Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 26 (2015) 223 – 228

www.elsevier.com/locate/procedia

12th Global Conference on Sustainable Manufacturing

# Quasi-linearization Approach for the Under-actuated Robots

Ahmad Albalasie[a,*], Arne Glodde[a], Guenther Seliger[a], Ahmed Abu Hanieh[b]

[a]*Technical University of Berlin, Pascalstr. 8-9, Berlin 10587, Germany*

[b]*Birzeit University, Birzeit, Palestine*

* Corresponding author. Tel.: +49 (0)30 / 314-21255; fax: +49 (0)30 / 314-22759. *E-mail address:* albalasie@mf.tu-berlin.de

**Abstract**

A novel technique to reduce energy consumption for industrial robots by using redundancy and under-actuated configurations is introduced in this paper. The study concentrates on kinematics including a passive axis which causes a highly nonlinear coupling of the Coriolis and centrifugal forces and high inertia coupling. The challenge of meeting the requirements of position accuracy, precision, and repeatability combined with the requirements for the speed, acceleration, and torque is coped with by solving a sequence of linear two point boundary value problems for controlling the movement of the end effector between two points. The advantages of this method are the ability of reducing the computation time dramatically, computer storage, and linearizing the nonlinear model. Furthermore the optimal trajectories are identified with minimizing energy consumption and enabling high speed capability. Numerical simulations are conducted to validate the theoretical analysis.



Peer-review under responsibility of Assembly Technology and Factory Management/Technische Universität Berlin.

*Keywords: under-actuated motion; robots; control; optimization; quasi-linearization; trajectories planing.*

## 1. Introduction

SAMARA is a prototype for industrial horizontal planar robot for material handling process. The configuration for SAMARA prototype is redundant. Furthermore SAMARA has three linked robot arms with three actuators. The two motors at the shoulder (joint 1), and the elbow (joint 2) are completely active but there is a passive axis in the third link in the phase of under-actuated motion see Fig.1.

SAMARA uses a two phases motion which consists of the null space motion to execute the pick and place tasks (in this paper the null space motion is not discussed) and the under-actuated motion to move from the initial point *(Point A)* to the desired point *(Point B)*. One of the main differences between the two phases is the last axis is a passive axis in the phase of the under-actuated motion, on the contrary the last axis is an active axis in the phase of null space motion.

The main goal of SAMARA robot is reducing the energy consumptions while maintain execution time. In addition position accuracy, repeatability, precision, and increasing the reliability must be satisfied. Since the current model of SAMARA is a nonlinear so it takes a long time to calculate how to execute each task as a result of that the control for SAMARA is based on offline control [1]. This paper uses the quasi-linearization to reduce the computation time, simplifying the model as much possible, and minimizing the energy consumption cost function for SAMARA robot to compute the angular positions, angular velocities, angular accelerations and input torques trajectories to execute each task.

Fig. 1 second generation of SAMARA robot

Controlling the under-actuated motion in SAMARA is a challenging problem. This is caused by the nonlinearity of



Peer-review under responsibility of Assembly Technology and Factory Management/Technische Universität Berlin.
doi:10.1016/j.procir.2014.07.168

the mathematical model currently used for SAMARA. This nonlinearity is caused by the large range of space tasks, the kinematic coupling between the mechanical components. Moreover the configuration is redundant so the torque control inputs less than the number of joints. Another challenging problem is searching for the optimal values responsible for reducing the energy consumption. Beside this, the passive axis in SAMARA manipulator is a second order non-holonmic constraint and thus increasing the difficulty of control [2]. However the controllability of the 3-DOF manipulator with a passive axis under a second order non-holonmic constraint was studied by using the constructive method [3] and the result is the system is globally controllable [2].

The under-actuated motion was solved by using other methods such as the partial feedback linearization [4], or the suggested solution by Oriolo and Nakamura [5] but the previous methods can't find the optimal trajectories for the angular positions, angular velocities, angular acceleration and input torques. Nevertheless there are a lot of methods that can solve the numerical optimization problem such as gradient method, quasi-linearization, neighbouring external method, shooting method, and others. This paper applies a quasi-linearization approach [6] to determine the optimal control for the under-actuated manipulators with two point boundary value problem with specified values of the boundary conditions and specified time. Furthermore the solution must satisfy the necessary conditions for the optimality to solve the Eular Equation and the sufficient conditions [6].

