Optimal Robot Control with Unspecified Initial and Final Conditions

Josip Kasac, Branko Novakovic

Abstract— **This paper presents the derivation of the numerical algorithm for optimal control of nonlinear multivariable systems with control and state vectors constraints and with unspecified initial and final conditions. The initial and final conditions depend on the parameter vector which represents a type of coordinate transformation. It is presented a detailed derivation of the algorithm for the calculation of the parameter vector, while the algorithm for time optimal control is described briefly. The algorithm derivation is based on the backpropagation-through-time algorithm, i.e. the chain rule for ordered derivations, and it is not based on Lagrange multiplier techniques and the calculus of variations. The derived algorithm is used for the solution of the minimum-time control of the robot with two degrees of freedom and with control vector constraint. The problem discussed in this paper is the determination of the control and parameter vector for the transition of the robot state from the initial to the final one in minimum time.**

Keywords— **Optimal Control, Robot Control, Minimum-time control, Backpropagation Through Time Algorithm, Nonlinear Systems.**

I. Introduction

The common formulation of the minimum-time robot control includes optimization in relation to the control vector and minimum time. The solution of the minimum time control problem, generally speaking, is not in the form of the bang-bang solution. Usually, some components of the control vector are in the form of the bang-bang solution, while other components are in the form of the continuous-time solution in the limits defined by control vector constraints. In other words, with control vector constraints and without full energy constraint, only the bang-bang solution of the all control vector components guarantees fully usable minimum-time control.

In robot control, one possibility to achieve the bang-bang solution of the control vector is the appropriate coordinate transformations of the robot placement. Coordinate transformations of the robot placement means parameter transformations of the initial and final condition of the optimal control problem formulation. In other words, by introducing coordinate transformations, the optimal control problem with fixed initial and final conditions becomes the optimal control problem with unspecified initial and final conditions.

As is already known from the classical optimal control theory, [1], the solution of the optimal control problem requires the solution of the first-order stationary conditions, or Euler-Lagrange equations. This is a two-point boundary-value problem and is difficult to solve numerically. Various methods of numerical solution of the optimal control problem have therefore been developed, such as gradient algorithms [1], reduction of the optimal control problem to a nonlinear programming one [2], discrete dynamic programming, quasi-linearisation algorithms [3], discrete time maximum principle [4], etc.

A good compromise between the accuracy of the solution and speed of convergence toward the optimal solution on the one hand and the computational cost and algorithm complexity on the other hand is to use the gradient algorithm [5]-[8] for the nonlinear programming representation of the optimal control problem [2], [9].

A standard method for reducing the optimal control problem to a nonlinear one is adding the penalty function for plant equation constraints to the cost function, and optimising the total cost function according to the control and state vector. The problem formulated in this way is numerically very unstable and has slow convergence (due to additional equality constraints for plant equation). However, the problem can be avoided by using the backpropagation through-time (BPTT) algorithm.

The BPTT algorithm, [10]-[12], is time generalization of the backpropagation algorithm (BP), in case when the error which is minimised is given along the specified time interval. The essence of the BP algorithm is a simple and precise calculation of derivations of the cost function in relation to system parameters. The BPTT algorithm expands this method through application to dynamic systems for which direct calculation of derivations can be very complex. A solution to this problem lies in the chain rule for ordered derivations [8], [10], [11], which results in error backpropagation, i.e. parameter adjustment backward in time.

II. Optimal Control Problem Formulation

A. Continuous Optimal Control Problem

We consider a nonlinear continuous optimal control problem with a fixed beginning and terminal time, which consists of choosing control vector $\mathbf{u}(t)$ and parameter vector $\mathbf{p}$ to minimize the cost function

$$J_0 = \min_{\mathbf{u}(t),\mathbf{p}} \int_{t_0}^{t_f} \hat{F}(\mathbf{x}(t), \mathbf{u}(t))\, dt, \tag{1}$$

subject to the constraints defined by the plant equations

$$\dot{\mathbf{x}}(t) = \hat{\mathbf{f}}(\mathbf{x}(t), \mathbf{u}(t)), \tag{2}$$

J. Kasac is with the Institute for Defense Studies, Research and Development, Bijenicka cesta 46, 10000 Zagreb, Croatia, e-mail: jkasac@lycos.com .

