ECCOMAS Thematic Conference on Multibody Dynamics
June 29 - July 2, 2015, Barcelona, Catalonia, Spain

Energy Optimal Manipulation of an Industrial Robot ECCOMAS Thematic Conference on Multibody Dynamics 2015

[1]**Thomas Lauß**[† *], [2]**Peter Leitner**[*], [3]**Stefan Oberpeilsteiner**[† *], [4]**Wolfgang Steiner**[† *]

[*]Faculty of Engineering and Environmental Sciences
University of Applied Science Upper Austria
Stelzhamerstraße 23, 4600 Wels, Austria

[†] Institute for Mechanics and Mechatronics
Vienna University of Technology
Getreidemarkt 9, 1060 Wien, Austria

ABSTRACT

The main goal of this contribution is to determine the excitation of an industrial robot, such that the energy consumption becomes a minimum during the manipulation of the tool center point (TCP) from a start position to a given end point within a predefined time. Such tasks can be restated as optimization problems where the functional to be minimized consists of the endpoint error and a measure for the energy. The gradient of this functional can be calculated by solving a linear differential equation, called the *adjoint system* [3, 5, 4]. On the one hand the minimum of the cost functional can be achieved by the method of steepest descent where a proper step size has to be found or on the other hand by a Quasi-Newton algorithm where the inverse of the Hessian can be appreciated.

Keywords: optimal control, multibody dynamics, adjoint system, optimization, calculus of variation.

1 INTRODUCTION

Nowadays it is impossible to imagine industrial facilities without robots. Many of these manipulators have to do one operation repeatedly. Therefore it is obvious that the movements should be planed such that the energy effort is as little as possible. Due to the high amount of robots in the industry and their nearly unlimited operating time, already a small reduction of the energy per manipulation has an enormous consequence on the total industrial energy consumption.

In this contribution an approach to such inverse dynamical problems is presented. It starts from an optimal control formulation of the problem by introducing a cost functional which has to be minimized subject to a differential equation as an additional condition (c.f. [3, 4]). The gradient computation of the cost functional is based on the so called adjoint equations. Due to better convergence a Quasi-Newton method is used instead of the simple gradient method to find a minimum of the cost functional. Therefore the Hessian matrix is approximated by using the BFGS-algorithm (c.f. [10]).

For an energy optimal manipulation of the robot two different formulations of the cost functional are tested. On the one hand, the quadratic consideration of the inputs and on the other hand the mechanical power is taken into account. A *scrap-function* is used in both formulations which is important to reach the given end point.

The identified movements were tested on a PUMA six axis robot. With the measured control variables and the generalized velocities the required energy was evaluated. Based on this test data a considerably energy reduction was detected.

[1] Scientific Assistant, E-Mail: `thomas.lauss@fh-wels.at`, Phone: +43 (0)50804-44498

[2] Scientific Assistant, E-Mail: `peter.leitner@fh-wels.at`

[3] Scientific Assistant, E-Mail: `stefan.oberpeilsteiner@fh-wels.at` Phone: +43 (0)50804-44488

[4] Professor, E-Mail: `wolfgang.steiner@fh-wels.at` Phone: +43 (0)50804-43230

2 PROBLEM DEFINITION

At first, let us consider a nonlinear dynamical system

$$\begin{aligned}\dot{\boldsymbol{q}} &= \boldsymbol{v}\\ \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{v}} &= \boldsymbol{F}(\boldsymbol{q},\boldsymbol{v},\boldsymbol{u},t),\end{aligned} \tag{1}$$

where $\boldsymbol{q} \in \mathbb{R}^n$ is the vector of generalized coordinates and $\boldsymbol{v} \in \mathbb{R}^n$ is the vector of generalized velocities. In addition, $\boldsymbol{M}$ is the $n \times n$ mass matrix and $\boldsymbol{F} \in \mathbb{R}^n$ the force vector. The vector $\boldsymbol{u}$ indicates the control variables in an opened or enclosed region $\Omega \subseteq \mathbb{R}^m$. By introducing the vector of state variables $\boldsymbol{x}^T = (\boldsymbol{q},\, \boldsymbol{v})^T$ we may rewrite Equation (1) by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},t) \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0. \tag{2}$$

In general the force vector $\boldsymbol{f}$ is a continuous vector field which depends on the states $\boldsymbol{x}$, controls $\boldsymbol{u}$ and on time t. In robotics $\boldsymbol{u}$ are usually torques in the joints which appear linear on the right side of Equation (2). For this special case the equations of motion result in

$$\dot{\boldsymbol{x}} = \boldsymbol{a}(\boldsymbol{x},t) + \boldsymbol{B}(\boldsymbol{x},t)\boldsymbol{u}(t)$$

where $\boldsymbol{B}$ is the $2n \times m$ state- and time-variant input matrix. If the number of degrees of freedom n and the number of controls m are equal the system is called *fully-actuated*. For the case that the number of controls m is less than the number of degrees of freedom n we call the system *under-actuated*.

