Technical University of Liberec

Faculty of Mechatronics,
Informatics and
Interdisciplinary Studies

Ing. Abbas Chatraei

Doctoral Thesis Statement

Technical University of Liberec
Faculty of Mechatronics, Informatics and Interdisciplinary Studies

Optimal Control of Robot Manipulators

Doctoral thesis statement for obtaining the academic title “Doctor” abbreviated to “Ph.D”.

Author:
Ing. Abbas Chatraei

Supervisor:
Doc. Mgr. Ing. Václav ZÁda, CSc.

October 24, 2011

Overview

This dissertation deals with the optimal positioning control of robot manipulators that is one of the sub-branches of motion control and trajectory planning. In fact, our main objective is to present the new method(s) to control a robotic system in such a way that some performance criterion is optimized. There is a wide range of performance criteria which can be considered like energy consumption, the required time to perform some desired task by the robot or a compound criterion including both energy and time. In this thesis, we present two new methods so that the first one solves unconstrained global optimal control problem (OCP) and second one solves the constrained OCP of robot manipulators. The first method presents an optimal feedback control system and obtains a global solution for the considered unconstrained OCP in a completely innovative manner. In the second proposed method which is actually a compound method, the optimal trajectories are computed by use of an iterative linearization method so that in each iterate a parametric optimization method is applied to obtain the optimal trajectories. It is proved that after a finite number of iterations, the sequence of optimal trajectories converge to the optimal trajectories of original nonlinear (robotic) system.

Another subject addressed in this thesis, is robot identification. For designing an optimal control scheme, we require an exact model of the robot. In this thesis a comprehensive procedure of identification experiment for a KUKA robot available in robotic laboratory of Mechatronic faculty of TUL is presented.

This dissertation has six chapters as follows: chapter 1 is an introduction to robot kinematics, dynamics and identification, as well as the OCP. Chapter 2 deals with the subjects such as robot kinematics, dynamics and identification, in detail. In this chapter the results of KUKA robot identification is presented. Chapter 3 addresses the subjects relating to the OCP formulation and the different methods to solve this problem. In chapter 4, we present our first proposed method to solve the unconstrained OCP of robot manipulators. Then, the proposed method to solve the constrained OCP of robot manipulators is dealt with in chapter 5. Eventually, in chapter 6 the concluding remarks are presented.

Contents

1 Introduction **2**

2 Optimal Control of Robot Manipulators **2**

3 Objectives of the Dissertation **3**

4 Kinematic and Dynamic Modeling of Serial Industrial Robots **4**
4.1 Kinematic Model of the Robot 4
4.2 Dynamic Model of the Robot 4

5 Robot Identification **5**

6 First Proposed Optimal Control Method **6**
6.1 New Proposed Theorem 7
6.2 Realization 8
6.2.1 Option of Matrix T 9
6.2.2 Solution of e_i and Optimal feedback Control 10
6.3 Adaptive Global Optimal Control 13

7 Second Proposed Optimal Control Method **16**
7.1 Formulation of Robot Optimal Control Problem 16
7.2 Proposed Method for Solving OCP of Robot Arms 18
7.3 Parametric Optimization Method 19
7.4 The Proposed Algorithm 20
7.5 Applying proposed method into KUKA robot 23

8 Conclusion **25**

9 References **27**

1 Introduction

Modern industrial robots are electro-mechanical systems whose history dates back a few recent decades. The first industrial robot was manufactured by George Charles Devol, in 1954 called Unimation. After that various universities and companies designed and built different industrial robots such as Puma 560, Stanford, SCARA, etc. Like human body, the members of each open chain manipulator include: waist, shoulder, elbow, wrist and end effector (hand). In fact, these are the joints of the robot by which the robot links are connected to each other.

With growing applications of robot manipulators in industrial factories, one of the key features which has been considered is to increase productivity with low energy consumption as much as possible. One of the possible ways is to design a controller for the robot manipulators which perform the respective task in minimum time and minimum energy consumption. Hence, designing such controllers is the goal in the optimal control technique and in this thesis is attempted to propose new method(s) for achieving this goal.

2 Optimal Control of Robot Manipulators

Usually researches face with some difficulties for solving optimal control problem (OCP) of robot manipulators. The major problem is that the robots have a highly nonlinear and coupled dynamics which anyone who derived these dynamics may realize the complexity of these equations which is to ever for a 6 degrees of freedom (DOF) manipulator. In general, there are three main approaches to solve the (continuous or discrete) OCP of nonlinear dynamical systems [10]:

- Dynamic programming,
- Indirect methods (variational approaches),
- Direct methods.

In the case of robot manipulators, firstly the time-OCP of robot manipulators was solved by indirect methods. In fact, this problem was seriously considered by Bobrow in his Ph.D dissertation [6]. He proposed a new method in which a modified Pontryagin's minimum principle is used to calculate the optimal control torque of each joint of a robot manipulator. After this proposed method, some other researches were presented by which a set of improvements were applied to the Bobrow's first work [5, 27], for instance by considering the singularity problem appeared in OCP [25, 28]. Some other studies added a energy term to the cost function [26] and also actuator dynamics [24].

Dynamic programming method also was employed to obtain the minimum-time optimal trajectories [29, 3, 11]. In [29], the Bobrow's method was used to solve the OCP of robot manipulators, but for computing the optimal controls, a dynamic programming algorithm has been developed to derive the reduced set of second order differential equations in terms of path parameter.

Although two above methods have been used successfully in many applications, but they have been replaced by direct methods in recent years. The basic idea of this method, in the case of robot manipulators, is that the joint trajectories are approximated by a parametric function such as spline functions and then using a nonlinear programming, the optimal values of the parameters in approximating function are achieved. One of the main advantages to this approximations is that usually the resulted parametric optimization problem has a feasible solution. Many researchers presented different approaches to generate the optimal joint trajectories. Among these works, polynomial cubic spline functions and B-splines have been used in many studies [4, 31, 32, 33, 18, 14]. In [4], B-spline functions were used to parameterize the joint motions and derive a general optimization technique for robots using Denavit-Hartenberg parameters of the robot and the full robot dynamics. In [31] a cubic spline trajectory is used to converting the OCP into a finite dimensional optimization problem by considering maximum values of velocity, acceleration and jerk for all robot joints. Point to point trajectory parametrization was performed in [32] by means of cubic B-splines. [33] proposes a method to obtain a global solution to OCP of robot manipulators. Incorporating both acceleration and jerk as the objectives is considered in [18]. Also in [14] an efficient algorithm is proposed to solve the OCP by using polynomial cubic spline functions. There is also another sub-part of the direct methods called *shooting methods*. These kinds of methods, which include single shooting, collocation and multiple shooting methods, use a constant piecewise function to parametrize the control inputs (robot joint's torques and forces) of the system. These methods have been considered in [10, 7], in detail.