B. Novakovic is with the Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Luciceva 5, 10000 Zagreb, Croatia, e-mail: branko.novakovic@fsb.hr .

and the initial and final condition of the state vector

$$\mathbf{x}(t_0) = \mathbf{x}_0(\mathbf{p}), \qquad \mathbf{x}(t_f) = \mathbf{x}_f(\mathbf{p}), \tag{3}$$

which depend on the parameter vector, subject to the control and state vector inequality constraints

$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \geq \mathbf{0}, \tag{4}$$

the parameter vector inequality constraints

$$\mathbf{r}(\mathbf{p}) \geq \mathbf{0}, \tag{5}$$

and the control and state vector equality constraints

$$\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{0}, \tag{6}$$

where $\mathbf{x}(t)$ is n-dimensional state vector, $\mathbf{u}(t)$ is m-dimensional control vector, $\mathbf{p}$ is N_p-dimensional parameter vector, $\mathbf{g}(\cdot)$ is N_g-dimensional vector function of inequality constraints, $\mathbf{r}(\cdot)$ is N_r-dimensional vector function of parameter vector inequality constraints, and $\mathbf{h}(\cdot)$ is N_h-dimensional vector function of equality constraints.

B. Time Discretization and Penalty Method Approach

The next step is time discretization of the problem (1) to (6). The result is

$$J_0 = \min_{\mathbf{u}(i),\mathbf{p}} \tau \sum_{i=0}^{N-1} \hat{F}(\mathbf{x}(i), \mathbf{u}(i)), \tag{7}$$

$$\mathbf{x}(i+1) = \mathbf{f}(\mathbf{x}(i), \mathbf{u}(i)), \tag{8}$$

$$\mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}), \qquad \mathbf{x}(N) = \mathbf{x}_f(\mathbf{p}), \tag{9}$$

$$\mathbf{g}(\mathbf{x}(i), \mathbf{u}(i)) \geq \mathbf{0}, \tag{10}$$

$$\mathbf{r}(\mathbf{p}) \geq \mathbf{0}, \tag{11}$$

$$\mathbf{h}(\mathbf{x}(i), \mathbf{u}(i)) = \mathbf{0}, \tag{12}$$

for $i = 0, 1, ..., N-1$, where N is the number of sampling intervals, $\tau = (t_f - t_0)/N$ is the sampling interval, while $\mathbf{x}(i) \equiv \mathbf{x}(t_0 + i\tau)$,

$$\mathbf{f}(\mathbf{x}(i), \mathbf{u}(i)) = \mathbf{x}(i) + \tau\hat{\mathbf{f}}(\mathbf{x}(i), \mathbf{u}(i)).$$

The next step is the expansion of the cost function (7) by adding penalty functions for constraints

$$J = J_0 + J_1 + J_2 + J_3 + J_4, \tag{13}$$

where

$$J_1 = K_B \sum_{k=1}^{n} (x_k(N) - x_k(t_f))^2, \tag{14}$$

is the penalty function for the final boundary condition, where K_B is the coefficient of the penalty function and $x_k(t_f)$ is the k-th component of the state vector in the terminal time. Further,

$$J_2 = \hat{K}_V \sum_{i=0}^{N-1} \sum_{k=1}^{N_g} g_k^2(\mathbf{x}(i), \mathbf{u}(i)) H^-(g_k(\mathbf{x}(i), \mathbf{u}(i))), \tag{15}$$

is the penalty function for inequality constraints (10), where $H^-(g_k)$ is Heaviside step function defined as follows

$$H^-(g_k) = \begin{cases} 0, & \text{if} \quad g_k \geq 0 \\ 1, & \text{if} \quad g_k < 0 \end{cases} \tag{16}$$

while $\hat{K}_V$ is the coefficient of the penalty function for inequality constraints.

The penalty function for parameter vector inequality constraints (11) is

$$J_3 = \hat{K}_P \sum_{k=1}^{N_r} r_k^2(\mathbf{p}) H^-(r_k(\mathbf{p})). \tag{17}$$

Finally, there is the penalty function for equality constraints (12)

$$J_4 = \hat{K}_E \sum_{i=0}^{N-1} \sum_{k=1}^{N_h} h_k^2(\mathbf{x}(i), \mathbf{u}(i)), \tag{18}$$

where $\hat{K}_E$ is the coefficient of the penalty function of equality constraints.

If $\hat{K}_V = \tau K_V$ and $\hat{K}_E = \tau K_E$ take in account then the equation (13) can be expressed as

$$J = \tau \sum_{i=0}^{N-1} F(\mathbf{x}(i), \mathbf{u}(i)) + J_1 + J_3, \tag{19}$$

where

$$\begin{aligned} F(\mathbf{x}(i), \mathbf{u}(i)) &= \hat{F}(\mathbf{x}(i), \mathbf{u}(i)) + \\ &+ K_V \sum_{k=1}^{N_g} g_k^2(\mathbf{x}(i), \mathbf{u}(i)) H^-(g_k(\mathbf{x}(i), \mathbf{u}(i)) + \\ &+ K_E \sum_{k=1}^{N_h} h_k^2(\mathbf{x}(i), \mathbf{u}(i)). \end{aligned} \tag{20}$$