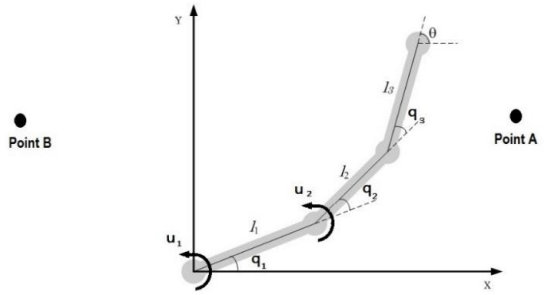


Fig. 2 SAMARA configuration with Point A and Point B in the phase of under-actuated motion.

The organization of this paper is as follows. Problem formulation to describe the problem details with mathematical equations in section 2. Procedures to convert the nonlinear differential equations model to linear time varying differential equations model in section 3. Section 4 contains the quasi-linearization algorithm which can control the under-actuated motion to solve the numerical optimization problem. In addition, section 5 shows two numerical examples to create the under-actuated trajectories for the SAMARA manipulator. In conclusion, section 6 discuss an outlook to future works and the conclusions for this paper

## 2. Problem Formulation

SAMARA has two phases of motion. The first phase is the null space motion to execute the pick and place tasks between multi-stations and the second phase is the under-actuated motion to move from one point to another one to start the new null space motion. Each cycle has two phases of null space motion and two phases of under-actuated motion.

Reducing the energy consumption in each cycle is an important issue. This paper focuses on minimizing the energy consumption in the under-actuated motion not only by using the last axis as a passive axis which means the motor in the last joint does not work in this phase, but also by using the dynamic programing optimization techniques to find the optimal value for the energy consumption cost function. As well as SAMARA must have the capability for high speed to reduce the operating expenses and increase the efficiency of the material handling process.

The reason for the optimization problem being difficult to solve is its nonlinear character. This difficulty is caused because of the requirements resulting from the calculus of variation. These must be satisfied in order to find the minimal optimal value for the cost function in the feasible set. The solution can be found by using an iterative process.

The idea focuses on controlling the end effector between two points *(Point A and Point B is shown in Fig. 2)*. These points are the start and the end points for the under-actuated motion by using quasi-linearization. The nonlinear equations represent the mathematical model for SAMARA prototype and must be formulated to minimize the energy consumption for the robot. Quasi-linearization applies to find the optimal angular positions, angular velocities, and angular accelerations per each axis to minimize the cost function.

The quasi-linearization solution is a sequence of a linear two-point boundary problem. By using the numerical integration in each sample, the problem can easily be solved by using an iterative process. The dynamic equations for the horizontal planer under-actuated manipulator is shown in Eq. (1).

$$M(q)\ddot{q} + C(q,\dot{q}) + D(\dot{q}) = U \quad (1)$$

The number of axes in the robot is $n$, as a result of that $M(q) \in R^{n*n}$ is a symmetric positive definite inertia matrix, also $C(q,\dot{q}) \in R^n$ are a Coriolis and centrifugal forces. In addition; $D(q,\dot{q}) \in R^n$ are the damping and friction moments, moreover $U \in R^n$ are the motor torques. Where the generalized coordinates for the angular positions are $q \in R^n$, as well as vector $q$ divided into two sub vectors. Vector $q_1 \in R^a$ which are the generalized coordinates for the active axes and $q_2 \in R^p$ which are the generalized coordinates for the passive axes. As a consequence Eq. (1) can be expressed in more details about the active and passive axes as shown below:

$$\begin{bmatrix} m_{aa}(q) & m_{ap}(q) \\ m_{pa}(q) & m_{pp}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} c_1(q,\dot{q}) \\ c_2(q,\dot{q}) \end{bmatrix} + \begin{bmatrix} d_1(\dot{q}) \\ d_2(\dot{q}) \end{bmatrix} = \begin{bmatrix} T \\ 0 \end{bmatrix} \quad (2)$$