In most studies mentioned above, the obtained optimal solution is a local one whereas a small number of researches are found to obtain the global optimal solution to the OCP of robot manipulators. In the our first proposed method, we consider this subject so that it yields a global optimal solution to the considered unconstrained OCP. In the second proposed method, we present a combined optimal control method through which the constrained OCP of robot manipulators is solved. In this approach it is attempted to dominate the complexity of robot dynamics and solve the constrained OCP of robot manipulators during some stages.

3 Objectives of the Dissertation

In this study, the main objective is to solve the optimal control problem (or optimal dynamic motion planning) of open-chain robot manipulators. In doing so, we first require a precise dynamic model of the considered system. Thus, in the case of robot manipulators, we developed an identification experiment to estimate the dynamic and friction parameters of the our case study, i.e. a KUKA robot available in robotic laboratory of Mechatronic faculty of TUL. As shown in Figure 1, it is a 6 degrees of freedom industrial robot for which we assume the last three joints are fixed in their home positions.

Figure 1: KUKA IR 364/10-VK 10 Robot Manipulator

Therefore, the objectives considered in this dissertation can be summarized as follows:

1. The elementary objective is to obtain a kinematic and dynamic models of the robot in closed-form using modified Denavit-Hartenberg notation and recursive Newton-Euler formulation, respectively.

2. The second objective is to develop a well-designed identification procedure to estimate the dynamic and friction parameters of the robot such as mass, inertial parameters and location of mass center of each link of the robot, as well as Coulomb and viscous friction parameters of each robot joint.

3. The third objective is to present an approach to solve the unconstrained OCP of robot manipulators.

4. The fourth objective is to propose a method to solve the optimal dynamic point to point control problem of robot manipulators.

4 Kinematic and Dynamic Modeling of Serial Industrial Robots

4.1 Kinematic Model of the Robot

Let us consider an n-axes robot manipulator, as shown in Figure 2. Then, a coordinate frame is adopted to each link so that the position and orientation of the last link , i.e., pose of frame $\{n\}$, relative to base link frame $\{0\}$ is described by the following coordinate transformation:

$$^{0}T_n = {}^{0}A_1 \, {}^{1}A_2 \, \ldots \, {}^{n-1}A_n \tag{1}$$

where $^{i-1}A_i$ denotes the homogeneous transformation frames i relative to frame $i-1$.

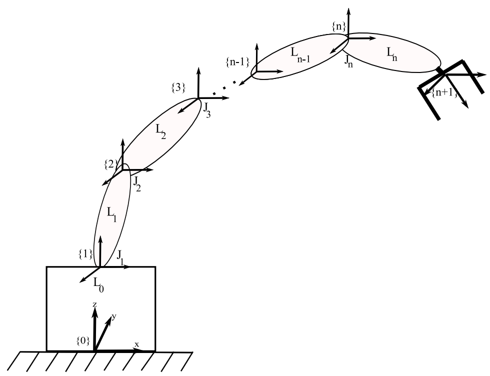

Figure 2: **Robot manipulator**

In order to describe the relationship between coordinate frames of a robot we can use either Denavit-Hartenberg (DH) notation [9] or modified Denavit-Hartenberg (MDH) one [19] which we use the latter to obtain the kinematic model of the KUKA robot. Then according to MDH, the transformation matrix $^{i-1}A_i$ describing frame i w.r.t $i-1$ is given as follows:

$$^{i-1}A_i = \begin{bmatrix} Cq_i & -Sq_i & 0 & d_i \\ C\alpha_i Sq_i & C\alpha_i Cq_i & -S\alpha_i & -r_i S\alpha_i \\ S\alpha_i Sq_i & S\alpha_i Cq_i & C\alpha_i & r_i C\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where S and C stand for functions "*sin*" and "*cos*" as well as d_i, α_i, q_i and r_i are MDH parameters of the link i of the robot.

4.2 Dynamic Model of the Robot

In order to derive dynamic equations of motion of a robot arm, usually three conventional formulation approaches can be used: Euler-Lagrange (EL) [20], recursive Newton-Euler (RNE) [23] and product of exponential (POE) [22] formulations. Using each of these methods, the dynamic model of a robot manipulator can be expressed in matrix form

$$\mathbf{M}\left(\mathbf{q}\left(t\right)\right)\ddot{\mathbf{q}}\left(t\right) + \mathbf{V}\left(\mathbf{q}\left(t\right), \dot{\mathbf{q}}\left(t\right)\right) + \mathbf{G}\left(\mathbf{q}\left(t\right)\right) + \mathbf{F}\left(\dot{\mathbf{q}}\right) = \boldsymbol{\tau}\left(t\right) \tag{3}$$

where $\mathbf{q}\left(t\right), \dot{\mathbf{q}}\left(t\right), \ddot{\mathbf{q}}\left(t\right)$, $\boldsymbol{\tau}\left(t\right)$ are $n \times 1$ vectors of joint variables, velocities, accelerations and torques, respectively. Moreover, $M\left(\mathbf{q}\right)$ is the $n \times n$ symmetric positive definite manipulator inertia matrix, $V\left(\mathbf{q}, \dot{\mathbf{q}}\right)$ is the n-vector containing centripetal and Coriolis terms and $\mathbf{G}\left(\mathbf{q}\left(t\right)\right)$ is gravity term and $\mathbf{F}\left(\dot{\mathbf{q}}\right)$ addresses the friction torque.

In order to obtain the dynamic equations of the robot, we developed a Robot Dynamics Modeler (RDM) GUI , as shown in Figure 3, by which the user can derive the dynamics equations of an open chain robot manipulator.