On this way the problem (7) to (12) can be expressed in the following form

$$J = \min_{\mathbf{u}(i),\mathbf{p}} \left(\tau \sum_{i=0}^{N-1} F(\mathbf{x}(i), \mathbf{u}(i)) + J_1 + J_3 \right), \tag{21}$$

$$\mathbf{x}(i+1) = \mathbf{f}(\mathbf{x}(i), \mathbf{u}(i)), \qquad \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}). \tag{22}$$

C. Gradient Algorithm for Parameter Vector

The gradient descent algorithm according to the control vector is given as follows:

$$\mathbf{u}^{(l+1)}(i) = \mathbf{u}^{(l)}(i) - \eta_{\mathrm{u}} \frac{\partial J}{\partial \mathbf{u}^{(l)}(i)} \tag{23}$$

$$\mathbf{p}^{(l+1)} = \mathbf{p}^{(l)} - \eta_{\mathrm{p}} \frac{\partial J}{\partial \mathbf{p}^{(l)}} \tag{24}$$

where $i = 0, 1, \ldots, N-1$, $l = 1, 2, \ldots, M$, while η_{u} and η_{p} are the convergence coefficients, index l represents the

l-th iteration of the gradient algorithm, M is the number of iterations of the gradient algorithm.

The algorithm for the calculation of the optimal control vector on the basis of expressions (21), (22) and (23) with fixed initial and final conditions is considered in more detail in [13]. The basic characteristic of this algorithm is derivation without using the calculus of variations and Lagrange multiplier techniques. This fact provides an obvious geometrical interpretation of the gradient algorithm, which on the other hand, provides rough approximation of the gradient of the penalty function without losing accuracy of the optimal solution.

This paper deals with the extension of the above-mentioned algorithm for the case of optimal control with unspecified initial and final condition. This means that the initial and final state are parameterized with the parameter vector of a coordinate transformations. In other words, it will be shown the derivation of the optimal parameter vector on the basis of the gradient algorithm (24).

The gradient of the cost function (21) according to the parameter vector in l-th iteration of the gradient algorithm is

$$\frac{\partial J}{\partial p_k} = \tau \sum_{i=0}^{N-1} \frac{\partial F(i)}{\partial p_k} + \frac{\partial J_1}{\partial p_k} + \frac{\partial J_3}{\partial p_k}, \tag{25}$$

where $F(i) \equiv F(\mathbf{x}(i), \mathbf{u}(i))$.

The next step is to determine the partial derivatives in the sum on the right side of the expression (25). Function $F(i)$ depends directly on the components of the state vector in i-th time interval, which on the other hand depend indirectly on the parameter vector through equation (22). It follows that

$$\frac{\partial F(i)}{\partial p_k} = \sum_{r=1}^{n} \frac{\partial F(i)}{\partial x_r(i)} \frac{\partial x_r(i)}{\partial p_k}. \tag{26}$$

Furthermore, the calculation of partial derivatives $\frac{\partial x_r(i)}{\partial p_k}$ follows on the basis of (22),

$$\frac{\partial x_r(i)}{\partial p_k} = \sum_{q=1}^{n} \frac{\partial f_r(i-1)}{\partial x_q(i-1)} \frac{\partial x_q(i-1)}{\partial p_k}, \tag{27}$$

for $r = 1, 2, \ldots, n$, $k = 1, 2, \ldots, N_p$, $i = 0, 1, \ldots, N$, where $f_r(j) \equiv f_r(\mathbf{x}(j), \mathbf{u}(j))$.

The above-mentioned iterative expression is the chain rule for ordered derivatives and has the initial state $\frac{\partial x_q(0)}{\partial p_k}$, which is determined from the equation (9).

What remains is the calculation of the gradient of the penalty function J_1. Following the similar reasoning it is obtained

$$\frac{\partial J_1}{\partial p_k} = \sum_{r=1}^{n} \frac{\partial J_1}{\partial x_r(N)} \frac{\partial x_r(N)}{\partial p_k} + \sum_{r=1}^{n} \frac{\partial J_1}{\partial x_r(t_f)} \frac{\partial x_r(t_f)}{\partial p_k}, \tag{28}$$

where $\frac{\partial x_r(t_f)}{\partial p_k}$ follows from the equation (9).

By using equations (25) to (28), the matrix representation in Appendix I and the auxiliary variable

$$S_k(i) = \sum_{j=0}^{i} \frac{\partial F(j)}{\partial p_k} \tag{29}$$

it is obtained the following algorithm:

1. *Initialisation of the gradient algorithm.* For control vectors ($\mathbf{u}_0^{(0)}$, $\mathbf{u}_1^{(0)}$,..., $\mathbf{u}_{N-1}^{(0)}$) and parameter vector $\mathbf{p}^{(0)}$ we put arbitrary values, which can be outside of the allowed area defined by the constraints.