where: $c_1(q,\dot{q}) \in R^a$, and $c_2(q,\dot{q}) \in R^p$ are the Coriolis and centrifugal forces for the active and passive axes respectively.
$d_1(\dot{q}) \in R^a$, and $d_2(\dot{q}) \in R^p$ are the damping and friction moments for the active and passive axes respectively.
$T \in R^a$ is the vector containing motor torques.
The state space representation for the under-actuated robot:

$$x_1 = q \quad , \quad x_2 = \dot{x}_1 = \dot{q} \quad , \quad \dot{x}_2 = \ddot{q}$$

$$\begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \begin{bmatrix} x_2 \\ M^{-1}(x_1) * (U - C(x_1, x_2) - D(x_2)\end{bmatrix} = f(x(t), u(t), t) \quad (3)$$

As discussed before Eq. (4) expressed the minimization problem for the energy consumption

$$J = \min \frac{1}{2}\int_{t_0}^{t_f} U^T * R * U \quad (4)$$
$$\text{sub to: } \dot{x}(t) = f(x(t), u(t), t)$$

where: $U^T$ is the transpose of vector $U$.
R is a positive semi-definite matrix with $R \in R^{n*n}$

Obviously to minimize the previous cost function it is necessary to have the values of the boundary conditions *(in Point A and Point B see Fig. 2)* to compute the under-actuated motion between two null space motions, but these values are known from the null space trajectories:

*Point A* is the end of the right/left null space motion.
*Point B* is the start of the left/right null space motion.

The Hamiltonian equation for the manipulator is shown below

$$\mathcal{H} = U^T * R * U + \sum_{i=1}^{n} \lambda_i * f(x(t), u(t), t)_i \quad (5)$$

Where $\lambda \in R^n$ is the Lagrange multipliers functions, also called the co-state variables.

Moreover the state and co-state equations are

$$\dot{x}^*(t) = \frac{\partial \mathcal{H}}{\partial \lambda} = f(x(t), u(t), t) \quad (6)$$

$$\dot{\lambda}^*(t) = -\frac{\partial \mathcal{H}}{\partial x} = -\frac{\partial f(x(t),u(t),t)}{\partial x}\lambda^*(t) \quad (7)$$

The last equation Eq. (7) is called the co-state equation

Quasi-linearization can solve the optimization problem after estimating the initial nominal trajectories for the state and co-state equations, then the differential equations are linearized around the nominal trajectories at each sample which must satisfy the boundary conditions at the end of the iteration process. There are many approaches to solve this problem, the first one is the adjoint method [7], the second approach is the complementary function method [8], another technique is to use the expansion of the Chebyshev series [9].

The algorithm can converge if and only if the initial estimate for the nominal state and co-state trajectories are not poor, otherwise the solution will diverge. Usually it is easy to estimate the nominal state trajectories because this is related to the dynamics of the robot. However, the problem is to estimate the nominal co-state trajectories because these trajectories are not related to the physics. However, there is a publication help to cope with this challenge e.g. [10].

## 3. Procedures to linearize the model

Linearization of the state and co-state equations is a sensitive problem. This is caused by the possibility of divergence for the modified nominal states and co-states trajectories. Under these circumstances this process also takes a lot of computations and therefore computation time, because usually this process is a periodically process in each iteration. Moreover the quasi-linearization modifies the nominal trajectories per iteration to satisfy the necessary and sufficient conditions for optimality [6] [11].

The algorithm converges if the stop criteria is satisfied. However, there are many reasons for divergence e.g. the initial estimated trajectories are so poor [10] or the linearization is not accurate enough. However, the most famous method to linearize the model around the nominal trajectories per each sample is the Taylor series. But before linearizing the model, it is necessary to find the solution of the following equations to minimize the cost function.

$$\frac{\partial \mathcal{H}}{\partial u} = 2 * U + \left[\frac{\partial f(x^*(t),u^*(t),t)}{\partial U}\right]^T \lambda^*(t) = 0 \quad (8)$$

Substituting the values of $u^*$ in Eq. (6) and Eq. (7) the result is reduced nonlinear state and co-state equations. These equations functions of states, Lagrange multipliers and the time. Linearized model Eq. (9) can clarify this point.