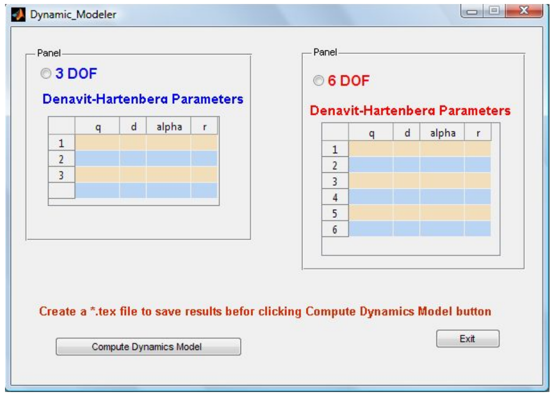

Figure 3: **Robot Dynamics Modeler (RDM) GUI**

5 Robot Identification

Robot identification procedure deals with the estimation of robot dynamic and friction parameters by means of commanding the robot with a particular trajectory, called *excitation trajectory*, and then measure the position and torque of each joint. These data are used in one of the following methods: Least square (LS) [21], weighted least square (WLS) [15, 13], Extended Kalman filter (EKF) [16], maximum likelihood or batch adaptive techniques, to estimate the dynamic and friction parameters of the robot. Eventually to verify the validation of estimated model, a new experiment is developed through which the robot is commanded to move along some different trajectory and then the torques produced by the robot controller are compared by those produced by the estimated model. This step is known as *identification validation*.

From identification point of view, there is a helpful property in the robot dynamic equations (3) which are linear in terms of dynamic and friction parameters of the robot, namely

$$\boldsymbol{\tau} = \boldsymbol{\tau_d} + \boldsymbol{\tau_f} = \mathbf{Y}\left(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}\right) \cdot \boldsymbol{\theta_d} + \mathbf{F}\left(\dot{\mathbf{q}}\right) \cdot \boldsymbol{\theta_f} \tag{4}$$

where

$$\begin{aligned} \boldsymbol{\theta_{d,i}} &= \left[I_{xxi}, I_{xy_i}, I_{xzi}, I_{yy_i}, I_{yz_i}, I_{zzi}, m_i\overline{x}_i, m_i\overline{y}_i, m_i\overline{z}_i, m_i, I_{ai}\right]^T \\ \boldsymbol{\theta_{f,i}} &= \left[\ F_{ci}, F_{vi}\right]^T \end{aligned} \tag{5}$$

The friction parameters can be estimated through a separate experiment in which a constant speed motion is used as excitation trajectory and then the parameters in $\boldsymbol{\theta_f}$ are estimated. On the other hand, only some of the elements in the vector $\boldsymbol{\theta_d}$ have really effect on the dynamic model of the robot. The set of such parameters are called **base parameters set** (BPS) which can be obtained by analytical or numerical approaches according to rules given in [12]. For KUKA robot, the number of BPSs together with friction parameters is 21. Furthermore, notice that the accuracy of the result of identification experiment is very dependent on excitation trajectory. It is a trajectory which “excites” all dynamics of the robot as well as, the sensitivity of the weighted least square method, which is used to estimate the base dynamic parameters, with respect to noise and model errors can be minimized along this trajectory [1, 8]. Excitation trajectory for a robot identification can be obtained by solving an optimization problem whose cost function is the condition number of robot observation matrix. Figure 4 shows the excitation trajectory for first three joints of KUKA robot. This robot has a SIMOTION control system (SCS) from the products of Siemens Industrial Automation which controls the robot. After calculating the excitation trajectories, the robot is commanded through SCS to move along these trajectories. The SCS software has measurement part which enables one to measure the desired signals. The SCS provides us the desired data in .xls format which is suitable to import to MATLAB for processing. Next, the data imported into MATLAB are used to remove their outliers. The

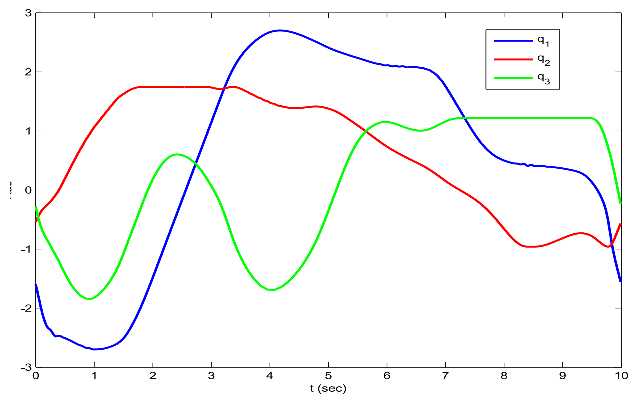

Figure 4: **Excitation trajectories of KUKA robot**

second step is to carry out the spectrum analysis on the measured joint positions, velocities and torques data. This analysis helps us to calculate the cutoff frequency of the filters to remove the noise on these data. We found that the original data are in the region before $20\,[Hz]$. For filtering the data obtained from previous steps, we can use the low-pass butterworth filters. In MATLAB, we can first use the function *buttord* to calculate the order and exact cutoff frequency of butterworth filter which were 7 and $21.4\,[Hz]$, respectively. These values together with obtained data from previous steps were input into functions *butter* and *filtfilt* to obtain the filtered data. Eventually, these processed data are used to estimate the dynamic and friction parameters thorough WLS, as given in tables 1 and 2, respectively. Moreover, figures 5 to 7 illustrates the results of identification validation for first three joints of KUKA robot.

Table 1: The value of the base dynamic parameters and their standard deviations of KUKA robot

Parameter	Estimated value	$\sigma_{\hat{\theta}_{bi}}$
θ_{b1}	31.95	0.25
θ_{b2}	38.53	0.92
θ_{b3}	-3.22	0.57
θ_{b4}	-0.675	0.88
θ_{b5}	-6.428	0.27
θ_{b6}	55.22	0.89
θ_{b7}	2.504	0.14
θ_{b8}	-33.9	0.65
θ_{b9}	1.347	0.47
θ_{b10}	1.7	0.25
θ_{b11}	-0.201	0.15
θ_{b12}	-0.47	0.27
θ_{b13}	20.697	0.35
θ_{b14}	4.717	0.57
θ_{b15}	-3.595	0.64

Table 2: The value of the estimated friction parameters and their standard deviations of KUKA robot

Parameter	Estimated value	$\sigma_{\hat{\theta}_{bi}}$
F_{c1}	34.995	0.57
F_{v1}	23.17	0.66
F_{c2}	65.338	0.48
F_{v2}	9.664	0.88
F_{c3}	32.7	0.54
F_{v3}	26.72	0.68