2. *Calculation of the state vectors*

$$\mathbf{x}^{(l)}(i+1) = \mathbf{f}(\mathbf{x}^{(l)}(i), \mathbf{u}^{(l)}(i)), \qquad \mathbf{x}^{(l)}(0) = \mathbf{x}_0(\mathbf{p}^{(l)})$$

for $i = 0, 1, \ldots, N-1$ in l-th iteration of the gradient algorithm.

3. *Calculation of the gradient*

$$\mathbf{J}_{\mathrm{p}} \equiv \frac{\partial J}{\partial \mathbf{p}^{(l)}}$$

for l-th iteration of the gradient algorithm.

3.1 *Initialisation* for $i = 0$.

$$\mathbf{S}(0) = \mathbf{X}_{\mathrm{p}}{}^{\mathrm{T}}(0) \cdot \mathbf{F}_{\mathrm{x}}(0), \tag{30}$$

3.2 *Iteration* for $i = 1$ to $i = N-1$.

$$\begin{aligned} \mathbf{X}_{\mathrm{p}}(i) &= \mathbf{X}(i-1) \cdot \mathbf{X}_{\mathrm{p}}(i-1) \\ \mathbf{F}_{\mathrm{p}}(i) &= \mathbf{X}_{\mathrm{p}}{}^{\mathrm{T}}(i) \cdot \mathbf{F}_{\mathrm{x}}(i), \\ \mathbf{S}(i) &= \mathbf{S}(i-1) + \mathbf{F}_{\mathrm{p}}(i) \end{aligned} \tag{31}$$

3.3 *Finalization* for $i = N$.

$$\begin{aligned} \mathbf{X}_{\mathrm{p}}(N) &= \mathbf{X}(N-1) \cdot \mathbf{X}_{\mathrm{p}}(N-1) \\ \mathbf{J}_{1\mathrm{p}} &= \mathbf{X}_{\mathrm{p}}{}^{\mathrm{T}}(N) \cdot \mathbf{J}_{1\mathrm{N}} + \mathbf{X}_{\mathrm{f}}{}^{\mathrm{T}} \cdot \mathbf{J}_{1\mathrm{f}} \\ \mathbf{J}_{\mathrm{p}} &= \tau \mathbf{S}(N-1) + \mathbf{J}_{1\mathrm{p}} + \mathbf{J}_{3\mathrm{p}} \end{aligned} \tag{32}$$

4. *Calculation of the new iteration of parametar vectors on the basis of the gradient algorithm*

$$\mathbf{p}^{(l+1)} = \mathbf{p}^{(l)} - \eta_{\mathrm{p}} \mathbf{J}_{\mathrm{p}}, \tag{33}$$

for $l = 0, 1, ..., M$. The index is shifted by one, $l \to l+1$ and go back to step two.

D. A Heuristic Approach to Time Optimal Control

The above-mentioned algorithm has been derived for the fixed terminal time t_f. This subsection describes briefly a heuristic approach to solving the problem of time optimal control (TOC), which is relatively effective in avoiding slow convergence and stability problems and which is characteristic for classical numerical methods in TOC. This method uses the characteristics of penalty functions for boundary conditions and constraints.

The minimum time, which is the solution of the TOC problem, can be marked with $t_{min} = N\tau_{min}$. The basic idea is to keep the previously obtained algorithm for calculating control vectors for the given constant sampling interval, so that along with calculating control vectors in every iteration of the gradient algorithm, the new value of the sampling interval is being calculated. To emphasize the variability of sampling interval τ we will hereinafter use the

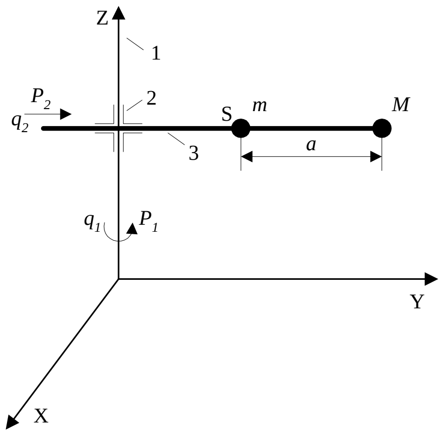

Fig. 1. Robotic manipulator with two degrees of freedom - rotation and translation.

symbol $\tau \to \tau^l$, which represents the sampling interval in l-th iteration of the gradient algorithm.

