Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2007
September 4-7, 2007, Las Vegas, Nevada, USA

DETC2007-34780

PATH-CONSTRAINED MOTION PLANNING FOR ROBOTICS BASED ON KINEMATIC CONSTRAINTS

N.J.M. van Dijk*
N. van de Wouw
H. Nijmeijer
Eindhoven University of Technology
Department of Mechanical Engineering
P.O. Box 513, 5600 MB Eindhoven
The Netherlands

W.C.M. Pancras
Bosch Rexroth Electric Drives and Controls B.V.
Product area semiconductor and medical
P.O. Box 7170, 5605 JD Eindhoven
The Netherlands

ABSTRACT

Common robotic tracking tasks consist of motions along predefined paths. The design of time-optimal path-constrained trajectories for robotic applications is discussed in this paper. To increase industrial applicability, the proposed method accounts for robot kinematics together with actuator velocity, acceleration and jerk limits instead of accounting for the generally more complex dynamic equations of a manipulator with actuator torque and torque-rate limits. Besides actuator constraints also constraints acting on process level are accounted for. The resulting non-convex optimization problem is solved using a cascade of genetic algorithms and Nelder-Mead's method. Simulations performed on a Puma 560 manipulator model show that for a proper choice of the kinematic constraints results can be obtained that match the quality of those obtained using the more complex dynamic constraint approach.

1 INTRODUCTION

One of the main issues in developing autonomous robots is robot motion planning. Robot motion planning consists of path planning and trajectory planning. Path planning handles the planning of a spatial path from the robot's initial configuration to the robot's goal configuration, while trajectory generation consists of deciding on the velocity, acceleration and jerk profiles along the planned path as a function of time. Considering robotic applications, two major tasks can be distinguished, namely (1) pick-and-place motions (e.g. component mounting applications) and (2) motions for which the path is fully known in advance, also termed path-constrained motion, (e.g. laser welding/cutting applications). Here, we will only address industrial applications for which a path is fully known in advance.

A common performance criterion to increase productivity and to lower cost prices of (industrial) robots is time-optimality. The problem of finding path-constrained time-optimal trajectories for industrial applications has received a lot of attention in literature. The problem was firstly solved in the mid 80's when Bobrow et al. [1], Shin and McKay [2] and Pfeiffer and Johanni [3] independently have presented algorithms for determining path constrained time-optimal motions including robot dynamics. In their approach, actuator torque constraints are rewritten to constraints in terms of the path parameter. Planning trajectories by accounting only for actuator torque constraints results in motions that are not second-order continuous. Instant changes in desired torque levels cannot be generated by an actuator due to its electrical dynamics. This implies that from the start of a trajectory a difference between the desired joint position and the actual joint position will exist. Furthermore, instant changes in torque level introduce a considerable amount of wear of the actu-

*Corresponding author (email: N.J.M.v.Dijk@tue.nl).
The work presented in this paper is part of the New*Motion* project, that is supported by Stimulus.

ator [4], and causes excessive vibrations. This limits the practical usability of time-optimal trajectory planning algorithms that result in bang-bang motions [5]. Different solutions are proposed to account for discontinuous torque levels, see [5, 6, 7].

All of the methods discussed above incorporate actuator torque limits (and possibly also torque rate or jerk limits) as constraints for the trajectory generation problem. This implies that the equations of motion of the manipulator under consideration must be known. From a practical point of view this is not desirable. Deriving the equations of motion asks for specific knowledge on multi-body dynamics, that may not be available in a industrial environment. Furthermore, in most algorithms only constraints on joint/actuator level are taken into account, while for certain industrial applications (such as e.g. welding and/or laser cutting applications) also constraints in the workspace of the manipulator can limit the motion of the end-effector. Zlajpah [8] and Dong and Stori [9] explicitly limit maximum velocities acting on the process. However, for some applications next to maximum process velocity also maximum acceleration and jerk of the manipulator's end-effector should be limited.

As shown in [7], the path-constrained motion planning problem can be written as a Time-optimal Control (TOC) problem with constraints on states and input. Time-optimal control problems with bounded controls, have been solved analytically using Pontryagin's minimum principle[1] [10]. Despite the research done in the field of time-optimal control, time-optimal control for high-order systems is still an open issue [11]. Numerical methods can be used to solve the problem. A method which uses a combination of a genetic algorithm and a local optimization method can be used for several classes of linear systems [12]. Recently, Chettibi [13] presented an optimization strategy for path constrained motions using genetic algorithms.

In this paper, a method is presented for determining time-optimal path-constrained motions subjected to velocity, acceleration and jerk constraints acting on both the manipulator actuators and the process it is performing. Using the path description, the manipulator's kinematic relations and the defined constraints, a non-convex optimization problem is defined that is solved using a hybrid optimization strategy. The applicability of this approach in an industrial environment is highly increased due to the fact that the approach presented here only uses the manipulator's kinematic relations which are easy to determine (compared to the generally complex equations of motion needed by the algorithms discussed above). This approach will result in a somewhat conservative solution due to the fact that the constraints are constant over the entire workspace. For the remainder of this paper it is assumed that obstacle avoidance is satisfied during the path definition and is therefore not considered explicitly.

This paper is organized as follows. In Section 2, the path-constrained motion planning problem is defined and the derivation of both process and actuator constraint limitations in terms of the path parameter will be discussed. Section 3 describes the hybrid optimization strategy for determining time optimal trajectories. Simulation results using the presented strategy, will be presented in Section 4. Finally, conclusions and an outlook on future work are given in Section 5.

2 PATH CONSTRAINED MOTIONS BASED ON KINEMATIC CONSTRAINTS

2.1 Problem formulation

As discussed in the previous section, the overall problem is to find time-optimal trajectories, i.e. to minimize the time span $T = t_f - t_0$ of the total motion, i.e. $\min T = \min \int_{t_0}^{t_f} dt$, subjected to manipulators kinematics, $\underline{P} = \underline{R}(\underline{q}(t))$, constraints on the path, $\underline{P} = \underline{P}(s)$, with s the so-called path parameter; actuator constraints,

$$|\dot{q}_i| \leq \dot{q}_{i,max},\ |\ddot{q}_i| \leq \ddot{q}_{i,max},\ |\dddot{q}_i| \leq \dddot{q}_{i,max}, \text{ for } i = 1,\ldots,n_j, \quad (1)$$

and process constraints,

$$|\dot{X}_i| \leq \dot{X}_{i,max},\ |\ddot{X}_i| \leq \ddot{X}_{i,max},\ |\dddot{X}_i| \leq \dddot{X}_{i,max}, \text{ for } i = 1,\ldots,6. \quad (2)$$

Herein, the path, that must be followed by the manipulator's end-effector, is represented by a six-dimensional vector $\underline{P}$, which consists of a translational component $\underline{P}_t$ (with three translational coordinates) and an angular component $\underline{P}_o$ (with three angular coordinates), relative to some reference frame $\underline{P} = (\underline{P}_t^T, \underline{P}_o^T)^T$. Furthermore, q_i is the displacement of joint i and n_j the number of joints ($\underline{q}^T = (q_1,\ldots,q_{n_j})$), X_i component i of the end-effector's pose vector $\underline{X}$ and $\underline{R}(\underline{q}(t))$ represents the manipulator's forward kinematics. It is assumed that a joint only allows a rotation or translation in one degree-of-freedom. Since the total motion time is unknown, the problem is to find a mapping from time onto the path parameter (i.e. $s(t)$), while respecting the constraints as defined above. Constraints are given as function of time. So a transformation from time to the path parameter must be defined. This is discussed in the following sections. First, the actuator constraint transformation will be discussed. Second, the process constraints will be treated and finally the results will be combined to derive the set of constraints on the path parameter.