In robotics the position and velocity of the tool center point (TCP) will be of particular interest instead of the joint angles and angular velocities. Hence, the system output $\boldsymbol{y} \in \mathbb{R}^l$ is given by

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}).$$

In order to meet a predefined end point we have to satisfy the boundary condition

$$\boldsymbol{g}(\boldsymbol{x}(t_f)) = \bar{\boldsymbol{y}}. \tag{3}$$

However, we substitute the boundary condition of Equation (3) by the optimal control problem

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},t)\\ J &= \int_{t_0}^{t_f} h(\boldsymbol{x},\boldsymbol{u},t)\; dt + S(t_f,\boldsymbol{x}(t_f)) \longrightarrow \min\end{aligned} \tag{4}$$

where the integral describes the energy consumption and the *scrap function* S measures the end point error. If the closed region Ω is not empty the solution of the *optimal control* problem of Equation (4) leads to an energy optimal manipulation of the dynamical system of Equation (2).

From Pontryagin's minimum principle (c.f. [2]) the controls and states which lead to the optimal solution can be obtained. Therefore a two-point boundary value problem has to be solved, which is derived from the *Hamiltonian*

$$H(\boldsymbol{x},\boldsymbol{p},\boldsymbol{u},t) = h(\boldsymbol{x},\boldsymbol{u},t) + \boldsymbol{p}^T \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},t) \tag{5}$$

where $\boldsymbol{p}(t) \in \mathbb{R}^n$ is a continuous and piecewise continuously differentiable function which denotes the *costate* or *adjoint* variables. In every region $t \in [t_0, t_f]$ where $\boldsymbol{u}^*(t)$ is continuous the minimum condition

$$H(\boldsymbol{x}^*,\boldsymbol{p},\boldsymbol{u}^*,t) = \min_{\boldsymbol{u}\in\Omega} H(\boldsymbol{x}^*,\boldsymbol{p},\boldsymbol{u},t) \tag{6}$$

as well as the *adjoint* (costate) equation

$$\dot{\boldsymbol{p}} = -H_{\boldsymbol{x}}(\boldsymbol{x}^*,\boldsymbol{p},\boldsymbol{u}^*,t) \tag{7}$$

must hold. In Equation (7) the term $H_{\boldsymbol{x}}$ is a vector of partial derivatives of H with respect to the components of the state $\boldsymbol{x}$. In addition at the endpoint t_f the *transversality condition*

$$\boldsymbol{p}(t_f) = S_{\boldsymbol{x}}\big(\boldsymbol{g}\left(\boldsymbol{x}(t_f)\right)\big)$$

must be taken into account. Finally, the solution of Equation (4) is equivalent to the boundary value problem

$$\begin{aligned} \dot{\boldsymbol{x}} &= H_{\boldsymbol{p}} \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \\ \dot{\boldsymbol{p}} &= -H_{\boldsymbol{x}} \qquad \boldsymbol{p}(t_f) = S_{\boldsymbol{x}}\big(\boldsymbol{g}\left(\boldsymbol{x}(t_f)\right)\big) \\ \boldsymbol{u}^* &= \arg\min_{\boldsymbol{u}} H \end{aligned} \tag{8}$$

where the term $H_{\boldsymbol{p}}$ is a vector of partial derivatives of H with respect to the components of the co-state $\boldsymbol{p}$. Hence, if the controls are torques in a mechanical system, the Hamiltonian of Equation (5) are linear in $\boldsymbol{u}$. As a consequence the control $\boldsymbol{u}$ does not appear in $H_{\boldsymbol{u}}$. On the one hand the control $\boldsymbol{u}$ becomes infinite, if the region Ω is not bounded and on the other hand the singular case can occur. If this is the case, the control cannot be eliminated from Equation (8) and so it leads to a differential-algebraic boundary value problem which is considerably harder to solve.