It is defined the new cost function as the sum of penalty functions depending on the variable sampling interval in l-th iteration of the gradient algorithm

$$J_T(\tau^l) = J_1 + J_2 + J_3 + J_4, \tag{34}$$

with the difference that the coefficients of penalty functions $\hat{K}_V$ and $\hat{K}_E$ are defined as constant and independent of sampling interval τ, so that the explicit dependence of function $J_T(\tau^l)$ on τ is avoided. Further, we introduce J_ε, which is the measure of accuracy of the solution of the TOC problem for the given τ. The lesser the value of the sum of penalty functions, the closer the solution to the optimum.

On the basis of the above-mentioned values, it has been employed the following heuristic algorithm

$$\tau^{l+1} = \tau^l - \Delta\tau H^- \left(J_T(\tau^l) - J_\varepsilon\right), \tag{35}$$

and initial condition $\tau^0 > \tau_{min}$. In other words, each time the condition $J_T(\tau^l) < J_\varepsilon$ is fulfilled, τ^l decreases by the constant value $\Delta\tau$. If function $J_T(\tau^l)$ doesn't reach the value of J_ε after sufficiently high number of iteration, this means that $\tau^l < \tau_{min}$, and $\tau_{min} \simeq \tau^{l-1}$ is taken as an approximation of the minimum-time.

This structure of the algorithm guarantees stability and convergence toward τ_{min}, because it does not change the value of τ^l until the value of function J_T falls below the given, sufficiently low value of J_ε.

This form of algorithm for the TOC problem enables simple generalization of the algorithm for the OC problem (with the given fixed control time interval) through the expansion of the fourth step of the algorithm with the expression (35).

III. Example: Minimum-Time Control of Robot with Two Degrees of Freedom

In this section, the derived algorithm will be applied to minimum-time control of a robot with two degrees of freedom.

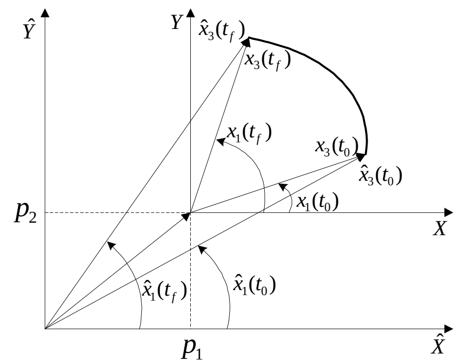

Fig. 2. Coordinate transformation of the initial coordinate system $X - Y$ ($p_1 = p_2 = 0$) to the optimal coordinate system $\hat{X} - \hat{Y}$.

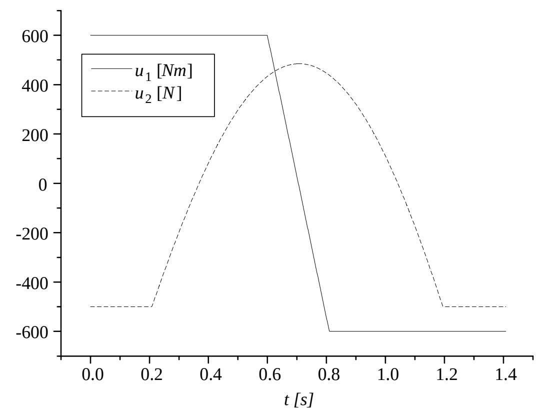

Fig. 3. Time dependence of control functions for $p_1 = p_2 = 0$, and $t_{min} = 1.41s$.

The optimal control of the non-linear robot model with a defined optimal criterion is still a relatively difficult task. The problem becomes more complex when two or more robots work in cooperation on a common task sharing workspace, time, constraints, and the cost function.

A. Dynamics of the Robot with Two Degrees of Freedom

The non-linear dynamic model of the manipulator with two degrees of freedom, [14] is presented through cylindrical coordinates in the form of

$$\begin{bmatrix} M_{11}(\mathbf{q}) & 0 \\ 0 & M_{22}(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} N_1(\mathbf{q}, \dot{\mathbf{q}}) \\ N_2(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} = \begin{bmatrix} P_1(t) \\ P_2(t) \end{bmatrix}, \tag{36}$$

where

$$\begin{aligned} M_{11} &= I_1 + I_2 + (m+M)q_2^2 + 2Maq_2 + Ma^2, \\ M_{22} &= m + M, \\ N_1 &= 2[(m+M)q_2 + Ma]\dot{q}_1\dot{q}_2, \\ N_2 &= -[mq_2 + M(a+q_2)]\dot{q}_1^2, \end{aligned}$$

where $\mathbf{q} = [\mathbf{q_1}\ \mathbf{q_2}]^\mathbf{T}$ are cylindrical coordinates of the center of the mass of link 3, (Fig. 1), M is total mass (manipulator hand and load), m is link mass, I_1 is the total moment of inertia of links 1 and 2 in relation to axis Z, I_2 is the moment of inertia of link 3 in relation to the axis which is parallel to axis Y and goes through point S, a is