6 First Proposed Optimal Control Method

What we present in this section is a new method which solves the OCP of robot manipulators globally. Usually the existing methods result in a local optimal solution for this problem, obtained by fulfilling a

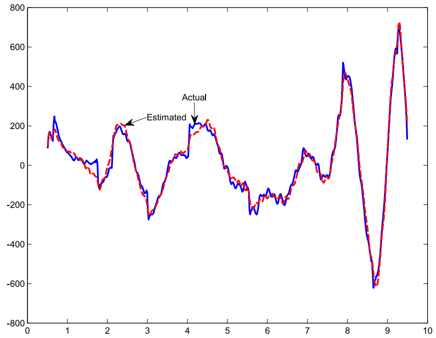

Figure 5: **Validation results for actual and estimated τ_1**

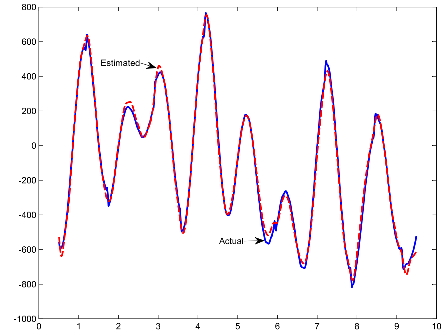

Figure 6: **Validation results for actual and estimated τ_2**

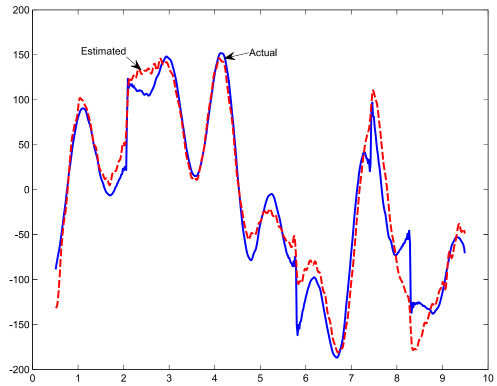

Figure 7: **Validation results for actual and estimated τ_3**

series of necessary conditions such as those presented in Ponntryagin's maximum principle or necessary KKT conditions in direct methods. This method, which is presented under a theorem, can be used for both set-point regulating tasks (e.g. pick and place parts or spot welding tasks) and trajectory tracking tasks such as painting or welding tasks. However, the proposed method has a limitation so that it can not support the physical constraints on robot manipulators.

6.1 New Proposed Theorem

In this subsection we present our first method to solve unconstrained OCP of robot manipulators which is formulated under the following theorem,

---

Theorem. Let A, B be positive definite matrices and let they be congruent, such that $B = P^T AP$. Suppose P is a stable matrix such that $P^T A = AP$. Then the criterion

$$J = \int_0^\infty \left(\dot{\xi}^T A \dot{\xi} + \xi^T B \xi\right) dt \tag{6}$$

has the global minimum value

$$J_{min} = \frac{1}{2}\xi^T(0)\; C\,\xi(0)$$

on the set of differentiable curves $\xi(t)$ such that $\lim_{t\to\infty} \xi(t) = 0$. The optimal solution is $\xi(t) = e^{Pt}\xi(0)$. The matrix C is $-2AP$.

6.2 Realization

For robot equation

$$M\ddot{q} + N\dot{q} + G = u \tag{7}$$

let us define a state

$$x = (q_1,\, \dot{q}_1,\, q_2,\, \dot{q}_2, \ldots,\, q_n,\, \dot{q}_n)^T. \tag{8}$$

Now define

$$e = x_d - x \tag{9}$$

where e is a $m \times 1$ vector with $m = 2n$. Let us study the problem to find a matrix T such that

$$\xi = -T\,e \tag{10}$$

where $\xi \in \mathbb{R}^{n\times 1}$ and $T \in \mathbb{R}^{n\times m}$. Our aim is to find a suitable form of matrix T.

From $\dot{\xi} = P\xi$ given in proof of the proposed theorem we obtain $T\dot{e} = PTe$ and so for $e(0) = e_0$ we have $\xi(0) = -Te_0$. Thus, we achieve the following matrix equation

$$T\dot{e} = PTe \tag{11}$$

which will play an important role. The state equation of (7) is

$$\dot{x} = f(x) + g(x)\,u \tag{12}$$

and so

$$\dot{e} = \dot{x}_d - \dot{x} = \dot{x}_d - f(x) - g(x)\,u,$$

where $x = x_d - e$. The equation (12) may be written as

$$\begin{aligned} \dot{x}_{i-1} &= x_i \\ \dot{x}_i &= f_i(x) + \sum_{j=1}^{n} g_{ij}(x)\,u_j \end{aligned}$$

for $i = 2, 4, 6, \ldots, m$, or we can write it as

$$\begin{aligned} \dot{x}_{2k-1} &= x_{2k} \\ \dot{x}_{2k} &= f_{2k}(x) + \sum_{j=1}^{n} g_{2k,j}(x)\,u_j \end{aligned} \tag{13}$$

for $k = 1, 2, 3, \ldots, n$.

6.2.1 Option of Matrix T

The equation (11) can be written as $(m > n)$

$$\begin{bmatrix} \sqsubset T_{11} & T_{12} \sqsupset & T_{13} & T_{14} & T_{15} & \cdots & & T_{1m} \\ T_{21} & T_{22} & \sqsubset T_{23} & T_{24} \sqsupset & T_{25} & \cdots & & T_{2m} \\ T_{31} & T_{32} & T_{33} & T_{34} & \sqsubset T_{35} & \cdots & & T_{3m} \\ \vdots & & & & & & & \\ T_{n1} & T_{n2} & & \cdots & & & \sqsubset T_{n,m-1} & T_{n,m} \sqsupset \end{bmatrix} \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \vdots \\ \dot{e}_m \end{bmatrix} = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ & & \\ & \vdots & \\ & & \\ P_{n1} & \cdots & P_{nn} \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1m} \\ T_{21} & T_{22} & \cdots & T_{2m} \\ T_{31} & T_{32} & \cdots & T_{3m} \\ \vdots & & & \\ T_{n1} & T_{n2} & \cdots & T_{nm} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_m \end{bmatrix} \tag{14}$$