$$\begin{bmatrix}\dot{x}^{i+1}(t) \\ \dot{\lambda}^{i+1}(t)\end{bmatrix} = \begin{bmatrix}a_{11} & a_{12} \\ a_{21} & a_{22}\end{bmatrix}\begin{bmatrix}x^{i+1}(t) \\ \lambda^{i+1}(t)\end{bmatrix} + \begin{bmatrix}e_1(t) \\ e_2(t)\end{bmatrix} \quad (9)$$

Where

$$a_{11} = \left[\frac{\partial f}{\partial x}\right]_{yo} = \frac{\partial f_y}{\partial x_o} \quad , \; a_{12} = \left[\frac{\partial f}{\partial \lambda}\right]_{yo} = \frac{\partial f_y}{\partial \lambda_o}$$

$$a_{21} = \left[\frac{\partial^2 \mathcal{H}}{\partial x^2}\right]_{yo} = \frac{\partial^2 \mathcal{H}}{\partial x_y \partial x_o} \; , \; a_{22} = \left[\frac{\partial^2 \mathcal{H}}{\partial x \partial \lambda}\right]_{yo} = \frac{\partial^2 \mathcal{H}}{\partial x_y \partial \lambda_o}$$

$$e_1(t) = f(x(t), \lambda(t), t) - a_{11}(t)x(t) - a_{12}(t)\lambda(t) \quad (10)$$

$$e_2(t) = -\frac{\partial \mathcal{H}}{\partial x} - a_{21}(t)x(t) - a_{22}(t)\lambda(t) \quad (11)$$

Even though the new system is a linear model, but its time varying differential equations. This issue is going to be discussed in the following section.

## 4. Quasi-linearization

There are two groups to solve the numerical optimization problem i.e. direct numerical optimization e.g. gradient method, and the steepest decent method suggested by Bryson [12] or indirect numerical optimization e.g. quasi-linearization, or the perturbation methods. In this paper the quasi-linearization solves the problem by integrating the linearized differential equations at each sample around the nominal trajectories to modify the nominal trajectories at each iteration to satisfy the optimality conditions. On the contrary, the perturbation methods which solve the problem by integrating the nonlinear differential equations around nominal trajectories [13].

The main difference between the indirect and direct methods is in the philosophy of searching the solution. In the case of the indirect methods the use of conditions is required for the mathematical optimality as starting point and seek by various iterative philosophies, to satisfy these conditions. On the contrary, the direct methods use only the desired terminal conditions and the equations of motion as starting to find the optimal value for the cost function [13]. In this paper one of the indirect methods is used to solve the numerical optimization problem. The reason for choosing an indirect method is that the convergence rate is very slow in the neighbourhood of the optimal solution in the case of the gradient method.[13].

Quasi-linearization was introduced the first time by Bellman and Kalaba [11]. The major advantages of the quasi-linearization is the process of computer implementation, computer storage and the computation time. In addition; it converges rapidly in order for the computation time to be significantly reduced in comparison to other methods (exception is the perturbation method). Quasi-linearization is an iterative method that approximates the original nonlinear boundary value problem with a series of linear ones which are easier to solve numerically. Paper by Tapley and Lewallen [13] compared this approach with other popular iterative methods such as gradient methods, perturbation methods and the second variation method in solving optimal control problems [10].

Initial nominal trajectories $x^{(0)}(t), \lambda^{(0)}(t)$ must be estimated to start the quasi-linearization, then the modified nominal trajectories are obtained by integrating the linearized form to satisfy the optimality conditions. This process is a periodical process at each sample and in each iteration. The coefficients used to form modified nominal trajectories are obtained from the previous nominal trajectories. The algorithm converges if and only if the optimality condition is satisfied Eq. (8) since the boundary condition is satisfied on each iteration and the system is a linear. Under appropriate conditions, the successive solution of the linearized equations converges to the solution to the original set of nonlinear equations.