3 GRADIENT COMPUTATION

To determine the gradient of the cost functional (4), Equation (2) is added to the cost functional

$$J = \int_{t_0}^{t_f} h(\boldsymbol{x},\boldsymbol{u},t) + \boldsymbol{p}^T \underbrace{\left(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},t) - \dot{\boldsymbol{x}}\right)}_{=0 \text{ Eq. (2)}} dt + S(t_f, \boldsymbol{x}(t_f)). \tag{9}$$

The Lagrange-multipliers $\boldsymbol{p}$ are denoted as adjoint variables and are arbitrary at this point. Integration by parts of the term $\int \boldsymbol{p}\dot{\boldsymbol{x}}$ leads to

$$\begin{aligned} J &= \int_{t_0}^{t_f} \left(h + \boldsymbol{p}^T \boldsymbol{f} + \dot{\boldsymbol{p}}^T \boldsymbol{x}\right) dt + S(t_f, \boldsymbol{x}(t_f)) - \boldsymbol{p}^T \boldsymbol{x}\Big|_{t_0}^{t_f} \\ &= \int_{t_0}^{t_f} \left(H + \dot{\boldsymbol{p}}^T \boldsymbol{x}\right) dt + S(t_f, \boldsymbol{x}(t_f)) - \boldsymbol{p}^T \boldsymbol{x}\Big|_{t_0}^{t_f}. \end{aligned} \tag{10}$$

In order to find a minimum of the cost functional J with respect to $\boldsymbol{u}$ we consider the variation of J according to a small change $\delta\boldsymbol{u}$ which is given by

$$\delta J = \int_{t_0}^{t_f} \left[\left(H_{\boldsymbol{x}}^T + \dot{\boldsymbol{p}}^T\right)\delta\boldsymbol{x} + H_{\boldsymbol{u}}^T \delta\boldsymbol{u}\right] dt + \left[S_{\boldsymbol{x}}^T(t_f, \boldsymbol{x}(t_f)) - \boldsymbol{p}^T(t_f)\right]\delta\boldsymbol{x}(t_f) + \boldsymbol{p}^T(t_0)\delta\boldsymbol{x}(t_0). \tag{11}$$

Due to the fact that no variation of the states at $t = t_0$ is allowed, the term $\boldsymbol{p}(t_0)\delta\boldsymbol{x}(t_0)$ is zero. If the adjoint variables are defined, such that

$$\dot{\boldsymbol{p}} = -H_{\boldsymbol{x}} \quad \text{and} \quad \boldsymbol{p}(t_f) = S_{\boldsymbol{x}}(t_f, \boldsymbol{x}(t_f)). \tag{12}$$

the variation of J according to Equation (11) is reduced to

$$\delta J = \int_{t_0}^{t_f} H_{\boldsymbol{u}}^T \delta\boldsymbol{u} dt. \tag{13}$$

Equation (12) may be solved backwards in time starting at $t = t_f$ after the system equations have been solved forward in the time interval $t \in [t_0, t_f]$. The largest possible increase of δJ is obtained, if $\delta\boldsymbol{u}(t)$ is chosen in the direction of $H_{\boldsymbol{u}}^T$. For that reason $H_{\boldsymbol{u}}^T$ may be considered as the gradient of the functional $J(\boldsymbol{u})$.

4 NUMERICAL DETERMINATION OF THE OPTIMAL CONTROL

Based on the adjoint gradient computation outlined in the previous section we may now search for a control $\boldsymbol{u}$ which minimizes the objective functional J. First of all, the method of steepest descent is described, where we always walk a certain distance along the negative gradient until we end up in a local minimum of J. Due to the costly line search step during every iteration and the slow convergence the gradient method is extended to a Quasi-Newton method. Therefore we solve the problem of finding $\boldsymbol{u}$ such that the gradient becomes zero.

4.1 The Method of Steepest Descent

The method of steepest descent tries to find a minimum of a function or subsequently of a functional by walking always along the direction of its negative gradient. This concept has first been developed to optimal control problems by H.J. Kelley [8] and A.E. Bryson [9].