The matrix T is $n \times m$ and we choose its quasi-diagonal as non-zero elements, others will be zero. The matrix P let be diagonal. Thus the matrix equation (14) can be rewritten in

$$\begin{aligned} & T_{11}\dot{e}_1 + T_{12}\dot{e}_2 = P_{11}T_{11}e_1 + P_{11}T_{12}e_2 \\ & T_{23}\dot{e}_3 + T_{24}\dot{e}_4 = P_{22}T_{23}e_3 + P_{22}T_{24}e_4 \\ & \vdots \\ & T_{n,m-1}\dot{e}_{m-1} + T_{nm}\dot{e}_m = P_{nn}T_{n,m-1}e_{m-1} + P_{nn}T_{nm}e_m \end{aligned} \tag{15}$$

From $\dot{x}_{2k-1} = x_{2k}$ in (13) we obtain equations

$$\dot{e}_1 = e_2,\ \dot{e}_3 = e_4,\ \cdots\ \dot{e}_{m-1} = e_m \tag{16}$$

and so we can rewrite (15) in the form

$$\begin{aligned} & T_{12}\dot{e}_2 = P_{11}T_{11}e_1 + (P_{11}T_{12} - T_{11})\, e_2 \\ & T_{24}\dot{e}_4 = P_{22}T_{23}e_3 + (P_{22}T_{24} - T_{23})\, e_4 \\ & \vdots \\ & T_{nm}\dot{e}_m = P_{nn}T_{n,m-1}e_{m-1} + (P_{nn}T_{nm} - T_{n,m-1})\, e_n \end{aligned} \tag{17}$$

The equation (16) and (17) we may write

$$\begin{aligned} & \dot{e}_{2k-1} = e_{2k} \\ & \dot{e}_{2k} = \frac{P_{kk}T_{k,2k-1}}{T_{k,2k}} e_{2k-1} + \left(P_{kk} - \frac{T_{k,2k-1}}{T_{k,2k}} \right) e_{2k} \end{aligned} \tag{18}$$

for $k = 1, 2, \cdots, n$.
Because here we have a fraction $\frac{T_{k,2k-1}}{T_{k,2k}}$, it will be better to choose

$$T_{k,2k} = 1 \tag{19}$$

and then for

$$M_k = \begin{bmatrix} 0 & 1 \\ & \\ P_{kk}T_{k,2k-1} & P_{kk} - T_{k,2k-1} \end{bmatrix} \tag{20}$$

we may rewrite (18) as

$$\begin{pmatrix} \dot{e}_{2k-1} \\ \dot{e}_{2k} \end{pmatrix} = M_k \begin{pmatrix} e_{2k-1} \\ e_{2k} \end{pmatrix} \tag{21}$$

<u>How it is with stability?</u>
Let us examine the eigenvalues of M_k as follows

$$\det\left(M_k - \lambda I\right) = \begin{vmatrix} -\lambda & 1 \\ P_{kk}T_{k,2k-1} & P_{kk} - T_{k,2k-1} - \lambda \end{vmatrix} = \lambda^2 - \lambda\left(P_{kk} - T_{k,2k-1}\right) - P_{kk}T_{k,2k-1}$$

The characteristic equation is $\det\left(M_k - \lambda I\right) = 0$, so

$$\left(\lambda - P_{kk}\right)\left(\lambda - T_{k,2k-1}\right) = 0.$$

Here we can write

$$\begin{aligned} \lambda_1 &= P_{kk} \\ \lambda_2 &= -T_{k,2k-1} \end{aligned}$$

$P_{kk}, T_{k,2k-1}$ are real numbers, thus from theory of stability $\lambda_1 < 0, \lambda_2 < 0$ and hence

$$\begin{aligned} P_{kk} &< 0 \\ T_{k,2k-1} &> 0 \end{aligned} \tag{22}$$

Therefore, matrix P must be stable which it is our assumption of the previous theorem. The matrix T then has the form

$$T = \begin{bmatrix} T_{11} & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & T_{23} & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & T_{35} & 1 & \cdots & 0 \\ \vdots & & & \vdots & & & & \vdots \\ 0 & & & & & & T_{n,m-1} & 1 \end{bmatrix} \tag{23}$$

The numbers $T_{k,2k-1}$ can be chosen arbitrary, but positive.

6.2.2 Solution of e_i and Optimal feedback Control

Let $a = P_{kk}T_{k,2k-1}$, $b = P_{kk} - T_{k,2k-1}$. Then characteristic equation of (18) is

$$\lambda^2 - b\lambda - a = 0 \Rightarrow \lambda_{1,2} = \frac{b \pm \sqrt{b^2 + 4a}}{2},$$

in which $b^2 + 4a = \left(P_{kk} - T_{k,2k-1}\right)^2 + 4P_{kk}T_{k,2k-1} = \left(P_{kk} + T_{k,2k-1}\right)^2 \geq 0$, so

$$\lambda_{1,2} = \frac{1}{2}\left(P_{kk} - T_{k,2k-1} \pm \left|P_{kk} + T_{k,2k-1}\right|\right) \Rightarrow \begin{cases} \lambda_1 = P_{kk} \\ \lambda_2 = -T_{k,2k-1} \end{cases}$$

Thus

$$\begin{aligned} e_{2k-1} &= c_{k1}e^{P_{kk}t} + c_{k2}e^{-T_{k,2k-1}t} \\ e_{2k} = \dot{e}_{2k-1} &= P_{kk}c_{k1}e^{P_{kk}t} - T_{k,2k-1}c_{k2}e^{-T_{k,2k-1}t} \end{aligned} \tag{24}$$

Eventually, from the equation

$$\dot{e} = \dot{x}_d - f\left(x\right) - g\left(x\right)u$$

From the equation

$$\dot{e} = \dot{x}_d - f\left(x\right) - g\left(x\right)u, \tag{25}$$

we have

$$\boxed{g(x)\ u = \dot{x}_d - \dot{e} - f(x)\,, \quad x = x_d - e.} \tag{26}$$

Now we can exploit two following ways to derive the optimal control u:

A. Multiplying (25) by T

$$T\,g(x)\ u = T(\dot{x}_d - f(x)) - T\dot{e}, \tag{27}$$

and according to (11) we will have

$$T\,g(x)\ u = T(\dot{x}_d - f(x)) - PTe, \tag{28}$$

Let us compute the multiplication $T\,g(x)$ in the left hand side of the above equation:

$$T \cdot g = \begin{bmatrix} T_{11} & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & T_{23} & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & T_{35} & 1 & \cdots & 0 \\ \vdots & & & \vdots & & & & \vdots \\ 0 & & & & & & T_{n,m-1} & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & \cdots & 0 \\ g_{21} & g_{22} & \cdots & g_{2n} \\ 0 & 0 & \cdots & 0 \\ g_{41} & g_{42} & \cdots & g_{4n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \\ g_{m1} & g_{m2} & \cdots & g_{mn} \end{bmatrix} = \begin{bmatrix} g_{21} & g_{22} & \cdots & g_{2n} \\ g_{41} & g_{42} & \cdots & g_{4n} \\ \vdots & \vdots & \vdots & \vdots \\ g_{m1} & g_{m2} & \cdots & g_{mn}. \end{bmatrix} \tag{29}$$