2.2 Constraint formulation

2.2.1 Actuator constraints The actuator limitations can be e.g. determined using the actuator data sheet or by performing some experiments. The path, defined by $\underline{P} = \underline{P}(s)$ can be written in terms of the joint displacements $\underline{q}$ using the kinematic relations of the manipulator $\underline{P}(s(t)) = \underline{R}(\underline{q}(t))$ with $\underline{R}(\underline{q}) =$

[1] In Russian literature it is referred to as Pontryagin's maximum principle due to a different sign convention.

$[R_1(\underline{q}),\ldots,R_6(\underline{q})]$. In order to determine the constraints on the path velocity $\dot{s}$, induced by the actuator constraints, $\underline{P}(s(t)) = \underline{R}(\underline{q}(t))$ is differentiated with respect to time. When the inverse of the Jacobian $\underline{R}_{\underline{q}}(\underline{q})$ exists and the manipulator is non-redundant we can determine the joint velocity angles as

$$\dot{\underline{q}} = (\underline{R}_{\underline{q}}(\underline{q}))^{-1}\underline{P}_s(s)\dot{s}, \tag{3}$$

with $(\underline{R}_{\underline{q}}(\underline{q}))_{ij} = \frac{\partial R_i(\underline{q})}{\partial q_j}$ and $P_{i,s}(s) = \frac{\partial P_i(s)}{\partial s}$, for $i = 1,\ldots,6$ and $j = 1,\ldots,n_j$. Using the actuator velocity constraints in (1), the constraint on $\dot{s}$ depending on actuator velocity constraints can be determined by,

$$\dot{s}_{i,ACvmax} \leq \frac{\dot{q}_{i,max}}{|(\underline{R}_{\underline{q}}(\underline{q}))^{-1}\underline{P}_s(s)|_i},\ i = 1,\ldots,n_j. \tag{4}$$

As a consequence, the constraint on $\dot{s}(t)$ depends on $s(t)$ (and thus also on $\underline{q}(t)$. Differentiating $\underline{P}(s(t)) = \underline{R}(\underline{q}(t))$ twice with respect to time results in

$$\dot{\underline{q}}^T R_{i,\underline{qq}}(\underline{q})\dot{\underline{q}} + R_{i,\underline{q}}(\underline{q})\ddot{\underline{q}} = P_{i,ss}(s)\dot{s}^2 + P_{i,s}(s)\ddot{s},\ i = 1,\ldots,6, \tag{5}$$

with $P_{i,ss}(s) = \frac{\partial^2 P_i}{\partial s^2}$ and $(R_{i,\underline{qq}}(\underline{q}))_{jk} = \frac{\partial^2 R_i(\underline{q})}{\partial q_k \partial q_j}$. Combining (3) and (5), we obtain

$$\ddot{\underline{q}} = \underline{R}_{\underline{q}}^{-1}(\underline{q})\left[(\underline{P}_{ss}(s) - \underline{Q})\dot{s}^2 + \underline{P}_s(s)\ddot{s}\right], \tag{6}$$

with $\underline{Q} = \{Q_i\}$ and $Q_i = \underline{P}_s^T(s)\underline{R}_{\underline{q}}^{-T}(\underline{q})R_{i,\underline{qq}}(\underline{q})\underline{R}_{\underline{q}}^{-1}(\underline{q})\underline{P}_s(s)$ for $i = 1,\ldots,6$.

For the case in which $\ddot{s} = 0$, i.e. when the acceleration capacity is available for centripetal acceleration, restrictions are posed on $\dot{s}$. Using $\ddot{s} = 0$ and substituting the acceleration constraints on joint level (i.e. $\ddot{\underline{q}}_{max,i},\ i = 1,\ldots,n_j$) into (6), the limitation on $\dot{s}$ due to maximum acceleration is given as

$$\dot{s}_{ACamax,i} = \sqrt{\frac{\ddot{q}_{max,i}}{|\underline{R}_{\underline{q}}(\underline{q})^{-1}\left(\underline{P}_{ss}(s) - \underline{Q}\right)|_i}},\ \text{for } i = 1,\ldots,n_j. \tag{7}$$

Consequently, using (4) and (7) the overall admissible $\dot{s}$ due to actuator velocity and acceleration constraints, can be determined by,

$$\dot{s}_{max,ac} = \min_{i\in\{1,\ldots,n_j\}}(\dot{s}_{ACamax,i}, \dot{s}_{ACvmax,i}). \tag{8}$$

Herein, the subscript $_{ac}$ stands for actuator constraints. Next, when $\ddot{s} \neq 0$ the limitation on $\ddot{s}$ is determined by

$$\ddot{s}_{i,acmin} \leq \ddot{s} \leq \ddot{s}_{i,acmax},\ i = 1,\ldots,n_j, \text{where} \tag{9}$$

$$\ddot{s}_{i,acmin} = \frac{-\ddot{q}_{i,max} + |\underline{R}_{\underline{q}}(\underline{q})^{-1}(\underline{P}_{ss}(s) - \underline{Q})|_i\dot{s}^2}{|\underline{R}_{\underline{q}}(\underline{q})^{-1}\underline{P}_s(s)|_i}, \tag{10}$$

$$\ddot{s}_{i,acmax} = \frac{\ddot{q}_{i,max} - |\underline{R}_{\underline{q}}(\underline{q})^{-1}(\underline{P}_{ss}(s) - \underline{Q})|_i\dot{s}^2}{|\underline{R}_{\underline{q}}(\underline{q})^{-1}\underline{P}_s(s)|_i}. \tag{11}$$

As can be seen from (9), (10) and (11) the constraints on $\ddot{s}$ depend on the maximum joint acceleration levels, the path parameter s and the pseudo-velocity $\dot{s}$. The limits on $\ddot{s}$ are determined by evaluating (9) at each point along the path, for pseudo-velocities varying from $0 \leq \dot{s} \leq \dot{s}_{max}$, where $\dot{s}_{max}$ will be defined later on when the limits due to actuator constraints and process constraints will be combined. So, at each point along the path with a certain pseudo-velocity $\dot{s}$ a maximum and minimum pseudo-acceleration can be determined.