The gradient is already derived from the adjoint system which is shown in Section 3. Now we use $H_{\boldsymbol{u}}^T$ and simply walk a short distance along the negative gradient of J. By reason of numerics the continuous functions are discretised. So the cost functional reads

$$J(\boldsymbol{u}) \approx \hat{J}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N) \tag{14}$$

where $\boldsymbol{u}_i = \boldsymbol{u}(t_i)$ and $t_1, \ldots, t_N$ is a sequence of consecutive time steps in the interval $[t_0, t_f]$. A variation of the controls $\boldsymbol{u}_i$ leads to a variation of the cost functional

$$\delta \hat{J} = \sum_{i=1}^{N} \frac{\partial \hat{J}}{\partial \boldsymbol{u}_i} \delta \boldsymbol{u}_i.$$

On the other hand, the variation $\delta \hat{J}$ can be expressed by Equation (13) which, after discretisation, results in

$$\delta \hat{J} = \sum_{i=1}^{N} H_{\boldsymbol{u},i}^T \Delta t_i \delta \boldsymbol{u}_i$$

where $H_{\boldsymbol{u},i}$ is the evaluation of $H_{\boldsymbol{u}}^T$ at $t = t_i$. Hence, the gradient of the discretised functional may be identified as

$$\frac{\partial \hat{J}}{\partial \boldsymbol{u}_i} = H_{\boldsymbol{u},i}^T \Delta t_i$$

with $\Delta t_i = t_i - t_{i-1}$. For walking in the direction of the negative gradient a small number $\kappa > 0$ has to be chosen to get the increment

$$\delta \boldsymbol{u}_i = -\kappa H_{\boldsymbol{u},i}^T \Delta t_i. \tag{15}$$

If κ is sufficiently small, the updated control $\boldsymbol{u}_i + \delta \boldsymbol{u}_i$ will always reduce the cost functional J. However, finding the number κ such that J is absolutely reduced may require several simulations of the system equations. For that purpose, the increments given by Equation (15) are considered as functions of κ. After solving the equations of motion with $\boldsymbol{u} + \delta \boldsymbol{u}$ as inputs also the objective function J becomes ultimately a function of κ. By means of a line search algorithm one may find a number κ in a predefined interval $[0, \kappa_{\max}]$ which minimizes J.

4.2 Application of a Quasi-Newton Method

It is well known that the convergence of the gradient method is rather slow. Hence, a Newton method provides an alternative approach to find the minimum of the cost functional J. The basic idea is the following one: If $\hat{\boldsymbol{u}} = (\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, \ldots, \boldsymbol{u}_N^T)^T$, the minimizing vector $\hat{\boldsymbol{u}}$ is defined by a zero gradient, i.e. by the equations

$$\nabla \hat{J} = \left[\frac{\partial \hat{J}}{\partial u_1}, \cdots, \frac{\partial \hat{J}}{\partial u_N} \right]^T = 0$$

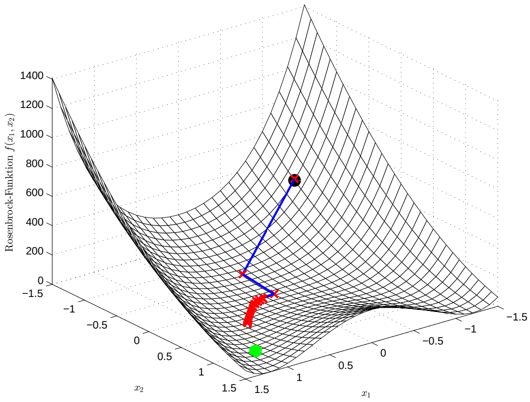

Figure 1. convergence of the gradient method

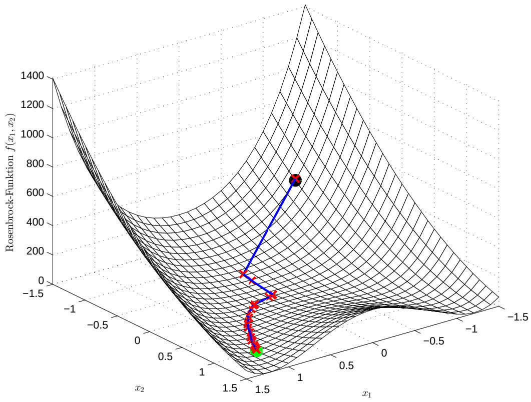

Figure 2. convergence of the Quasi-Newton method

which can be solved for $\hat{\boldsymbol{u}}$ by Newton's method. However, the Hessian $\boldsymbol{H} = \left(\nabla \hat{J}\right)_{\hat{\boldsymbol{u}}}$ is required for that purpose. To avoid the full computation of $\boldsymbol{H}$, which would be extremely time consuming, several quasi-Newton methods have been developed. They all approximate the Hessian by using the gradients of successive Newton-iterations. For example, the Hessian can be estimated efficiently by the well known *Broyden-Fletcher-Goldfarb-Shanno* (BFGS)-Algorithm (c.f. [10]). Even its inverse can be efficiently obtained by applying the *Sherman-Morrison formula* (c.f. [11]).