Eq. (9) has (*2*n*) differential equations. It contains of two major parts. The first part is a homogeneous and the second part is the particular part. Eq. (13) clarifies only the (*2*n*) homogeneous solutions which can be found by using the forward integration with these boundary conditions:

$$\begin{matrix} x^{H1}(t_0) = 0 \ , & \lambda^{H1}(t_0) = [1 \ \ 0 \ \ 0 \dots 0]^T \\ x^{H2}(t_0) = 0 \ , & \lambda^{H2}(t_0) = [0 \ \ 1 \ \ 0 \dots 0]^T \end{matrix} \qquad (12)$$

$$x^{Hn}(t_0) = 0 \ , \quad \lambda^{Hn}(t_0) = [0 \ \ 0 \ \ 0 \dots 1]^T$$

The linearized homogeneous equations:

$$\begin{bmatrix} \dot{x}^{i+1}(t) \\ \dot{\lambda}^{i+1}(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x^{i+1}(t) \\ \lambda^{i+1}(t) \end{bmatrix} \qquad (13)$$

The particular solution can be found by using the boundaries in Eq. (14) and the forward integration for the nonhomogeneous equations Eq. (9)

$$x^p(t_0) = x_0 \quad , \quad \lambda^p(t_0) = 0 \qquad (14)$$

The superposition principle *(for homogeneous and particular solutions)* can be used to estimate the behaviour of the nonlinear equations as shown in Eq. (15).

$$x^{i+1}(t) = [x^{H1}(t) \ \ x^{H2}(t) \dots x^{Hn}(t)]c + x^p(t) \qquad (15)$$

Where $c \in R^n$ is an unknown vector. However, the specified preselected time $(t_e)$ and the specified state space at the start and end points *(point A and point B in Fig. 2)* in the under-actuated motion Eq. (17) are used to find the value of *c* vector

$$x^{i+1}(t_e) = [x^{H1}(t_e) \ \ x^{H2}(t_e) \dots x^{Hn}(t_e)]\boldsymbol{c} + x^p(t_e) \qquad (16)$$

$$c = [x^{H1}(t_e) \dots x^{Hn}(t_e)]^{-1} * (x^{i+1}(t_e) - x^p(t_e)) \qquad (17)$$

Thus the algorithm now completes one iteration if the stop criteria is satisfied the optimal solution converges. Otherwise is a new iteration process initiated by the algorithm including the new boundary conditions resulting from the previous iteration [6] [11].

The stop criteria for the algorithm is:

$$\left\| \begin{bmatrix} x^{i+1}(t) \\ \lambda^{i+1}(t) \end{bmatrix} - \begin{bmatrix} x^{i}(t) \\ \lambda^{i}(t) \end{bmatrix} \right\| \leq \varphi \qquad (18)$$

Where $\varphi$ is preselected as a small number close to zero.

## 5. Simulation and Numerical Results

The validation of the previous approach will be discussed in this section. There are two under-actuated trajectories between two null spaces motions which will be discussed in this section.

The basic dynamics and kinematics of SAMARA was already discussed by Brett [1]. Even though this paper uses the same kinematic and dynamic equations as Brett, the model is significantly enhanced by the use of new parameters. Those parameters reflect the second generation of SAMARA manipulator shown in Fig. (1).

**Case 1:**

Find the optimal torques for the active axes and the optimal paths that minimize the following cost function:

$$J = \min \tfrac{1}{2} \int_0^{1.55} [u_1 \ u_2 \ 0] * R * [u_1 \ u_2 \ 0]^T \qquad (19)$$

$$\text{sub to } \dot{x}(t) = f(x(t), u(t), t)$$

Whereas $\varphi$ =0.00001, execution time $T_e = 1.55$ sec. and R $\in R^3$ is a unity diagonal matrix,

Also the two point boundary conditions Point A and Point B are the (end right / start left) null space points are

$$\text{Point A}=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\\x_6\end{bmatrix}=\begin{bmatrix}0.2340\\0.1439\\-1.4822\\1.9108\\-4.4502\\2.0509\end{bmatrix},\quad \text{Point B}=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\\x_6\end{bmatrix}=\begin{bmatrix}2.9077\\-6.4272\\7.7654\\1.9987\\-4.6550\\2.1450\end{bmatrix}$$

The result of the quasi-linearization algorithm are angular position, angular velocity, angular acceleration, and input torque for each axis. The results are shown in Fig. (3) and the iteration process for the cost function is shown in Table (1).