Finally, the constraints due to the jerk limitations on the actuator need to be determined. Differentiating (5) with respect to time results in

$$\begin{aligned} R_{i,\underline{qqq}}(\underline{q},\dot{\underline{q}})\dot{\underline{q}} + 3\ddot{\underline{q}}^T R_{i,\underline{qq}}(\underline{q})\dot{\underline{q}} + R_{i,\underline{q}}(\underline{q})\dddot{\underline{q}} = \\ P_{i,sss}(s)\dot{s}^3 + 3P_{i,ss}(s)\dot{s}\ddot{s} + P_{i,s}(s)\dddot{s},\ \text{for } i = 1,\ldots,n_j, \end{aligned} \tag{12}$$

where $\dot{q}$ and $\ddot{q}$ can be determined using (3) and (6), respectively, and

$$\left(R_{i,\underline{qqq}}(\underline{q},\dot{\underline{q}})\right)_j = \dot{\underline{q}}^T R_{i,\underline{qqq}j}(\underline{q})\dot{\underline{q}}, \tag{13}$$

with $\left(R_{i,\underline{qqq}}(\underline{q})\right)_{kl} = \frac{\partial^3 R_i(\underline{q})}{\partial q_l \partial q_k \partial q_j}$. From (12) we can derive the following limitations on $\dddot{s}$

$$\frac{-|R_{i,\underline{q}}\dddot{\underline{q}}_{i,max} + Z_i|}{|P_{i,s}(s)|} \leq \dddot{s} \leq \frac{|R_{i,\underline{q}}\dddot{\underline{q}}_{i,max} + Z_i|}{|P_{i,s}(s)|}, \text{with} \tag{14}$$

$$\begin{aligned} Z_i = R_{i,\underline{qqq}}\underline{R}_{\underline{q}}^{-1}(\underline{q})\underline{P}_s\dot{s} + 3\left[\underline{R}_{\underline{q}}^{-1}(\underline{q})((\underline{P}_{ss}(s) - \underline{Q})\dot{s}^2 + \right. \\ \left.\underline{P}_s\ddot{s})\right]^T R_{i,\underline{qq}}\underline{R}_{\underline{q}}^{-1}(\underline{q})\underline{P}_s\dot{s} - 3P_{i,ss}\dot{s}\ddot{s} - P_{i,sss}\dot{s}^3, \\ \text{for } i = 1,\ldots,n_j, \end{aligned} \tag{15}$$

with $\underline{Q}$ as similar defined in (6). At this point, it assumed that the influence of the jerk joint constraints on the limitation of pseudo-velocity $\dot{s}$ and pseudo-acceleration $\ddot{s}$ can be neglected. This is done to simplify the constraint evaluation. Furthermore, in practice jerk limitations are used as a smoothing factor, and

are therefore relatively large in comparison to velocity and acceleration limits such that the influence on pseudo-velocity and acceleration is generally not critical.

2.2.2 Process constraints The process constraints acting on the path can be determined in the same manner as the actuator constraints. Since the path is defined in the workspace of a manipulator, the kinematics of the manipulator are not needed for this transformation. The path in the workspace is defined as $\underline{X}(t) = \underline{P}(s(t))$ with $\underline{X}(t)$ the predefined path of the robot hand as a function of time. Differentiating this expression with respect to time and by using the process constraints (2), we can determine the maximum admissible value of $\dot{s}$ due to the maximum velocity allowed in the workspace,

$$\dot{s}_{PCvmax,i} = \begin{cases} \dfrac{\dot{X}_{i,max}}{|P_{i,s}(s)|}, \text{ if } P_{i,s}(s) \neq 0 \\ \infty, \text{ if } P_{s,i}(s) = 0, \end{cases} \tag{16}$$

for $i = 1, \ldots, 6$. By differentiating $\underline{X}(t) = \underline{P}(s(t))$ twice with respect to time and regarding the case for which $\ddot{s} = 0$, the maximum pseudo-velocity $\dot{s}_{max,pc}$ due to the maximum process acceleration can be determined as

$$\dot{s}_{PCamax,i} = \sqrt{\frac{\ddot{X}_{i,max}}{|P_{i,ss}(s)|}} \text{ for } i = 1, \ldots, 6, \tag{17}$$

$$\dot{s}_{max,pc} = \min_{i \in \{1,\ldots,6\}} (\dot{s}_{PCvmax,i}, \dot{s}_{PCamax,i}). \tag{18}$$

Herein, the subscript $_{pc}$ stands for process constraints. The maximum pseudo-acceleration $\ddot{s}_{max,pc}$ is determined as

$$\begin{aligned} \frac{-\ddot{X}_{i,max} + |P_{i,ss}(s)|\dot{s}^2}{|P_{i,s}(s)|} &\leq \ddot{s} \\ &\leq \frac{\ddot{X}_{i,max} - |P_{i,ss}(s)|\dot{s}^2}{|P_{i,s}(s)|}, \; i = 1, \ldots, 6. \end{aligned} \tag{19}$$

Finally, the limitations on the third derivative of the path parameter s with respect to time, induced by jerk process constraints are defined as follows

$$\begin{aligned} \frac{-(\dddot{X}_{i,max} - |\underline{P}_{sss}(s)\dot{s}^3 + 3\underline{P}_{ss}(s)\dot{s}\ddot{s}|_i)}{|P_{i,s}(s)|} &\leq \dddot{s} \\ \leq \frac{(\dddot{X}_{i,max} - |\underline{P}_{sss}(s)\dot{s}^3 + 3\underline{P}_{ss}(s)\dot{s}\ddot{s}|_i)}{|P_{i,s}(s)|}&, \; i = 1, \ldots, 6. \end{aligned} \tag{20}$$

2.2.3 Combining actuator and process constraints Here, the constraints on the pseudo-velocity, -acceleration and -jerk determined using the actuator constraints and process constraints are gathered such that one constraint definition is obtained. For the pseudo-velocity this results in $0 < \dot{s} \leq \min(\dot{s}_{max,ac}, \dot{s}_{max,pc})$, where $\dot{s}_{max,ac}$ and $\dot{s}_{max,pc}$ are defined in (8) and (18), respectively.

Similarly, the pseudo-acceleration constraint is formulated as $a(\dot{s},s) \leq \ddot{s} \leq b(\dot{s},s)$, with $a(\dot{s},s) = \max_{i \in \{1,\ldots,n_j\}, k \in \{1,\ldots,6\}} (a_{ac,i}(\dot{s},s), a_{pc,k}(\dot{s},s))$ and $b(\dot{s},s) = \min_{i \in \{1,\ldots,n_j\}, k \in \{1,\ldots,6\}} (b_{ac,i}(\dot{s},s), b_{pc,k}(\dot{s},s))$ and the values $a_{ac,i}$, $b_{ac,i}$, $a_{pc,k}$ and $b_{pc,k}$ defined as

$$a_{ac,i} = \frac{-\ddot{q}_{i,max} + |\underline{R}_{\underline{q}}(\underline{q})^{-1}(\underline{P}_{ss}(s) - \underline{Q})|_i \dot{s}^2}{|\underline{R}_{\underline{q}}(\underline{q})^{-1}\underline{P}_s(s)|_i}, \tag{21}$$

$$b_{ac,i} = \frac{\ddot{q}_{i,max} - |\underline{R}_{\underline{q}}(\underline{q})^{-1}(\underline{P}_{ss}(s) - \underline{Q})|_i \dot{s}^2}{|\underline{R}_{\underline{q}}(\underline{q})^{-1}\underline{P}_s(s)|_i}, \; i = 1, \ldots, n_j \tag{22}$$

$$a_{pc,k} = \frac{-\ddot{X}_{k,max} + |\underline{P}_{ss}(s)|_k \dot{s}^2}{|P_{k,s}(s)|}, \tag{23}$$

$$b_{pc,k} = \frac{\ddot{X}_{k,max} - |\underline{P}_{ss}(s)|_k \dot{s}^2}{|P_{k,s}(s)|}, \; k = 1, \ldots, 6. \tag{24}$$