We compute an approximation $\tilde{\boldsymbol{H}}^{-1}$ of the inverse of the Hessian from the BFGS-algorithm. Then, an increment $\delta\hat{\boldsymbol{u}}$ of the discretised control signal is given by

$$\begin{pmatrix} \delta \boldsymbol{u}_1 \\ \delta \boldsymbol{u}_2 \\ \vdots \\ \delta \boldsymbol{u}_N \end{pmatrix} = -\tilde{\boldsymbol{H}}^{-1} \nabla \hat{J} \tag{16}$$

Note, that it is strongly recommended to use a quasi-Newton method which directly approximates the inverse of the Hessian. Otherwise, if the original Hessian is computed, a very large and dense matrix must be inverted, since the number of components of J might become large.

The inverse of the Hessian after $k+1$ iterations is given by

$$\tilde{\boldsymbol{H}}_{k+1}^{-1} = \left(\boldsymbol{I} - \frac{\boldsymbol{p}_k \boldsymbol{q}_k^T}{\boldsymbol{q}_k^T \boldsymbol{p}_k}\right) \tilde{\boldsymbol{H}}_k^{-1} \left(\boldsymbol{I} - \frac{\boldsymbol{q}_k \boldsymbol{p}_k^T}{\boldsymbol{q}_k^T \boldsymbol{p}_k}\right) + \frac{\boldsymbol{p}_k \boldsymbol{p}_k^T}{\boldsymbol{q}_k^T \boldsymbol{p}_k} \tag{17}$$

where $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{p}_k$ is the gradient direction of the k^{th}-iteration and $\boldsymbol{q}_k$ is the change of the gradient during the last iteration [11].

Table 1. comparison of the two numeric methods

method	number of Iterations	$f(\boldsymbol{x}^*)$	$\|(\nabla f)(\boldsymbol{x}^*)\|$
Gradient method	57	0.123	1.198
Quasi-Newton method (BFGS)	23	$5.4 \cdot 10^{-12}$	$9.2 \cdot 10^{-6}$

In Figure 1 and Figure 2 the comparison of the convergence of the gradient and the Quasi-Newton method is pictured. Therefore the function `fminunc` from the MATLAB-Optimization toolbox is used. The function which has to be minimized is called Rosenbrock's banana function. It is a non-convex function which is often used as a performance test for optimization algorithms. In Figure 1, it is conspicuous that the convergence of the gradient method is very bad near the

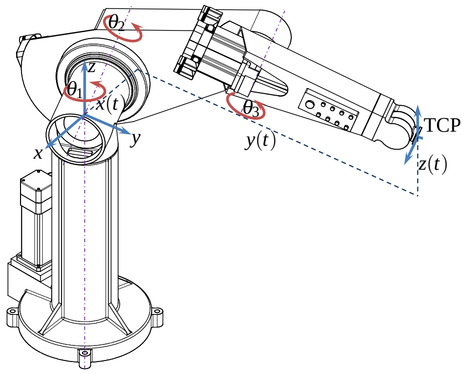

Figure 3. schematics of the six-axis PUMA robot

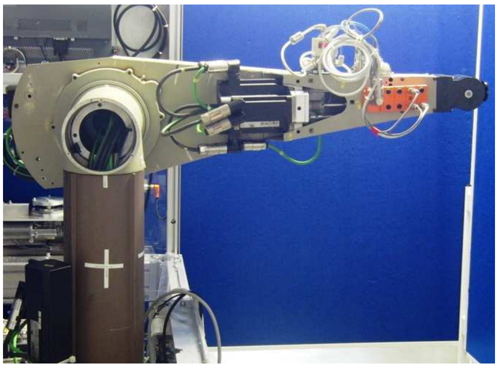

Figure 4. image of the six-axis PUMA robot

minimum. However the Quasi-Newton method is considerably better due to the use of the Hessian matrix. In Table 1 the comparison of the two methods is shown.

5 APPLICATION TO THE SIX-AXIS-ROBOT

The presented method is used to optimize the energy consumption of the robot with respect to two different definitions of the cost functional. The reason why we have chosen this PUMA robot [5] is that a lot of different parameters are available which are necessary for the evaluation and verification of the results. Afterwards the simulation results are verified at a real six-axis-robot which is shown in Figure 4.