Let us denote

$$\tilde{g} = \begin{bmatrix} g_{21} & g_{22} & \cdots & g_{2n} \\ g_{41} & g_{42} & \cdots & g_{4n} \\ \vdots & \vdots & \vdots & \vdots \\ g_{m1} & g_{m2} & \cdots & g_{mn}. \end{bmatrix} \tag{30}$$

We see $\tilde{g}$ is a square matrix of type $n \times n$ and suppose $det\,\tilde{g} \neq 0$; hence it is a regular matrix and we can solve (28)

$$u(t) = (\tilde{g})^{-1}\,T(\dot{x}_d - f(x)) - (\tilde{g})^{-1}\,P\,T\,e \tag{31}$$

Therefore, in this way we obtained an optimal control of our problem by (31).

B. Remember, we can use the equation of robot motion (7) for establishing of the control $u(t)$. In fact, using (18) and (9) we obtained $x = x(t)$ and from ((8)) we can get $q(t)$ and $\dot{q}(t)$ and by derivative of $\dot{q}$ we have $\ddot{q}(t)$. If we substitute these results into ((7)), we obtain the control vector $u = u(t)$.

Remark 2. The method A is more general, because there is used the formula (12). So we employ only method A to obtain the optimal control $u(t)$.

Let us now consider two cases:

a. Let be given $e = e(t)$, for example by solving (21). Then by (9) we are able to find $x(t)$ and by a substitution into (31) we obtain the optimal control.

b. Contrarily, let be given an optimal control $u = u^*(t)$. Then by (12) we can compute $x(t)$ and then by substitution of these results into (25) we obtain a vector $\dot{e}(t)$, from which follows $e(t)$.

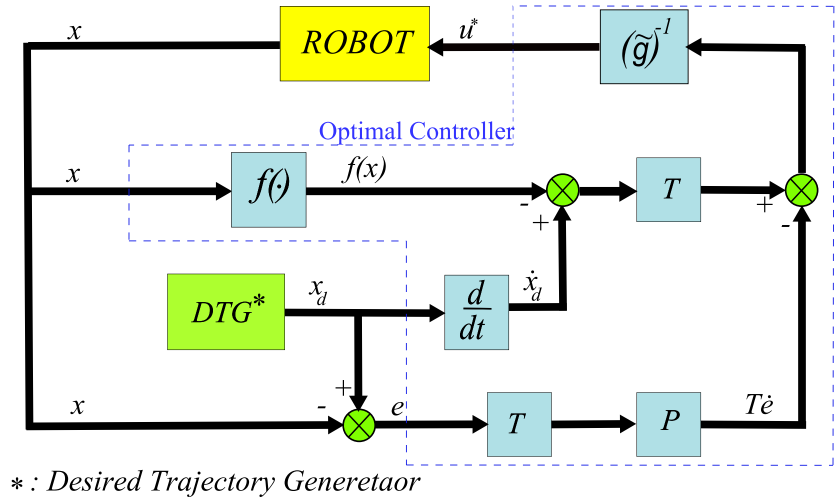

Figure 8: **Optimal feedback control schematic**

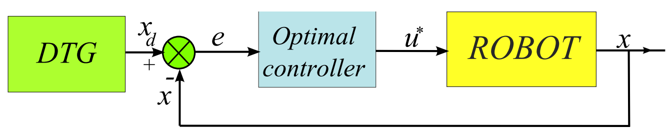

Figure 9: **Simplified optimal feedback control system**

In this manner we opened a way into the **optimal feedback control** for the precise model of robot (12). The feedback control is schematically depicted on the Figure 8. Really, we can mutually interchange the block ROBOT onto the relation (12), because all positions and motions are precisely described by (12) about our presumption. This schematic can be simplified as shown in Figure 9.

This schematic can be adjusted onto an adaptive control, if the equation (12) is not a precise model of robot motion.

Let us now apply this method into the KUKA robot whose dynamics were obtained as explained in sections 4 and 5. Because of space limitation in this text, we only present the optimal results of trajectory tracking case obtained by this method. Thus, let the following desired trajectories for the first three joints of the robot:

$$\begin{aligned} q_{d1} &= 0.3 + 0.1\sin(\pi\, t) \\ q_{d2} &= 0.8 + 0.2\sin(2\pi\, t) \\ q_{d3} &= 0.5 + 0.3\sin(3\pi\, t) \end{aligned} \tag{32}$$

The objective is to track the above desired trajectories by the robot so that at the same time the cost functional (6) is minimized. Thus, let the following matrices

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix},\ T = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 1 \end{bmatrix}$$

where A is a positive definite matrix and matrix T is chosen according to (23). As a result, the optimal trajectories (joint disposition, velocity and torque of first three joints) of KUKA robot are obtained by this method as shown in Figure 10 . Note that the robot is in its home position in $t = 0$. Of course, it can be in any other initial configuration. In this case the minimum value of the cost functional (6) is 50.138.