The total pseudo-jerk limitation $\dddot{s}$ is defined as follows, $c(s,\dot{s},\ddot{s}) \leq \dddot{s} \leq d(s,\dot{s},\ddot{s})$ with $c(s,\dot{s},\ddot{s}) = \max_{i \in \{1,\ldots,n_j\}, k \in \{1,\ldots,6\}} (c_{ac,i}, c_{pc,k})$ and $d(s,\dot{s},\ddot{s}) = \min_{i \in \{1,\ldots,n_j\}, k \in \{1,\ldots,6\}} (d_{ac,i}, d_{pc,k})$. The variables $c_{ac,i}$, $d_{ac,i}$, $c_{pc,k}$, $d_{pc,k}$ are defined by,

$$c_{ac,i} = \frac{-|R_{i,\underline{q}}\dddot{q}_{i,max} + Z_i|}{|P_{i,s}(s)|}, \tag{25}$$

$$d_{ac,i} = \frac{|R_{i,\underline{q}}\dddot{q}_{i,max} + Z_i|}{|P_{i,s}(s)|}, \; i = 1, \ldots, n_j \tag{26}$$

$$c_{pc,k} = \frac{-(\dddot{X}_{k,max} - |\underline{P}_{sss}(s)\dot{s}^3 + 3\underline{P}_{ss}(s)\dot{s}\ddot{s}|_k)}{|P_{k,s}(s)|}, \tag{27}$$

$$d_{pc,k} = \frac{(\dddot{X}_{k,max} - |\underline{P}_{sss}(s)\dot{s}^3 + 3\underline{P}_{ss}(s)\dot{s}\ddot{s}|_k)}{|P_{k,s}(s)|}, \; k = 1, \ldots, 6, \tag{28}$$

where $\underline{Z}_i$ is defined by (15).

3 TIME OPTIMAL TRAJECTORY GENERATION

In principle, the problem of path-constrained motion planning is to find a function $s(t)$ satisfying the constraints, as defined in the previous section, and the requirement that the motion is time-optimal. This problem can be written as a Time-optimal Control (TOC) problem as shown by Constantinescu and

Croft [7]. The system can be formulated as

$$\dot{\underline{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \underline{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u, \tag{29}$$

with the states of the system defined as $\underline{x} = [s,\ \dot{s},\ \ddot{s}]^T$, and input $u = \dddot{s}$, and fulfilling the performance criterion $\min T = \min \int_{t_0}^{t_f} dt$. The problem is now to find input u such that T is minimized, subject to the dynamics defined by (29), with boundary conditions, $\underline{x}_0 = [s_0,\ \dot{s}_0,\ \ddot{s}_0]$ and $\underline{x}_f = [s_f,\ \dot{s}_f,\ \ddot{s}_f]$, while respecting the constraints on the states $\underline{x}$ (i.e. velocity and acceleration constraints), and on the control saturation levels $u_{min} \leq u \leq u_{max}$ (i.e. jerk constraints), both derived in Section 2. In order to solve the problem, an optimization problem needs to be solved. Here, instead of solving the time-optimal control problem directly, we opt to solve the trajectory planning problem in the $(s,\dot{s})-$plane. The same approach as presented by Constantinescu and Croft [7] is followed, however a different optimization algorithm is applied. The goal of the optimization strategy is to find a set of points in the $(s,\dot{s})-$plane that are interpolated using cubic splines. Cubic splines are chosen because these represent the lowest degree polynomials which results in a smooth motion, i.e. continuous and differentiable everywhere [7]. The points that are interpolated using cubic splines are the switching points in the case that no jerk constraints (or torque rate constraints in case of Constantinescu and Croft) are taken into account. In comparison, the algorithm used by Constantinescu and Croft is the Flexible Tolerance method [14] that is based on the Nelder-Mead method [15]. A disadvantage of this optimization algorithm is that it can converge to local minima in case of non-convex optimization problems. Constantinescu and Croft try to overcome this disadvantage by running the optimization four times. The best solution is then used as initial estimation for a final optimization run. However, this approach still does not guarantee that a global optimum is found. Since it is not guaranteed that the remaining solution space of the trajectory planning problem is convex, the obtained solution may represent a local minimum. Therefore, here a different approach is proposed.

An overview of the optimization strategy is depicted in Figure 1. As can be seen from this figure, the chosen optimization strategy is a hybrid optimization strategy that determines points in the $(s,\dot{s})-$plane which are interpolated using cubic splines. Such strategy is chosen since the solution space may be non-convex. A global optimization strategy is proposed. Since in general the outcome of a global optimization strategy does not give the optimum solution, the global optimization strategy is followed by a local optimization strategy to obtain a more accurate optimum solution.

The optimization problem is defined as

$$\min\ f(\underline{O}),\ \underline{O} \in \mathbb{R}^n, \tag{30}$$

where $\underline{O}$ represents the set of optimization variables such that fitness function $f(\underline{O})$ is minimized. The Fitness function determines the total motion time. The input to the function are optimization variables $\underline{O}$. The optimization variables are interpolated using cubic splines. First, it is checked whether the constraints on pseudo-velocity, -acceleration and -jerk are violated. When a constraint is violated the motion time for the set of optimization variables is set to, $T = 1 \cdot 10^{99}$, with T the total motion time. The optimization variables to this optimization problem are the parameters for determining a cubic spline. The variables are given as

$$\underline{O} = \left[\left(\frac{d\dot{s}}{ds}\right)_0, \dot{s}_1, \ldots, \dot{s}_p, \left(\frac{d\dot{s}}{ds}\right)_f \right]. \tag{31}$$

Herein, p indicates the number of intermediate switching points. As can be seen from the optimization variables, here a directional cubic spline is used instead of a natural cubic spline. A visualization of the optimization variables for a case with $p = 2$ can be found in Figure 2. The variables $\left(\frac{d\dot{s}}{ds}\right)_0$ and $\left(\frac{d\dot{s}}{ds}\right)_f$ represent the slope of the directional cubic spline at the start ($s = s_0$) and end point ($s = s_f$), respectively. The slopes are part of the optimization variables for two reasons. First, often a begin and/or end pseudo-velocity is required, so these pseudo-velocities cannot be part of the set of optimization variables. Second, the start and end slopes represent the velocity at which the actuator torques leave or approach the static equilibrium values. Therefore, steeper slopes represent faster motions [7].