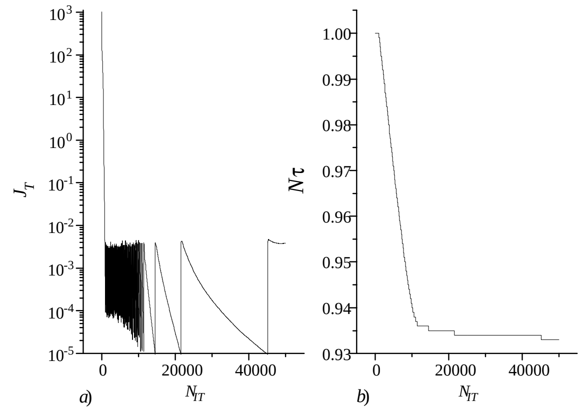

Fig. 4. Dependence of $a)$ J_T and $b)$ $N\tau$ on the number of iterations for $\Delta\tau = 0.000001\ s$, $N = 1000$ and $J_\varepsilon = 0.001$.

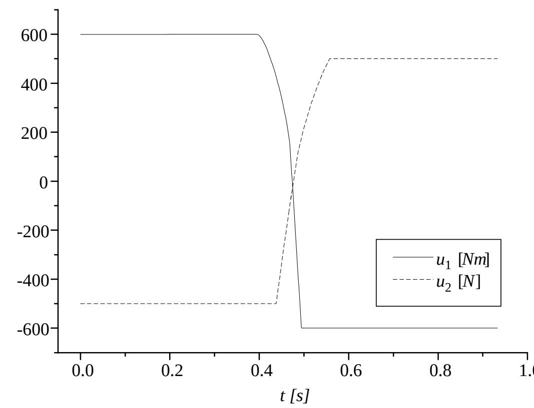

Fig. 5. Time dependence of control variables for $p_1 = -0.90313\,m$, $p_2 = 1.2625\,m$, and $t_{min} = 0.933s$.

the distance between the center of mass M and point S. $P_1(t)$ stands for the control moment of rotation q_1, while $P_2(t)$ is the control force of the translation q_2. The numerical values of the above-mentioned parameters are:
$M = 50kg$, $m = 97kg$, $I_1 + I_2 = 193kgm^2$, $a = 1.1m$, $P_1^{\max} = 600Nm$, $P_2^{\max} = 500N$,
where $P_1^{\max}$ is the maximum allowed moment and $P_2^{\max}$ is the maximum allowed force. If the above-mentioned system of the second-order differential equations is transformed into the system of the first-order differential equations, the following coordinate transformation is introduced:

$x_1 = q_1$, $x_2 = \dot{q}_1$, $x_3 = q_2$, $x_4 = \dot{q}_2$,
$u_1 = P_1$, $u_2 = P_2$,

and, for the sake of a more elegant expression, the following constants are introduced:

$A_1 = I_1 + I_2 + Ma^2$, $A_2 = 2Ma$, $A_3 = m + M$.

Thus, final form of the manipulator model is given by the equations

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= \frac{-2A_3x_2x_3x_4 - A_2x_2x_4}{A_1 + A_2x_3 + A_3x_3^2} + \frac{u_1}{A_1 + A_2x_3 + A_3x_3^2}\\ \dot{x}_3 &= x_4\\ \dot{x}_4 &= x_3x_2^2 - \frac{A_2}{2A_3}x_2^2 + \frac{u_2}{A_3}\end{aligned} \tag{37}$$

Here we have transformed the dynamics of the robot with two degrees of freedom to a system of four non-linear first-order differential equations.

B. Problem Formulation and Optimization Results

We should determine control functions $u_1(t)$ and $u_2(t)$ and parameters p_1 and p_2 of the coordinate transformations (Fig. 2)

$$\begin{aligned}\hat{x} &= x_3(\theta)\cos x_1(\theta) - p_1,\\ \hat{y} &= x_3(\theta)\sin x_1(\theta) - p_2,\end{aligned} \tag{38}$$

$$\begin{aligned}\hat{x}_1(\theta) &= \arctan(\hat{y}/\hat{x}),\\ \hat{x}_3(\theta) &= \sqrt{\hat{x}^2 + \hat{y}^2},\end{aligned} \tag{39}$$

where $\theta = t_0, t_f$, for the transformation of the robot state from the initial

$x_1(0) = \pi/18\,rad$, $x_2(0) = 0\,rad \cdot s^{-1}$,
$x_3(0) = 1\,m$, $x_4(0) = 0\,m \cdot s^{-1}$,

to the final one

$x_1(t_f) = 4\pi/9\,rad$, $x_2(t_f) = 0\,rad \cdot s^{-1}$,
$x_3(t_f) = 1\,m$, $x_4(t_f) = 0\,m \cdot s^{-1}$,

for minimum time $t_{min} \equiv t_f$, with control constraints

$$\begin{aligned}|\ u_1(t)\ | &\le u_1^{\max},\\ |\ u_2(t)\ | &\le u_2^{\max},\end{aligned} \tag{40}$$

where $u_1^{\max} = 600\,N \cdot m$, $u_2^{\max} = 500\,N$.