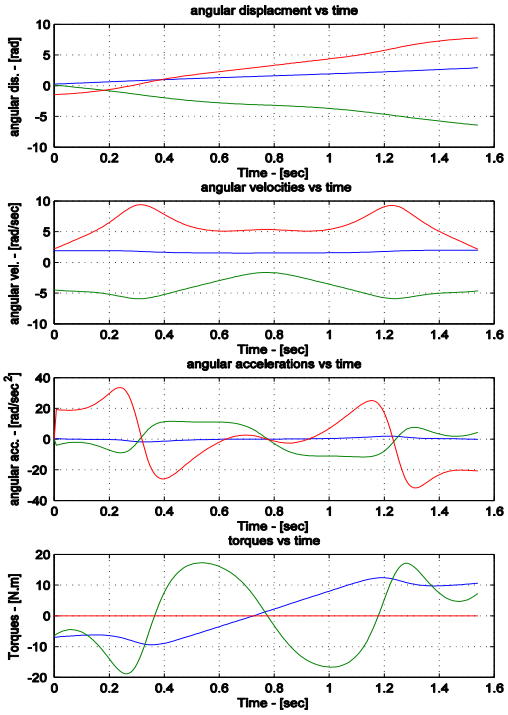


Fig. 3 optimal paths and torques (axis 1- blue, axis 2 - green, axis 3 - red)

Table 1. Iteration process for case 1.

| Iteration No. | Cost Function value | Error |
|---|---|---|
| 1 | 212.141935 | 1637.289061 |
| 2 | 223.758131 | 533.163634 |
| 3 | 155.251109 | 369.962165 |
| 4 | 158.643186 | 40.141543 |
| 5 | 158.167481 | 3.789478 |
| 6 | 158.180413 | 0.220495 |
| 7 | 158.177817 | 0.013147 |
| 8 | 158.177923 | 0.000776 |
| 9 | 158.1779131 | 0.000080 |
| 10 | 158.1779136 | 0.000004541 |

**Case 2:**

Find the optimal torques for the active axes and the optimal paths that minimize the following cost function:

$$J=\min\tfrac{1}{2}\int_0^{1.55}[u_1\ u_2\ 0]*R*[u_1\ u_2\ 0]^T \qquad (20)$$

$$\text{sub to } \dot{x}(t)=f(x(t),u(t),t)$$

Whereas $\varphi$ =0.00001, execution time $T_e = 1.55$ sec. and R $\in R^3$ is a unity diagonal matrix. Furthermore the two point boundary conditions Point A and Point B are the (end left / start right) null space points are

$$\text{Point A}=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\\x_6\end{bmatrix}=\begin{bmatrix}-3.3157\\-0.2823\\1.5382\\1.9886\\-4.5623\\1.5895\end{bmatrix},\quad \text{Point B}=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\\x_6\end{bmatrix}=\begin{bmatrix}0.1768\\-6.0070\\-1.5361\\1.9015\\-4.3658\\1.5430\end{bmatrix}$$

The result of the quasi-linearization algorithm are shown in Fig. (4) and the iteration process for the cost function is shown in Table (2)

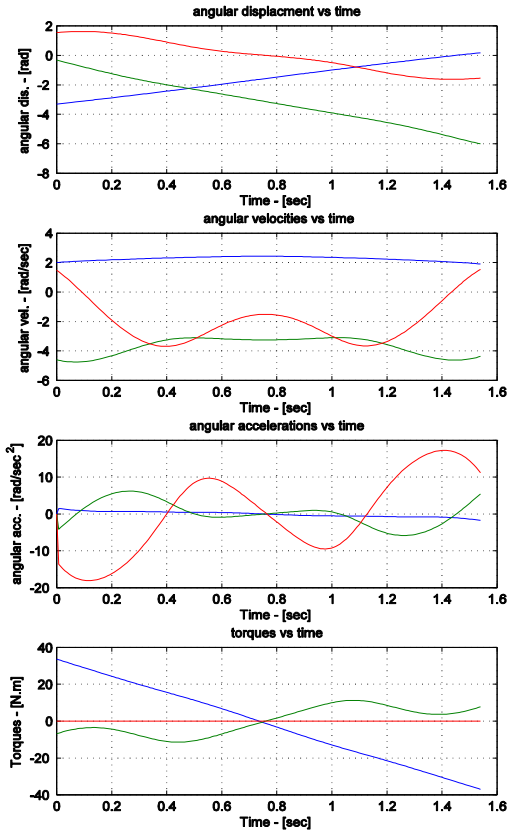


Fig. 4 optimal paths and torques (axis 1- blue, axis 2 - green, axis 3 - red)