The intermediate knotpoint positions for cubic spline interpolation, are determined using a strategy discussed by Shin and McKay [2] and Pfeiffer and Johanni [3]. The knotpoints could also be included as optimization parameters in the global optimization algorithm. However, this will increase the computational time and is therefore not considered.

After the intermediate points are determined, the global optimization strategy is performed. Several global optimization algorithms are available. Here the global optimization method suggested by Sim et al. [12], namely a genetic algorithm approach, is used. Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics [16], and are often used in solving path planning problems see e.g. [17, 18].Genetic algorithms are in principle composed of three operators, namely reproduction (or sometimes called selection), crossover and mutation. During reproduction, a selection out of an initial population of solutions is made. Once a solution is selected for reproduction, it enters a tentative new population.

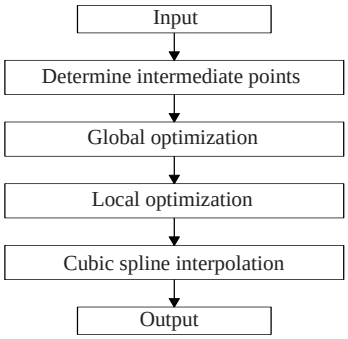


Figure 1: Optimization strategy to solve the path-constrained trajectory planning. problem

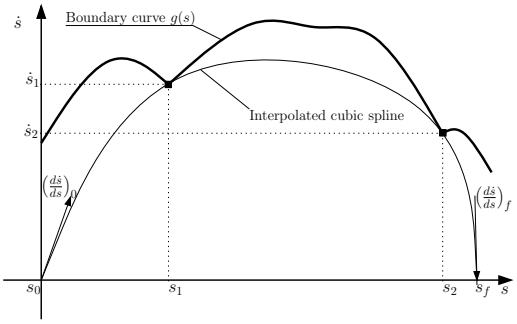


Figure 2: Visualization of optimization strategy.

Crossover proceeds by randomly mating two tentative solutions Each of the mated solutions undergoes a crossover by exchanging solution substrings of two mated solutions. Finally, mutation is a random walk through the string space. When a solution string is binary coded, this means changing a 1 to a 0 and vice versa.

The outcome of the global optimization strategy is a set of optimization variables which in general will not result in the optimum solution. This fact and in order to speed up the optimization process, an optimization algorithm is used that is suitable for local optimization problems. Here, the Nelder-Mead method is chosen [15]. The Nelder-Mead method is chosen above the Steepest-Descent and line search methods. In comparison the latter methods are computationally more efficient, however they show slow convergence near the optimum solution.

The outcome of the local optimization strategy is interpolated using cubic splines, see Figure 1. This completes the optimization strategy.

4 SIMULATIONS

The strategy for determining path-constrained time optimal motions with kinematic constraints presented above is validated by simulations. The strategy is compared with a dynamically constrained, path-constrained motion planning strategy using torque and torque-rate constraints (for which a dynamic model for the manipulator is needed). For each approach, i.e. using kinematic or dynamic constraints, the optimization strategy presented in Section 3 is used to generate trajectories. The parameters for the genetic algorithm optimization can be found in Table 1. The parameter settings are verified by performing reproducibility simulations (i.e. the parameters of the optimization strategy are chosen such that the outcome of the optimization reproduces). In simulations only actuator constraints will be taken into account, although with the developed strategy it is also possible to account for process constraints. For simulation purposes we consider a PUMA 560 manipulator, see Figure 3. This type of robot resembles a typical industrial manipulator. Here, we will only consider positional degrees of freedom, so the wrist (i.e. joints 4, 5 and 6) is fixed. The dynamic model of the robot and its parameters are given in the appendix (For a 6 DOF dynamic model of the PUMA 560 manipulator we refer to [19]). Friction in the joints is modeled using Coulomb ($T_{c,i}$ for joint i) and viscous friction ($T_{v,i}$ for joint i), with related parameters $T_c = [6.3,\ 5.5,\ 2.6]\ Nm$ and $T_v = 0.1T_c$ (in Nms). The robot is controlled using independent joint PD-control with gravity compensation where the parameters ($K_{p,i}$ and $K_{v,i}$ for joint i), with $K_p = [700,\ 600,\ 340]\ Nm/rad$ and $K_v = [70,\ 102,\ 27]\ Nms/rad$, by tuning for a rise time of 0.2 seconds and an overshoot of maximally 5%. The simulations are performed using

Table 1: Genetic algorithm parameters.

Parameter	Value
Population size	500
Maximum number of generations	60
Reproduction	Uniform selection
Crossover rate	80%
Mutation rate	25%
Mutation distribution	Gaussian

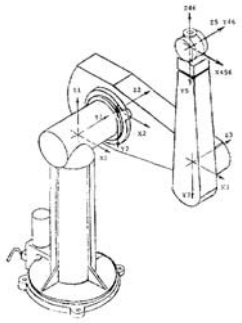


Figure 3: The PUMA 560 manipulator in zero position with attached coordinate frames using modified DH-convention, [19].

MATLAB/Simulink [20].

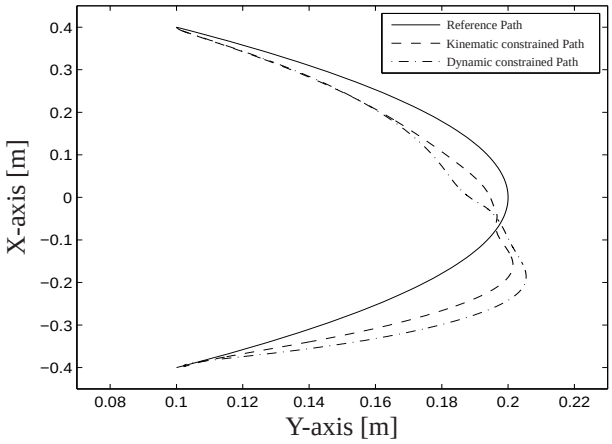
Three simulation case studies are considered, where jerk limits (in case of kinematic constraints) and torque-rate limits (in case of dynamic constraints) are related to acceleration and torque constraints respectively, i.e. $\dddot{q}_{max} = c\ddot{q}_{max}$ and $\dot{T}_{max} = cT_{max}$ with $c = 100$ (case 1), $c = 10$ (case 2) and $c = 1$ (case 3). The kinematic velocity and acceleration constraints are determined by performing simulations, while regarding the torque saturation level and the response of the robotic system and are defined as $\dot{q}_{max} = [3.84,\ 2.81,\ 9.41]\ rad/s$ and $\ddot{q}_{max} = [24.5,\ 32.5,\ 76]\ rad/s^2$. In practice, the constraints are typically given by an experienced robot-user or are obtained by performing experiments. The torque constraints are taken from [19]. Jerk and torque-rate constraints are chosen arbitrarily to smoothen the motion.