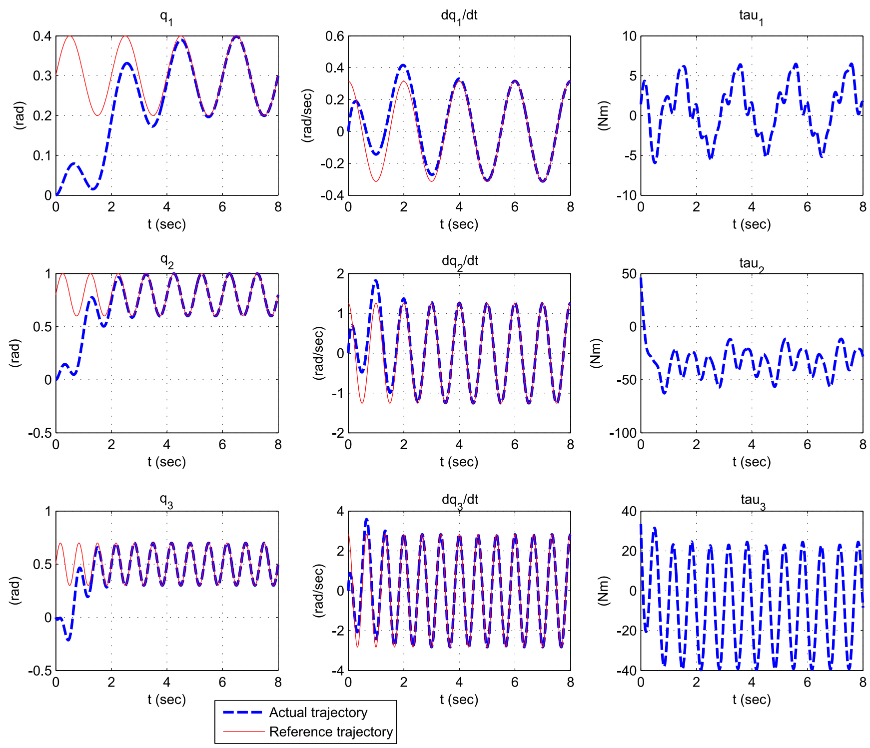

Figure 10: **Optimal trajectories of KUKA robot obtained by the proposed method in trajectory tracking case**

6.3 Adaptive Global Optimal Control

In the previous subsections we developed an unconstrained global optimal controller for robot manipulators. However, very often, there are some uncertainties in the dynamic model of the robot manipulators. One possibility in controlling such systems whose exact models are not available is *adaptive control* technique [2, 30].

In this subsection we attempt to extend our proposed controller in more general case in which an exact model of the considered robot does not exist. In fact, our objective is to design an *adaptive optimal controller* (AOC) whose central core is the optimal trajectory generator (OTG) proposed in the previous sections.

As explained earlier, the dynamic model of an n-axes robot manipulator can be expressed as the following form

$$M\left(q\right)\ddot{q} + N\left(q,\dot{q}\right)\dot{q} + G\left(q\right) = Y\left(q,\dot{q},\ddot{q}\right)\theta = \tau \tag{33}$$

where Y is an $n \times m$ matrix whose elements are nonlinear functions of $q, \dot{q}, \ddot{q}$ and $\theta \in \mathbb{R}^{m\times 1}$ is a vector whose entries are identifiable parameters of the considered robot. The elements of vector θ are functions of dynamic and friction parameters of the robot whose values usually are not provided by robot manufacturers and researchers have to measure these values themselves by robot identification experiments (as explained in section 5, in detail). Therefore, the uncertain state space representation of the robot can be written as the following form

$$\dot{x} = f\left(x,\hat{\theta}\right) + g\left(x,\hat{\theta}\right)u \tag{34}$$

Let us now consider the structure of the proposed AOC as illustrated in Figure 11 which actually is an adaptive self-tuning controller. In this structure the values of the elements of vector θ are estimated in an on-line manner and then the estimated control input of the robot ($\hat{\tau}$) is calculated on the basis of these estimated parameters in each time instance. Before explaining the internal structure of the on-line estimator, we require to introduce some variables in the proposed AOC:

- predicted torque defined as $\hat{\tau}\left(t\right) = Y(q\left(t\right), \dot{q}\left(t\right), \ddot{q}\left(t\right))\hat{\theta}\left(t\right)$

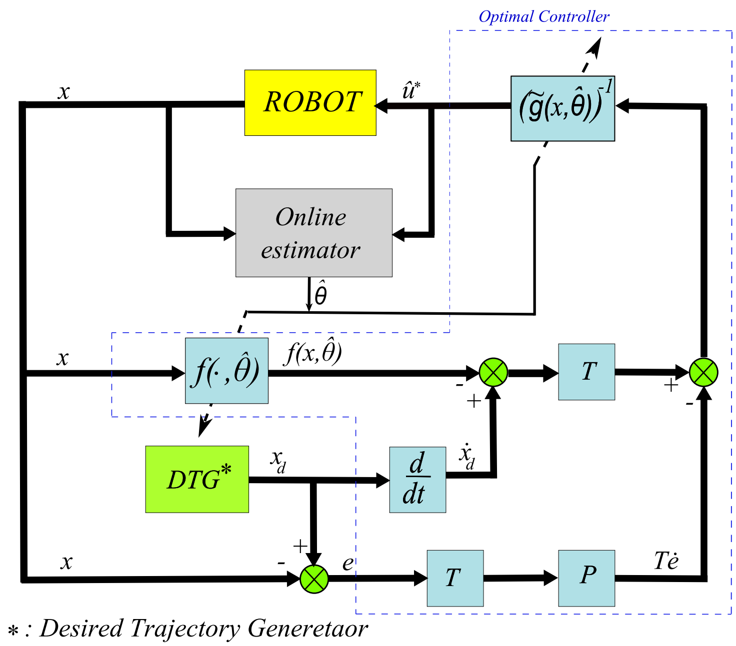

Figure 11: **General structure of adaptive optimal controller of robot manipulators**

- exact value of unknown parameters, denoted θ
- parameter estimation error defined as $\tilde{\theta} = \hat{\theta} - \theta$
- prediction error defined as $\mathbf{e}(t) = Y\hat{\theta}(t) - Y\theta(t)$

We can use the procedure used in the standard least-square (LS) scenario to design the on-line estimator. In fact, the estimation of the parameters can be obtained by minimizing the following total prediction error with respect to $\hat{\theta}$

$$J = \int_0^t \left\| \tau(s) - Y(q(s), \dot{q}(s), \ddot{q}(s))\,\hat{\theta}(s) \right\|^2 ds. \tag{35}$$

According to LS method, the solution of the above minimization problem is obtained as follows:

$$\hat{\theta}(t) = \left[\int_0^t Y^T Y ds\right]^{-1} \int_0^t Y^T \tau(s)\, ds. \tag{36}$$

However, in computation point of view, the equation (36) is not efficient and it can be converted into a more appropriate form with defining the following square matrix

$$\Phi(t) = \left[\int_0^t Y^T Y ds\right]^{-1}. \tag{37}$$

whose derivative is

$$\frac{\mathrm{d}\left[\Phi^{-1}(t)\right]}{\mathrm{d}t} = Y^T(q(t), \dot{q}(t), \ddot{q}(t))\; Y(q(t), \dot{q}(t), \ddot{q}(t)). \tag{38}$$