Fig. 3 represents the solution of the problem with fixed robot placement, i.e. without optimization according to the parameters of the coordinate transformation, $p_1 = p_2 = 0$. In this case, the minimum time is $t_{min} = 1.41\,s$ and control function $u_2(t)$ does not have the form of the bang-bang solution, which means that the available energy is not completely usable.

By using the algorithm for the calculation of the optimal parameter vector obtained are control and state variables which are shown in Fig. 5 and Fig. 6 for minimum time $t_{min} = 0.933\,s$. As can be seen in Fig. 4, with $\Delta\tau = 0.000001\,s$, $N = 1000$ and the given accuracy $J_\varepsilon = 0.001$, obtained is the minimum time $t_{min} = 0.933\,s$, i.e. $\tau_{min} = 0.000933\,s$.

The values of the optimal parameters are

$p_1 = -0.90313\,m$, $p_2 = 1.2625\,m$

(stationary state in Fig. 7. a). Convergence properties of the algorithm are illustrated in Fig. 7. b.

As can be seen in Fig. 5, control functions are in the form of the bang-bang solution, which means that the available energy is completely usable.

IV. Conclusions

In this paper it is presented an off-line algorithm for solving the problem of open-loop optimal control for nonlinear dynamic plants with unspecified initial and final conditions which are parameterized with a coordinate transformations. Also, it is derived a simple heuristic extension of that algorithm for the time optimal control problem. The specific application of the derived algorithm is demonstrated on the problem of minimum-time robot control.

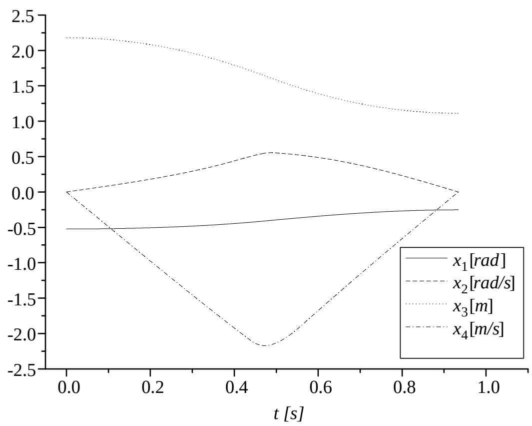

Fig. 6. Time dependence of state variables for $p_1 = -0.90313\,m$, $p_2 = 1.2625\,m$, and $t_{min} = 0.933s$.

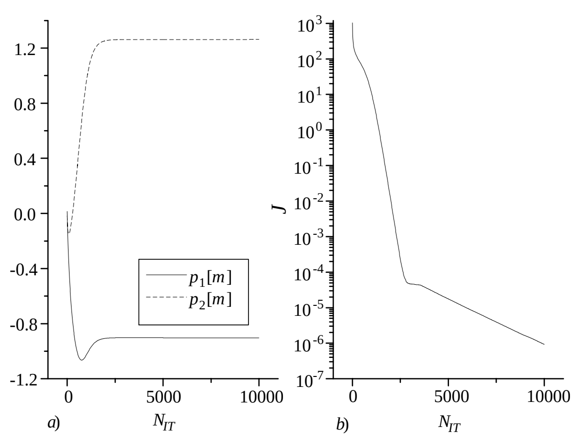

Fig. 7. Dependence of a) parameter vector on the number of iterations, and b) cost function on the number of iterations.

It can be seen that coordinate transformations of the robot placement provides an overall bang-bang solution of the minimum-time problem, which means that the available energy is completely usable. In other words, if robot placement is not strictly determined, only the appropriate coordinate transformations of the robot placement will guarantee the bang-bang solution of the minimum-time problem. Any other solution, which is not in the form of bang-bang control, can be considered as suboptimal.

The speed of convergence of the algorithm does not depend so much on the order of the system as it depends on the number of constraints, i.e. penalty functions.

Since proving the existence of a solution to the optimal control constraints problem is generally a very difficult problem, treating constraints with penalty functions is very useful also as an indication that the problem can be solved. If the penalty function does not converge to zero (for a stable algorithm with a constant convergence coefficient), this is a definite sign that the problem does not have a solution within the given constraints. Exactly this property of penalty functions has been used in the algorithm for time-optimal control.