During simulation we want the manipulator to move from $P_0 = [0.4,\ 0.1,\ 0.42]^T$ to $P_e = [-0.4,\ 0.1,\ -0.25]^T$ defined in the workspace of the manipulator. This cannot be realized by a straight line segment due to singular configurations of the manipulator along this line. Therefore the path to be followed during simulation is a parabolic path and is given as

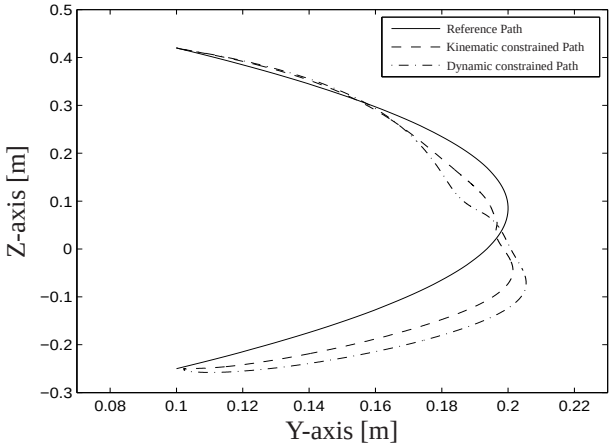
$$\underline{P}(s) = \begin{bmatrix} 0.4 - 0.8s \\ -0.4(s-0.5)^2 + 0.2 \\ 0.42 - 0.67s \end{bmatrix},\ s \in [0,1]. \qquad (32)$$

The simulation results for the three simulation cases are depicted in Figures 4, 5, 6 and Table 2. In these figures, the solid lines represent the reference path, the dashed lines the simulated controller tracked path while considering kinematic constraints (i.e. on velocity, acceleration and jerk) and the dash-dotted lines the simulated controller tracked path while considering dynamic constraints (i.e. on torque and torque-rate). The tracked path deviates from the reference path due to the modeled joint friction. Furthermore, some noise on the torque profiles is seen. This can be ascribed to the fact that the controller uses a numerical derivative of the reference signal in the D-action. From Table 2 we can see that for decreasing jerk/torque-rate limits, the motion time and path following accuracy increases. This is due to the smoothing effect of the jerk/torque-rate limits on the resulting torque profiles, see Figures 4c, 5c and 6c. Regarding the computational time of a trajectory it can be said that recently Behzadipour and Khajepour [21] concluded that the computational time of time-optimal trajectories along predefined paths is more than 1000 times larger than the motion time. This research supports that conclusion. So, despite many research efforts, determining path-constrained motions still costs a tremendous amount of computational effort.

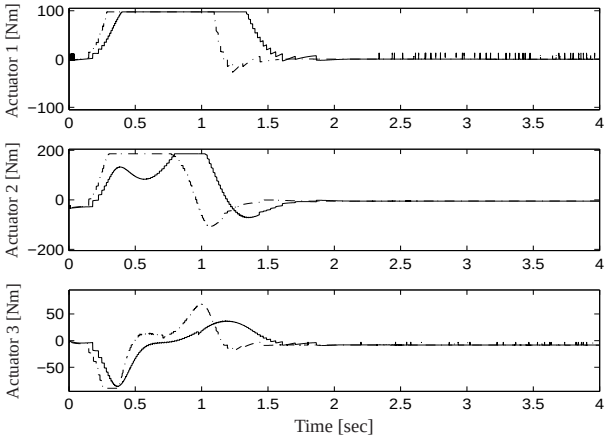
When comparing simulation results of the kinematically constrained path and the dynamically constrained path, we can see that (except for simulation case 3) the motion time for the



(a) Y-X plot of path in manipulator workspace.

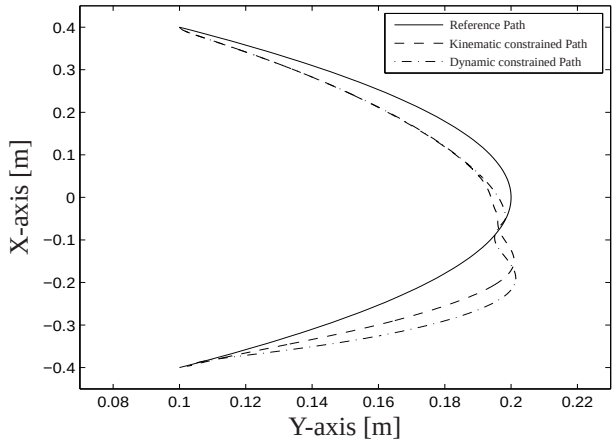


(b) Y-Z plot of path in manipulator workspace.

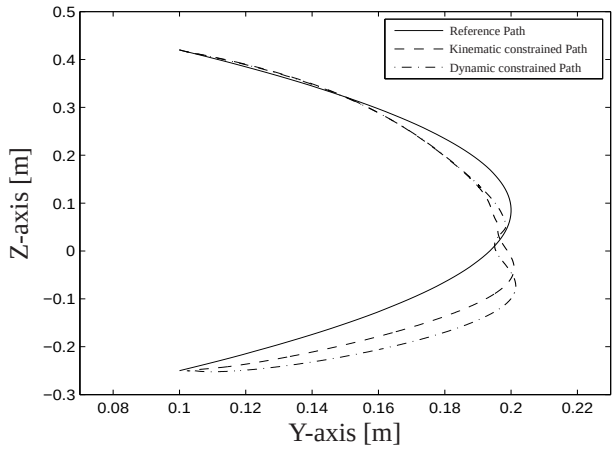


(c) Actuator torques; Solid lines represent kinematically constrained torques; Dash-dotted lines represent dynamically constrained torques.

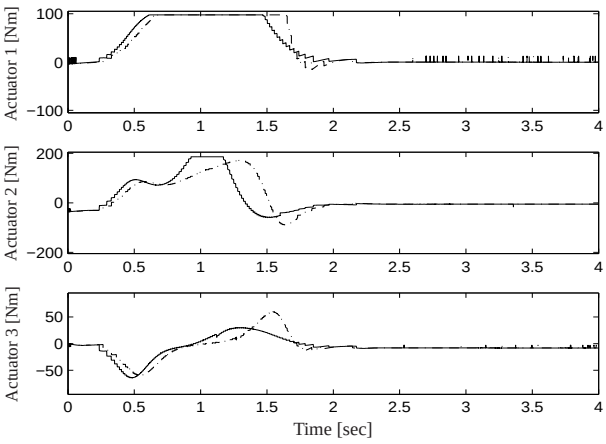
Figure 4: Simulation results for simulation case 1.



(a) Y-X plot of path in manipulator workspace.

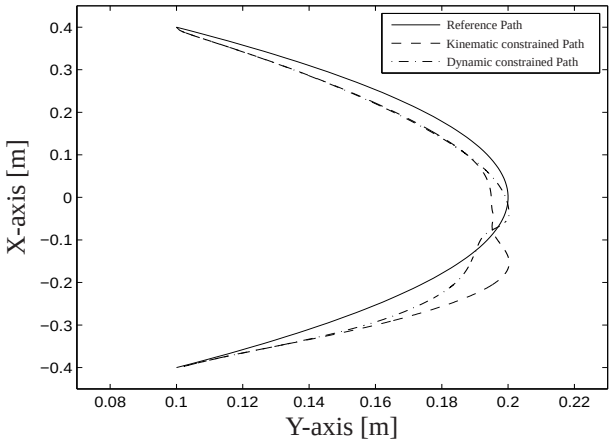


(b) Y-Z plot of path in manipulator workspace.