5.1 Problem definition

The system consists of three degrees of freedom, θ_1, θ_2 and θ_3 which denote the relative rotation angles of the joints. Due to the complicated structure of the equations of motion and the minor influence on the energy consumption the three wrist joints are fixed. First of all the equations of motions are derived and have the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0$$

where $\boldsymbol{u} = [M_1, M_2, M_3]^T$ contains the torques of the motors and $\boldsymbol{x} = [\theta_1, \theta_2, \theta_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^T$ is the vector of states of the dynamical system. The system output

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x})$$

is a nonlinear function which depends on the states and describes the coordinates of the tool center point $\boldsymbol{y} = [x(t), y(t), z(t)]^T$.

Table 2. start and end position of the robot

	start position	final position	start velocity	final velocity
θ_1	$0°$	$-90°$	$0\,\text{rad/s}$	$0\,\text{rad/s}$
θ_2	$0°$	$-10°$	$0\,\text{rad/s}$	$0\,\text{rad/s}$
θ_3	$0°$	$45°$	$0\,\text{rad/s}$	$0\,\text{rad/s}$
x_{TCP}	$-0.15320\,\text{m}$	$0.81441\,\text{m}$	$0\,\text{m/s}$	$0\,\text{m/s}$
y_{TCP}	$0.92112\,\text{m}$	$-0.15320\,\text{m}$	$0\,\text{m/s}$	$0\,\text{m/s}$
z_{TCP}	$0.02032\,\text{m}$	$0.22233\,\text{m}$	$0\,\text{m/s}$	$0\,\text{m/s}$

[5] The PUMA (Programmable Universal Machine for Assembly, or Programmable Universal Manipulation Arm) is an industrial robot arm developed by Victor Scheinman for General Motors at the Stanford University.

For the energy optimal manipulation of the robot from a start-point $\boldsymbol{x}_0$ to a given end-point $\bar{\boldsymbol{y}}$, $\dot{\bar{\boldsymbol{y}}}$ (c.f. Table 2) within a predefined time t_f two different definitions of the cost functional are tested. First, a very common approach in pertinent literature that minimizes the quadratic signal energy (c.f. [6]) is applied. In this case the cost functional is given by

$$J_1 = \underbrace{\int_{t_0}^{t_f} \boldsymbol{u}^T \boldsymbol{u}\, dt}_{\text{signal-energy}} + S(t_f, \boldsymbol{x}(t_f)). \tag{18}$$

The advantage of this method is that the convergence rate of the optimization process is quite good due to the quadratical formulation. On the other hand this measure is not of practical importance, it is rather a mathematical construct which has been established due to good convergence. Hence, it suggests itself that one could take the real mechanical energy into account. In this case the cost functional

$$J_2 = \underbrace{\int_{t_0}^{t_f} \boldsymbol{u}^T \boldsymbol{F}_{\boldsymbol{u}}^T \boldsymbol{v}\, dt}_{\text{mechanical energy}} + S(t_f, \boldsymbol{x}(t_f)) \tag{19}$$

has to be minimized, where $\boldsymbol{v}$ is the velocity of the appropriate degree of freedom (DOF) and $\boldsymbol{F}_{\boldsymbol{u}}$ combines the input signals with the proper DOF. The integrand of Equation (19) symbolizes the mechanical power at one point in time.

The scrap-function S of Equation (18),(19) describes the endpoint error and is specified by

$$S(\boldsymbol{x}, t) = \alpha \left\{ \beta \underbrace{\left[\boldsymbol{y}(\boldsymbol{x}) - \bar{\boldsymbol{y}}\right]^2}_{\text{error in position}} + \underbrace{\left[\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} - \dot{\bar{\boldsymbol{y}}}\right]^2}_{\text{error in velocity}} \right\} \tag{20}$$

where α and β are proper weighting factors and $\bar{\boldsymbol{y}}$, $\dot{\bar{\boldsymbol{y}}}$ contains the position and velocity of the endpoint in coordinates of the system output.

5.2 Results

The left diagram of Figure 5 shows the signal energy effort of the standard manipulation in comparison to the optimization with respect to the signal energy. For the sake of completeness also the signal energy expenditure of the optimization with respect to the mechanical energy is pictured. As a result the reduction of the signal energy after the optimization process is about 47% with respect to the standard manipulation of the robot control. On the right hand side of Figure 5 the real mechanical energy effort is pictured. A reduction of 34% could be achieved if the mechanical energy is taken into account in the cost functional. Hence, the minimal signal energy does not ensure an energy optimal manipulation of the robot.