Appendix I: Matrix Notation

$$\mathbf{X}(i) = [X_{lr}(i)]_{n\times n}, \qquad X_{lr}(i) = \frac{\partial f_l(i)}{\partial x_r(i)}$$

$$\mathbf{X}_{\mathrm{p}}(i) = [X_{\mathrm{p},rk}(i)]_{n\times N_p}, \qquad X_{\mathrm{p},rk}(i) = \frac{\partial x_r(i)}{\partial p_k}$$

$$\mathbf{X}_{\mathrm{f}}(i) = [X_{\mathrm{f},rk}(i)]_{n\times N_p}, \qquad X_{\mathrm{f},rk}(i) = \frac{\partial x_r(t_f)}{\partial p_k}$$

$$\mathbf{F}_{\mathrm{x}}(i) = [F_{\mathrm{x},k}(i)]_{n\times 1}, \qquad F_{\mathrm{x},k}(i) = \frac{\partial F(i)}{\partial x_k(i)}$$

$$\mathbf{F}_{\mathrm{p}}(i) = [F_{\mathrm{p},k}(i)]_{N_p\times 1}, \qquad F_{\mathrm{p},k}(i) = \frac{\partial F(i)}{\partial p_k}$$

$$\mathbf{J}_{\mathrm{p}} = [J_{\mathrm{p},k}]_{N_p\times 1}, \qquad J_{\mathrm{p},k} = \frac{\partial J}{\partial p_k}$$

$$\mathbf{J}_{1\mathrm{p}} = [J_{1\mathrm{p},k}]_{N_p\times 1}, \qquad J_{1\mathrm{p},k} = \frac{\partial J_1}{\partial p_k}$$

$$\mathbf{J}_{3\mathrm{p}} = [J_{3\mathrm{p},k}]_{N_p\times 1}, \qquad J_{3\mathrm{p},k} = \frac{\partial J_3}{\partial p_k}$$

$$\mathbf{J}_{1\mathrm{N}} = [J_{1\mathrm{N},k}]_{n\times 1}, \qquad J_{1\mathrm{N},k} = \frac{\partial J_1}{\partial x_k(N)}$$

$$\mathbf{J}_{1\mathrm{f}} = [J_{1\mathrm{f},k}]_{n\times 1}, \qquad J_{1\mathrm{f},k} = \frac{\partial J_1}{\partial x_k(t_f)}$$

References

[1] A. E. Bryson Jr. and Y. Ho *Applied Optimal Control*, New York: Blaisdell, 1969.

[2] D. G. Hull, "Conversion of Optimal Control Problems into Parameter Optimization Problems", *J. of Guidance, Control, and Dynamics*, Vol. 20, pp. 57-60, 1997.

[3] R. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, 1961.

[4] L. T. Fan and C. S. Wang, *The Discrete Maximum Principle*, New York, John Wiley, 1964.

[5] K. S. Narendra and K. Parthasarathy, "Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks", *IEEE Trans. on Neural Networks*, vol. 2, no. 2, pp. 252-262, 1991.

[6] P. Baldi, "Gradient Descent Learning Algorithm Overviev: A General Dynamical Systems Perspective", *IEEE Trans. on Neural Networks*, vol. 6, no. 1, pp. 182-195, 1995.

[7] B. A. Pearlmutter, "Gradient Calculations for Dynamic Recurrent Neural Networks: A Survey", *IEEE Trans. on Neural Networks*, vol. 6, no. 5, pp. 1212-1228, 1995.

[8] S. W. Piche, "Steepest Descent Algorithms for Neural Network Controllers and Filters", *IEEE Trans. on Neural Networks*, vol. 5, no. 2, pp. 198-212, 1994.

[9] A. Cichocki and R. Unbehauen *Neural Networks for Optimization and Signal Processing*, New York, John Wiley, 1993.

[10] P. J. Werbos, "Backpropagation through time: What it does and how to do it", *Proc. IEEE*, vol. 78, pp. 1550-1560, 1990.

[11] F. Beaufays and E. A. Wan, "Relating real-time backpropagation and backpropagation through time: An application of flow graph interreciprocity", *Neural Computation*, vol. 6, no. 2, 1994.

[12] E. S. Plumer, "Optimal Control of Terminal Processes Using Neural Networks", *IEEE Trans. on Neural Networks*, vol. 7, no. 2, pp. 408-418, 1996.

[13] J. Kasac, B. Novakovic, "Neural Network Application to Optimal Control of Nonlinear Systems", will be published in Proc. of 7^{th} Int. Conf. OPTI 2001, May 2001, Bologna, Italy.

[14] B. Heiman, H. Loose, G. Schuster, "Contribution to optimal control of an industrial robot", 4^{th} *CISM-IFTOMM Symp. on Theory and Pract. of Rob. and Manip.*, pp. 211-219, Zaborow, Poland, 1981.