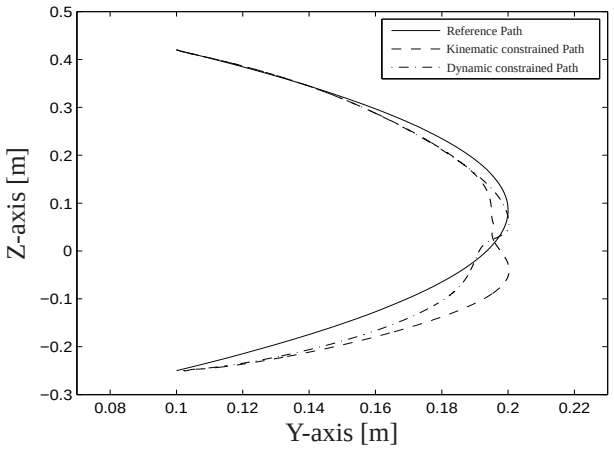


(c) Actuator torques; Solid lines represent kinematically constrained torques; Dash-dotted lines represent dynamically constrained torques.

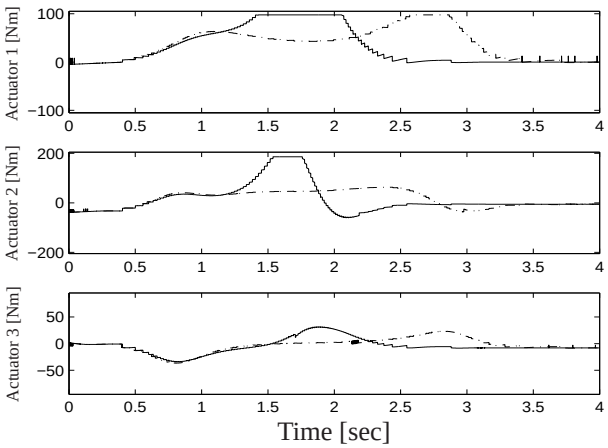
Figure 5: Simulation results for simulation case 2.



(a) Y-X plot of path in manipulator workspace.



(b) Y-Z plot of path in manipulator workspace.



(c) Actuator torques; Solid lines represent kinematically constrained torques; Dash-dotted lines represent dynamically constrained torques.

Figure 6: Simulation results for simulation case 3.

Table 2: Simulation data.

	Kinematically constrained path		
Simulation Case	1	2	3
Total motion time [sec]	1.8643	2.1737	2.881
Path following accuracy q_1 [rad]	0.1326	0.1284	0.1229
Path following accuracy q_2 [rad]	0.1098	0.1091	0.1015
Path following accuracy q_3 [rad]	0.0633	0.0581	0.0514
Settling error q_1 [mrad]	9.7	9.8	10.1
Settling error q_2 [mrad]	-6.8	7.1	-7.3
Settling error q_3 [mrad]	5.7	6.0	6.6

(a) Simulation results for kinematically constrained path.

	Dynamically constrained path		
Simulation Case	1	2	3
Total motion time [sec]	1.440	2.145	3.732
Path following accuracy q_1 [rad]	0.1523	0.1316	0.0922
Path following accuracy q_2 [rad]	0.1212	0.0993	0.0729
Path following accuracy q_3 [rad]	0.0755	0.0628	0.0503
Settling error q_1 [mrad]	9.7	9.9	10.5
Settling error q_2 [mrad]	-6.4	-6.8	-7.5
Settling error q_3 [mrad]	5.4	6.3	7.0

(b) Simulation results for dynamically constrained path.

dynamically constrained path is somewhat smaller than for the kinematically constrained path. This is due to the fact that kinematically constraints are constant along the entire path and therefore the motion can be somewhat conservative from a time-optimality point of view. The fact that this is not the case for case 3 can be related to the formulation of the constraints as described in Section 2. Herein, the influence of jerk limits on pseudo-velocity and pseudo-acceleration is neglected. Since simulation case 3 for kinematically constrained approach does not result in a conservative solution in comparison to the dynamic constraint approach, it seems that this assumption does not hold for relatively small jerk limits.

From the discussion above it can be concluded that, by using kinematic constraints, motions are determined that give comparable results to motions determined using dynamic constraints. The 'tuning' of the kinematic constraints may take some time; however the industrial applicability of a kinematic constraint approach is highly increased in comparison with a dynamic constraint approach when accounting for the time and expertise needed to build a reliable dynamic model.

5 CONCLUSIONS AND RECOMMENDATIONS

In this paper, a method is presented for determining time-optimal trajectories for industrial manipulators along a pre-defined path. Instead of taking into account dynamic equations with actuator torque limits and possible torque-rate limits of a manipulator, we only require knowledge on the kinematics of a manipulator together with actuator velocity, acceleration and jerk limits. The industrial applicability is highly increased, firstly, due to the fact that it uses kinematic relations of a manipulator (which are far easier to obtain than the dynamic equations) and secondly due to the fact that constraints acting on the process can be taken into account. By transforming the constraints to one domain, a time-optimal control problem can be formulated. However, no analytic solution for this kind of time-optimal control problem is available to this date. The proposed optimization algorithm consists of a combination of a genetic algorithm and the Nelder-Mead method. In this way a global optimum, for a path constrained motion with a minimum time criterion, can be found. From simulations on a Puma 560 manipulator model, it can be concluded that, for a proper choice of the kinematic constraints, results can be obtained that are comparable to those obtained using the dynamic constraint approach. However, where the dynamic constraint approach asks for a specific knowledge on multi-body dynamics, the 'tuning' of the kinematic constraints may take some time.

For further research it is recommended to reduce the computational time of the optimization strategy such that the industrial applicability is increased even more. Furthermore, as observed in the simulation case with relatively low jerk limits, the influence of jerk limitations on pseudo-velocity and pseudo-acceleration cannot be neglected. Therefore, the constraint formulation should be extended such that jerk influence on pseudo-velocity and pseudo-acceleration is accounted for. Moreover, for further performance evaluations, experiments on industrial applications with challenging mechanisms, such as a robotic system of double-scara type, should be performed. Finally, it can be interesting to apply the developed algorithm on applications other than industrial manipulators, for example, a mobile robotic system, with non-holonomic constraints.