Table 2. Iteration process for example 2.

| Iteration No. | Cost Function value | Error |
|---|---|---|
| 1 | 624.943971 | 3553.998971 |
| 2 | 541.716760 | 3616.440512 |
| 3 | 351.185439 | 1474.096654 |
| 4 | 366.075488 | 213.343595 |
| 5 | 366.066474 | 1.783673 |
| 6 | 366.064791 | 0.022397 |
| 7 | 366.064806 | 0.000373 |
| 8 | 366.0648055 | 0.00000305355 |

## 6. Conclusion

In this paper, a numerical algorithm has been developed to solve the problem of optimal control for the under-actuated manipulators. The SAMARA 2nd generation prototype can execute these trajectories easily but the main benefit of this algorithm is the convergence rate consequently the computation time is very short in comparison to the previous approach. It is shown in the iteration tables, that the system converges faster than the previous code indicated in [1]. In addition; another interesting result is the possibility controlling SAMARA robot with a passive axis under a non-holonomic constraint but with taking into account finding the minimum value for the cost function without using any breaks in this type of the manipulator.

The quasi-linearization method is used to convert a nonlinear optimal control problem into a sequence linearized two point value problems which are solved by forward integration. The update laws for the nominal trajectories ensure satisfaction of the terminal conditions. Compared to nonlinear programming based methods, the approach offers significant advantages in computational efficiency.

Future works will concentrate on the effect of physical constraints for SAMARA on the control theory, and execute the pick and place tasks in minimum time. Furthermore, the possibility for mixing direct and indirect algorithms to improve the numerical calculations for the optimization problem has to be explored.

## References

[1] T. Brett, Effictive Motion Design Applied to Energy-Efficient Handling Processes, Berlin: Fraunhofer Verlag, 2013.

[2] H. Aria, “Controllability of a 3-DOF Manipulator with a Passive Joint under a Nonholonomic Constraint,” in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on (Volume:4 )*, Minneapolis, 1996.

[3] J.-P. Laumond, “Feasible Trajectories for Mobile Robots with Kinematic and Environment Constraints,” in *Intelligent Autonomous Systems*, Amsterdam, 1987.

[4] M. W. Spong, “Partial Feedback Linearization of Underactuated Mechanical Systems,” in *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on (Volume:1 )*, Munich, 1994.

[5] G. Oriolo and Y. Nakamura, “Control of mechanical systems with second-order nonholonomic constraints: underactuated manipulators,” in *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on*, 1991.

[6] D. E. Kirk, Optimal Control Theory An Introduction, Mineolla, New York: Dover Publications, Inc., 2004.

[7] . T. R. Goodman and . G. N. Lance , “The numerical integration of two-point boundary value problems,” *Math. Comp. 10 ,* pp. 82-86, 1956.

[8] L. Fox, Numerical Solution of Ordinary and Partial Differential Equations, Massachusetts: Addison-Wesley Publishing Company, 1962.

[9] C. W. CLENSHAW, “THE NUMERICAL SOLUTION OF LINEAR DIFFERENTIAL,” *Mathematical Proceedings of the,* pp. 134-149, 1956.

[10] B. P. YEO, K. J. WALDRON and B. S. GOH, “Optimal initial choice of multipliers in the quasilinearization method for optimal control problems with bounded controls,” *International Journal of Control, vol. 20,* pp. 17-33, 1974.

[11] R. E. Bellman and R. E. Kalaba, “Quasilinearization and nonlinear boundary-value problems,” RAND Corporation, 1965.

[12] A. E. Bryson and W. F. Denham, “A Steepest-Ascent Method for Solving Optimum Programming Problems,” *Journal of Applied Mechanics | Volume 29 | Issue 2 ,* pp. 247-257, 1962.

[13] B. D. Tapley and J. M. Lewallen, “Comparison of several numerical optimization methods,” *JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 1, No. 1,* pp. 1-32, 1967.