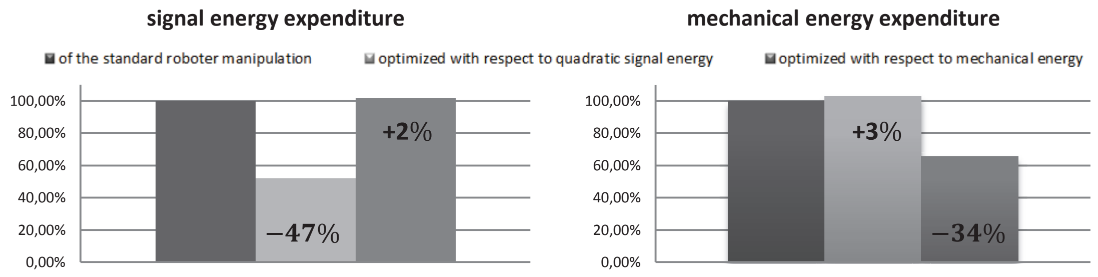

Figure 5. comparison of the energy expenditure

In Figure 6 the joint angles of the optimized motions in comparison to the standard manipulation of the robot are plotted over time. It can be seen that the prescribed end position of the motion, optimized with respect to the mechanical energy, is not met exactly. This angular deviation results in a small endpoint error with a magnitude of 2.49% with respect to the TCP-vector $[x(t_f),y(t_f),z(t_f)]^T$ at the end-time t_f which is marked in Figure 3. In practical terms this means that one could reduce the energy expenditure by 34% if the end position is modified slightly.

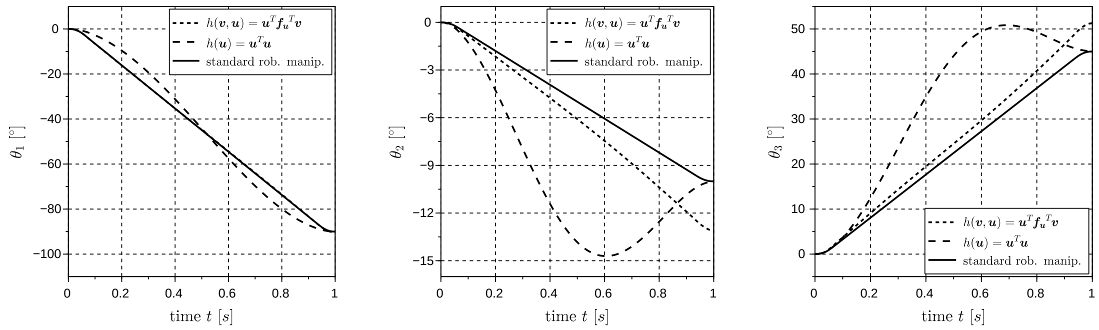

Figure 6. trajectory of the states

In consideration of the fact that we do not take friction into account, the comparison of the measure versus the simulation is quite good. The trend of the energy expenditure over the time during the manipulation is shown in Figure 7. The solid line results from a real measurement of the robot which is pictured in Figure 4 while the dashed line shows the energy consumption of the simulation.

The energy conservation for the measured signal amounts to 0.84Ws while the simulated results reduces the energy about 2.5Ws. Both values differ clearly from each other because we do not take friction into account.

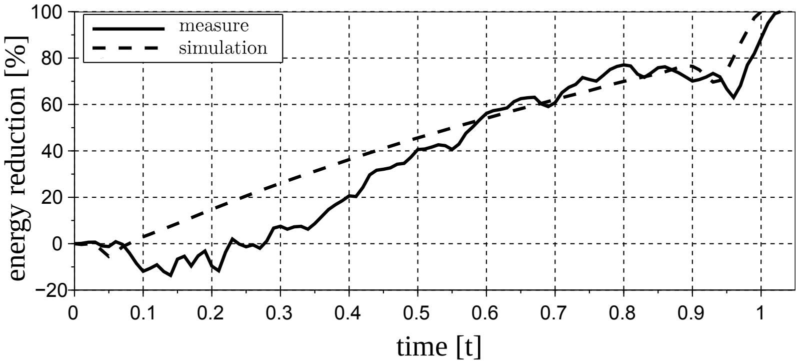

Figure 7. build-up of the mechanical energy reduction

6 CONCLUSIONS AND OUTLOOK

There are two main statements summing up the results. On the one hand this paper should reveal that the trajectory with minimal signal energy does not lead automatically to the energy optimal manipulation of the robot. Nevertheless in practice such quadratic input terms are often taken into account because this leads to less stress of the motors. In simply terms you can say that the electrical parts are more protected against overheating and the operation life span is increased additionally if the torques remain small.