A DYNAMIC MODEL OF PUMA560 WITH THREE DEGREES OF FREEDOM

The dynamic model of the Puma560 is of the following form; $\underline{M}(\underline{q})\ddot{\underline{q}} + \dot{\underline{q}}^T \underline{C}(\underline{q})\dot{\underline{q}} + \underline{G}(\underline{q}) = \underline{T}$. Herein, $\underline{q} \in \mathbb{R}^n$ represents the vector of joint positions, $\underline{T} \in \mathbb{R}^n$ the vector of actuator torques, $\underline{M}(\underline{q}) \in \mathbb{R}^{n\times n}$ the inertia matrix of the manipulator, $\underline{C}(\underline{q}) \in \mathbb{R}^{n\times n\times n}$ the third-order tensor representing centrifugal and Coriolis terms, $\underline{G}(\underline{q}) \in \mathbb{R}^n$ the gravity vector and $\dot{}$ denotes the derivative with respect to time.

For the three-degree-of-freedom model used here $n = 3$. The nonzero elements of the inertia matrix (in kgm^2) are given as

$$M_{11} = 2.57 + 1.38c_{q_2}c_{q_2} + 0.3s_{q_2+q_3}s_{q_2+q_3} + 0.744c_{q_2}s_{q_2+q_3} \tag{33}$$

$$M_{12} = 0.69s_{q_2} - 0.134c_{q_2+q_3} + 0.0238c_{q_2} \quad (34)$$
$$M_{13} = -0.134 + c_{q_2+q_3} - 3.97 \cdot 10^{-3} s_{q_2+q_3} \quad (35)$$
$$M_{21} = M_{12} \quad (36)$$
$$M_{22} = 6.79 + 0.744s_{q_3} \quad (37)$$
$$M_{23} = 0.33 + 0.372s_{q_3} - 0.011c_{q_3} \quad (38)$$
$$M_{31} = M_{13} \quad (39)$$
$$M_{32} = M_{23} \quad (40)$$
$$M_{33} = 1.16, \quad (41)$$

where c_q represents $\cos(q)$ and s_q represents $\sin(q)$. The nonzero Christoffel symbols (in Nms^2) are given as

$$C_{112} = -2.76s_{q_2}c_{q_2} + 0.744c_{2q_2+q_3} + 0.6s_{q_2+q_3}c_{q_2+q_3} - 0.0213(1 - 2s_{q_2+q_3}s_{q_2+q_3}) \quad (42)$$
$$C_{113} = 0.744c_{q_2}c_{q_2+q_3} + 0.6s_{q_2+q_3}c_{q_2+q_3} + 0.022c_{q_2}s_{q_2+q_3} - 0.0213(1 - 2s_{q_2+q_3}s_{q_2+q_3}) \quad (43)$$
$$C_{122} = 0.69c_{q_2} + 0.134s_{q_2+q_3} - 0.0238s_{q_2} \quad (44)$$
$$C_{123} = 0.267s_{q_2+q_3} - 7.58 \cdot 10^{-3} c_{q2+q3} \quad (45)$$
$$C_{133} = 0.5C_{123} \quad (46)$$
$$C_{211} = -0.5C_{112} \quad (47)$$
$$C_{223} = 0.022s_{q_3} + 0.744c_{q_3} \quad (48)$$
$$C_{233} = 0.5C_{223} \quad (49)$$
$$C_{311} = -0.5C_{113} \quad (50)$$
$$C_{322} = -C_{233}, \quad (51)$$

and nonzero gravity vector entries (in Nm) are given below

$$G_2 = -37.2c_{q_2} - 8.4s_{q_2+q_3} + 1.02s_{q_2} \quad (52)$$
$$G_3 = -8.4s_{q_2+q_3} + 0.25c_{q_2+q_3}. \quad (53)$$

REFERENCES

[1] Bobrow, J. E., Dubowsky, S., and Gibson, J. S., 1985. "Time-optimal control of robotic manipulators along specified paths". *Int. J. Robotic Research*, **4**, pp. 3–17.

[2] Shin, K. G., and McKay, N. D., 1985. "Minimum-time control of robotic manipulators with geometric path constraints". *IEEE Trans. Automatic Control*, **30**(6), pp. 531–541.

[3] Pfeiffer, F., and Johanni, R., 1987. "A concept for manipulator trajectory planning". *IEEE J. Robotics and Automation*, **3**(2), pp. 115–123.

[4] Craig, J. J., 1986. *Introduction to robotics*. New York: Addision-Wesley.

[5] Pietsch, I. T., Becker, O., Krefft, M., and Hesselbach, J., 2003. "Time-optimal trajectory planning for adaptive control of plane parallel robots". In IEEE Int. Conf. Control and Automation, pp. 639–643.

[6] Shiller, Z., 1994. "Time-energy optimal control of articulated systems with geometric path constraints". In IEEE Int. Conf. Robotics and Automation, pp. 2680–2685.

[7] Constantinescu, D., and Croft, E. A., 2000. "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths". *J. Rob. Syst.*, **17**(5), pp. 233–249.

[8] Žlajpah, L., 1996. "On time optimal path control of manipulators with bounded joint velocities and torques". In IEEE Int. Conf. Robotics and Automation, pp. 1572–1577.

[9] Dong, J., and Stori, J. A., 2006. "A generalized time-optimal bidirectional scan algorithm for constrained feedrate optimization". *ASME J. Dyn. Syst., Meas.,Control,*, **128**(2), pp. 379–390.

[10] Athans, M., and Falb, P. L., 1966. *Optimal control: an introduction to the theory and its applications*. London: McGraw-Hill.

[11] Penev, B. G., and Christov, N. D., 2005. "A fast time-optimal control synthesis algorithm for a class of linear systems". In Proc. of the ACC, pp. 883–888.

[12] Sim, Y. C., Leng, S. B., and Subramaniam, V., 2000. "A combined genetic algorithms-shooting method approach to solving optimal control problems". *Int. J. of Syst. Science*, **31**(1), pp. 83–89.

[13] Chettibi, T., 2006. "Synthesis of dynamic motions for robotic manipulators with geometric path constraints". *Mechatronics*, **16**(9), pp. 547–563.

[14] Paviani, D., and Himmelblau, D. M., 1969. "Constrained nonlinear optimization by heuristic programming". *Operations Research*, **17**, pp. 872–882.

[15] Nelder, J. A., and Mead, R., 1965. "A simplex method for function minimization". *Computer J.*, **7**(4), pp. 308–313.

[16] Goldberg, D. E., 1989. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Amsterdam.

[17] Davidor, Y., 1990. "Robot motion planning using a genetic algorithm". In IEEE Int. Conf. on Comp. Syst. and Software Eng., pp. 186–191.

[18] Ashiru, I., and Czarnecki, C., 1995. "Optimal motion planning for mobile robots using genetic algorithms". In IEEE Int. Conf. on Indus. Automation and Control, pp. 297–300.

[19] Armstrong, B., Khatib, O., and J.Burdick, 1986. "The explicit dynamic model and inertial parameters of the puma 560 arm". *Int. Conf. on Robotics and Automation*, **3**, pp. 510–518.

[20] The Mathworks Inc., 2006. *Matlab User's Guide*. ver. 7.

[21] Behzadipour, S., and Khajepour, A., 2006. "Time-optimal trajectory planning in cable-based manipulators". *IEEE Trans. on Robotics*, **22**(3), pp. 559–563.