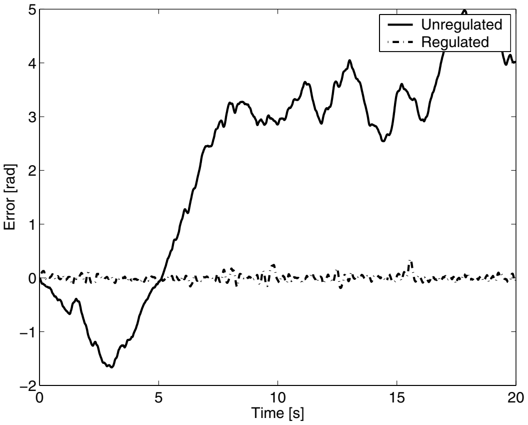


Fig. 4. Experimentally obtained position errors in the first experiment.

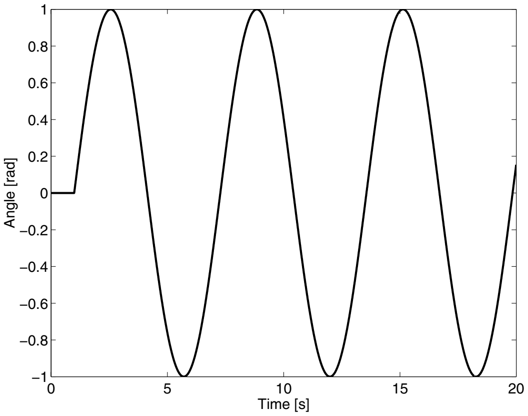


Fig. 5. Reference for second experiment.

in Fig. 1. The data-based algorithm presented in Section II-C generates the control torque which drives the motion in joint 1. This control torque corresponds to the control input $u$ to the plant, indicated in Fig. 1. Because the robot links adjacent to joints two and three were free swinging, and no compensation for friction is present, the dynamics from $u$ to $q_1$ is far from linear but rather nonlinear.

In the first experiment, an external noise signal $n$ was added to the system, see Fig. 1. When no control was applied, the external noise caused movement of joint 1, which is illustrated in Fig. 4 by the solid line. The objective of data-based control was to prevent the joint motion, in other words, the position error ($e = -q_1$) had to be regulated around zero. In the experiment, the sample time was set to $0.006$ seconds, the length of the control horizon was chosen to be $N = 10$, so $p = N+1 = 11$ and the length of the data was $L = 3p+10 = 43$. These values were selected based on the computational capacity of the control system. The input and output weightings in (3) were $R = 0.001$ and $Q = 3.5$, respectively, which indicates that there is almost no input weighting present. The value of $Q$ is obtained by trial-and-error, starting at 1 and increasing until an acceptable error level was reached, while preserving stability. It is clear from Fig. 4 that the data-based controller bounds the position error (dash-dotted line) to $0.3$ radians, which means that the regulation task has been successfully accomplished.

In the second experiment, the tracking ability of the controller was investigated. The reference joint motion $r$ is shown in Fig. 5 and, like in the regulation task, the data-based controller had to keep the position error $e = r - q_1$ around zero. The sample time, the length of the horizon, $p$, $L$, and the input weighting $R$ were the same as in the first experiment. External noise $n$ was added to ensure the input/output data was sufficiently rich. Two weightings for the position error were considered $Q = 1$ and $Q = 3.5$, and the achieved position errors are presented in Fig. 6. The noise added is this experiment is much smaller then in the first experiment, which explains the large difference between the regulation and tracking errors. Clearly, putting relatively more weight on $e$ results in more accurate tracking of the reference motion (smaller magnitude of the position error). To demonstrate the effectiveness of the data-based observer, Fig. 7 shows the difference between the measured position error and its reconstruction provided by the observer. Since in general this difference is small except for some peaks due to starting the trajectory, changing velocity direction, and disturbances, we can conclude that the performance of the observer is satisfactory. To illustrate the nonlinearity of the plant dynamics, the first, fifth, and tenth Markov parameters are depicted in Fig. 8. These parameters are on-line estimated using the algorithm from Section III. If the plant would be linear and time-invariant, all parameters would stay at constant levels. This is clearly not the case, since in Fig. 8 one may observe that the parameters are changing in time. These changes are due to the nonlinear plant dynamics. The spiky variations of the parameters are caused by the fact that the data in the matrices (21) and (22) does not belong to an LTI system, while Markov parameters are a representation of such a system. The results presented in Fig. 6 show that the data-based algorithm suggested in this paper is also capable to realize tracking control tasks.

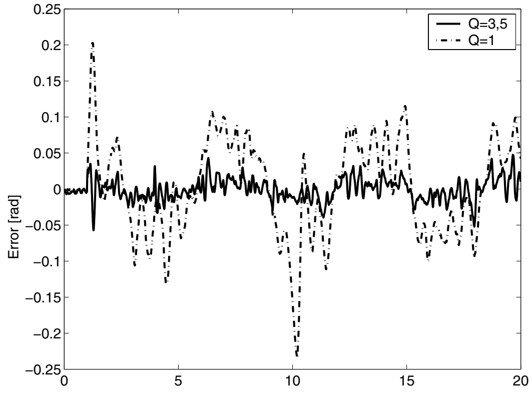


Fig. 6. Tracking error.