On the other hand the effort of the mechanical energy results from a small endpoint error. Due to the fact that the mathematical model of the robot is conservative, no dissipation of the energy is possible. In other words, the energy consumption is independent of the specific motion and depends only on the initial and final state of the robot. Certainly this begs the question about the purpose of this paper. Nevertheless some practical applications exists. A typical example for this field of application would be the planing of a production line. First the positions of the robots are determined such that the robot can reach every essential location. Only a small change of the basic position of the robot reduces the energy effort of the manipulation.

If we consider the functional of Equation (19) in detail one can recognize that the integrand can be a negative value for example if the velocity is positive and the torque is negative. That means that the motor decelerates the motion and recuperates energy which reduces the total consumption. Nowadays modern robots are able to recuperate energy during a manipulation. However, if older robots are used which do not have the ability to recuperate energy during braking the cost functional can be adapted such that this effects are considered.

A video can be found at `https://youtu.be/i2K2znfmEBI` where the standard manipulation is overlaid with the optimized motions. The small endpoint error for the mechanical energy optimal manipulation can be seen very well in the video.

For the results in Section 5.2 we neglected the three degrees of freedom of the wrist and fixed them to keep the equations of motion and the necessary matrices simple. However, if we consider this joint angles in the system equations it is possible to reach a predefined endpoint in different ways. This means that more than one final configuration of the robot exists which meet the end point in the coordinates of the tool center point. Therefore this algorithm leads to the solution with the lowest potential energy. In other words the movements of the heavy and large bodies which require most of the energy are as small as possible.

For such cases the optimization process can be implemented in the robot control. Furthermore the identification can be done during operation. Instead of the forward simulation the measures of the previous manipulation can be used to solve the adjoint system and calculate the gradient. Hence, the required energy decreases during the manipulations of the robot. A big advantage is that it is not necessary to exchange any part of the robot, only an update of the robot control is required.

ACKNOWLEDGEMENTS

This project was supported by the program "Regionale Wettbewerbsfähigkeit OÖ 2010-2013", which is financed by the European Regional Development Fund and the government of Upper Austria.

REFERENCES

[1] V.I. Arnol'd. *Mathematical Methods of Classical Mechanics*. Graduate Texts in Mathematics. Springer Science & Business Media, 1989

[2] Donald E. Kirk. *Optimal Control Theory - An Introduction*. Courier Dover Publications, Mineola, New York, 2012

[3] S. Reichl, W. Steiner. The Optimal Control Approach to Dynamical Inverse Problems. *Journal of Dynamic Systems, Measurement, and Control*. Vol. 134, Is. 2, Doi. 10.1115/1.4005365, 2012

[4] K. Nachbagauer, S. Oberpeilsteiner, K. Sherif, W. Steiner. The Use of the Adjoint Method for Solving Typical Optimization Problems in Multibody Dynamics. *Journal of Computational and Nonlinear Dynamics*. doi: 10.1115/1.4028417. 2014

[5] D. Bestle, P. Eberhard. Analyzing and Optimizing Mutlibody Systems. *Mechanics of structures and machines*, Vol. 20, pp. 67–92, 1992.

[6] S. Breun. Optimale Steuerung redundanter Roboter auf Mannigfaltigkeiten: Strukturanalyse und numerische Realisierung. *Disseration*. 2007

[7] R.E. Bellman. *Dynamic Programming*. Princeton, N.J.: Princeton University Press, 1957.

[8] H.J. Kelley. *Method of Gradients, Optimization techniques with applications to aerospace systems* Mathematics in Science and Engineering, Elsevier Science, 1952

[9] A.E. Bryson. Optimal Programming Problems with Inequality Constraints II: Solution by Steepest-Ascent *AIAA Journal*, (1964), 25-34.

[10] J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, C.A. Sagastizábal. *Numerical Optimization - Theoretical and Practical Aspects.* Springer Berlin Heidelberg, 2006

[11] Jack Sherman and Winifred J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Ann. Math. Statist.*, 21(1):124–127, 03 1950.