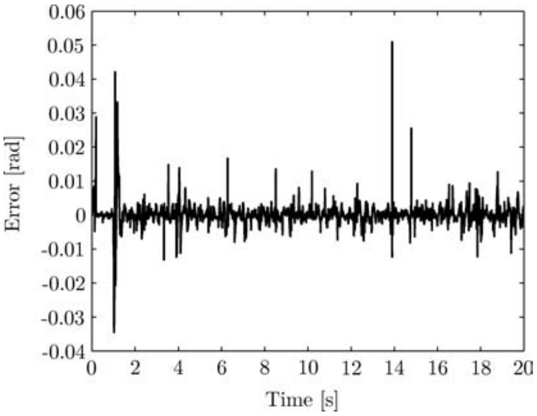


Fig. 7. Difference between measured and observed tracking error.

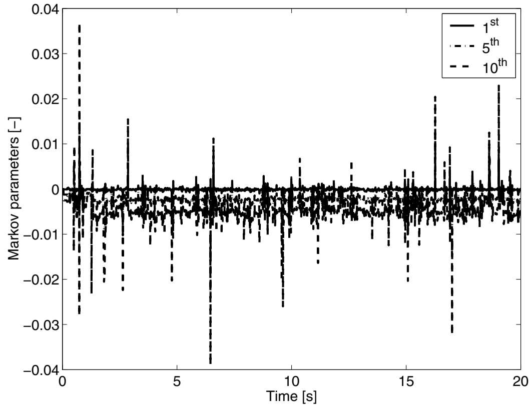


Fig. 8. On-line identified Markov parameters.

## VI. Conclusions

Data-based optimal control over an arbitrary time interval is suggested in this paper. The control input is computed using only measured input/output data of the system, by means of a data-based controller state observer and a Markov parameter based optimal feedback control algorithm. No parametric model of the plant nor controller is required. Although the control algorithm is based on linear time invariant system dynamics, on-line implementation enables the suggested data-based strategy to be used for control of nonlinear plants. Experimental evaluation on a direct-drive robot shows that the data-based strategy can be used for regulation and tracking control. Although the algorithm is implemented successfully real-time, there are still some issues to be resolved. At the moment the performance of this data-based controller is not sufficient to be useful for high performance motion systems. It is expected, based on simulations, that the results will improve if a larger control horizon is taken into account and if the sample time is lowered, which calls for a more computationally efficient implementation of the proposed algorithm.

## References

[1] M. G. Safonov and T. C. Tsao, “The Unfalsified Control Concept and Learning.” *IEEE Trans. Autom. Control*, Vol. 42, No. 6, 1997, 843–847.

[2] H. Hjalmarsson and M. Gevers, “Iterative Feedback Tuning: Theory and Applications.” *IEEE Control Syst. Mag.*, Vol. 18, No. 4, 1998, 26–41.

[3] J.K. Bennighof, S.M. Chang, and M. Subramaniam, “Minimum Time Pulse Response Based Control of Flexible Structures.” *J. Guid., Control and Dyn.*, Vol. 16, No. 5, 1993, 874–881.

[4] R. E. Skelton and G. Shi, “Markov Data-Based LQG Control.” *J. of Dyn. Systems, Meas., and Control*, Vol. 122, 2000, 551–559.

[5] R. K. Lim and M. Q. Phan, “Identification of a Multistep-Ahead Observer and Its Application to Predictive Control.” *J. Guid., Control and Dyn.*, Vol. 20, No. 6, 1997, pp. 1200-1206.

[6] A. Lecchini, M. C. Campi, and S. M. Savaresi, “Virtual Reference Feedback Tuning for Two Degree of Freedom Controllers.” *Internat. J. Adapt. Control Sig. Process*, Vol. 16, 2002, 355–371.

[7] R. L. Toussain, J. C. Boissy, M. L. Norg, M. Steinbuch, and O.H. Bosgra, “Suppressing Non-periodically Repeating Disturbances In Mechanical Servo Systems.” *Proc. IEEE Conf. Dec. Control*,Tampa, Florida, 1998, 2541–2542.

[8] J.C. Spall and J.A. Criston, “Model-free control of nonlinear stochastic systems with discrete-time measurements.” *IEEE Trans. Autom. Control*, Vol. 43, No. 9, 1998, 1198–1210.

[9] R. K. Lim, M. Q. Phan, and R.W. Longman, “State Estimation with ARMarkov Models.” *Technical report No. 3046*, Princeton University, Princeton, NJ, October 1998.

[10] F. L. Lewis and V. L. Syrmos, *Optimal Control*. John Wiley & Sons Inc., 1995.

[11] K. Furuta and M. Wongsaisuwan, “Closed-form solutions to discrete-time LQ optimal control and disturbance attenuation.” *Systems & Control Letters*, Vol. 20, 1993, 427–437.

[12] R. K. Lim, M. Q. Phan, and R.W. Longman, “State-Space System Identification with Identified Hankel Matrix.” *Technical report No. 3045*, Princeton University, Princeton, NJ, September 1998.

[13] B. van Beek and B. de Jager, “RRR-robot design: Basic outlines, servo sizing, and control.” *Proc. IEEE Internat. Conf. on Control Applications*, 1997, 36–41.

[14] B. van Beek and B. de Jager, “An experimental facility for nonlineair robot control.*” Proc. IEEE Internat. Conf. on Control Applications*, 1999, 668–673.

[15] K.S. Fu, R.C. Gonzales and C.S.G. Lee, *Robotics: Control, sensing, vision and intelligence*. McGraw-Hill, London, UK, 1987.

[16] D. Kostić, B. de Jager, M. Steinbuch and R. Hensen, “Modeling and Identification for Model-based Robot Control: An RRR-Robotic Arm Case Study.” *IEEE Trans. on Control Systems Techn.*, Vol. 12, No. 6, 2004, 904–919.