Let us now consider the following identity

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\Phi\Phi^{-1}\right] = \dot{\Phi}\Phi^{-1} + \Phi\frac{\mathrm{d}}{\mathrm{d}t}\left[\Phi^{-1}\right] = 0, \tag{39}$$

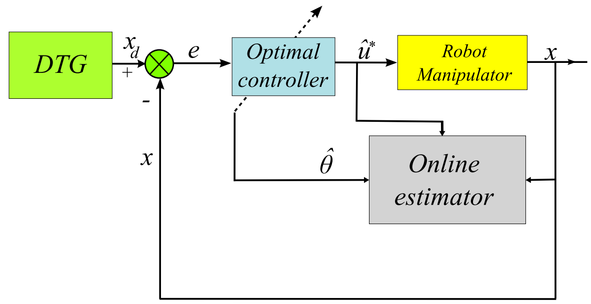

Figure 12: **Simplified adaptive optimal control system for robot manipulators**

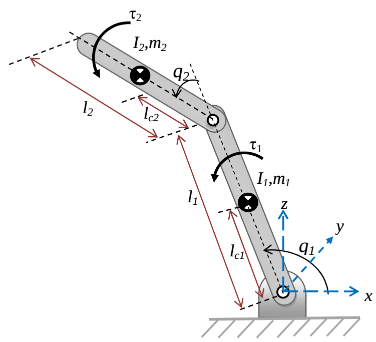

Figure 13: **Vertical two links robot manipulator**

and so

$$\boxed{\dot{\Phi} = -\Phi Y^T Y \Phi}. \tag{40}$$

Eventually, the unknown parameters can be updated by the following causal equation which is obtained by differentiating (36) and using (37)

$$\boxed{\dot{\hat{\theta}} = -\Phi\left(t\right) W^T \mathbf{e}}. \tag{41}$$

For investigating the convergence of the above on-line estimator, it is easy to obtain the following equation using equations (38) to (41)

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\Phi^{-1}\left(t\right)\tilde{\theta}\left(t\right)\right] = 0, \tag{42}$$

and hence

$$\tilde{\theta}\left(t\right) = \Phi\left(t\right)\Phi^{-1}\left(0\right)\tilde{\theta}\left(0\right). \tag{43}$$

Therefore, if smallest eigenvalue of the integral $\int_0^t Y^T\, Y\, ds$ (according to 37) goes to infinity as $t \to \infty$, then in (43) $\Phi \to 0$ and so $\tilde{\theta} \to 0$ and each trajectory that satisfies this condition is called *persistent excitation trajectory*. It is worth to be noted also that, according to (43), if the initial value $\Phi\left(0\right)$ is large enough, then it results in smaller parameter error. Note that the structure given in Figure 11 can be simplified as shown in Figure 12

Let us apply the proposed AOC into a vertical two links robot, shown in Figure 13, whose regressor dynamics is as follows

$$\begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} & Y_{15} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} & Y_{25} \end{bmatrix} \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix} = \begin{bmatrix} \tau 1 \\ \tau 2 \end{bmatrix} \tag{44}$$

where

$$\begin{aligned} Y_{11} =& l_1^2 \ddot{q}_1 + g l_1 \cos(q_1), \\ Y_{12} =& 2 l_1 \cos(q_2) \ddot{q}_1 + l_1 \cos(q_2) \ddot{q}_2 + g \cos(q_1 + q_2) - l_1 \sin(q_2) \left(2 \dot{q}_1 \dot{q}_2 + \dot{q}_2^2\right), \\ Y_{13} =& \ddot{q}_1,\ Y_{14} = \ddot{q}_1 + \ddot{q}_2,\ Y_{15} = g \cos(q_1), \\ Y_{21} =& 0,\ Y_{22} = l_1 \cos(q_2) \ddot{q}_1 + l_1 \sin(q_2) \dot{q}_1^2 + g \cos(q_1 + q_2),\ Y_{23} = 0, \\ Y_{24} =& \ddot{q}_1 + \ddot{q}_2,\ Y_{25} = 0, \end{aligned}$$

and

$$\theta_1 = m_2,\ \theta_2 = m_2\, l_{c2},\ \theta_3 = I_1 + m_1\, l_{c1}^2,\ \theta_4 = I_2 + m_2\, l_{c2}^2,\ \theta_5 = m_1\, l_{c1}.$$

so that according to Table 3, the exact values of these parameters are $\theta_1 = 1$, $\theta_2 = 0.2$, $\theta_3 = 0.68$, $\theta_4 = 0.54$, $\theta_5 = 0.6$. The objective is that the joints of the robot track the following desired trajectories:

Table 3: Some typical function spaces

m_1	m_2	l_1	l_2	l_{c1}	l_{c2}	I_1	I_2
2 kg	1 kg	0.6 m	0.4 m	0.3 m	0.2 m	o.5 $kg \cdot m^2/rad$	o.5 $kg \cdot m^2/rad$

$$\begin{aligned} q_{d1} &= 0.3 + 0.1 \sin(\pi\, t) \\ q_{d2} &= 0.8 + 0.2 \sin(2\pi\, t) \end{aligned}$$

Therefore, assuming a disturbance as $d(t) = o.5 \sin(50t)$ in the system, the optimal trajectories are obtained as shown in Figure 14. In this figure the blue dashed trajectories are actual ones while the trajectories with continuous line are desired trajectories. In addition, Figure 15 depicts the estimation parameters $\hat{\theta}_1$ to $\hat{\theta}_5$ existed in dynamic model of the system.

7 Second Proposed Optimal Control Method

7.1 Formulation of Robot Optimal Control Problem

As stated in subsection 4.2, dynamics of the serial robot manipulators can be expressed as follows:

$$M(\mathbf{q})\, \ddot{\mathbf{q}} + N(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau},\ q(t_0) = q_0, q(t_f) = q_f \tag{45}$$

where

$$N(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{V}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) + \mathbf{G}(\mathbf{q}(t)) + \mathbf{F}(\dot{\mathbf{q}}) \tag{46}$$

In order to obtain the state space representation of the robot, let us define position/velocity state $x \in \mathbb{R}^{2n}$ as

$$x = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \tag{47}$$

where n is the number of robot's DOF. Then, the state space representation of the system (45) may be expressed as

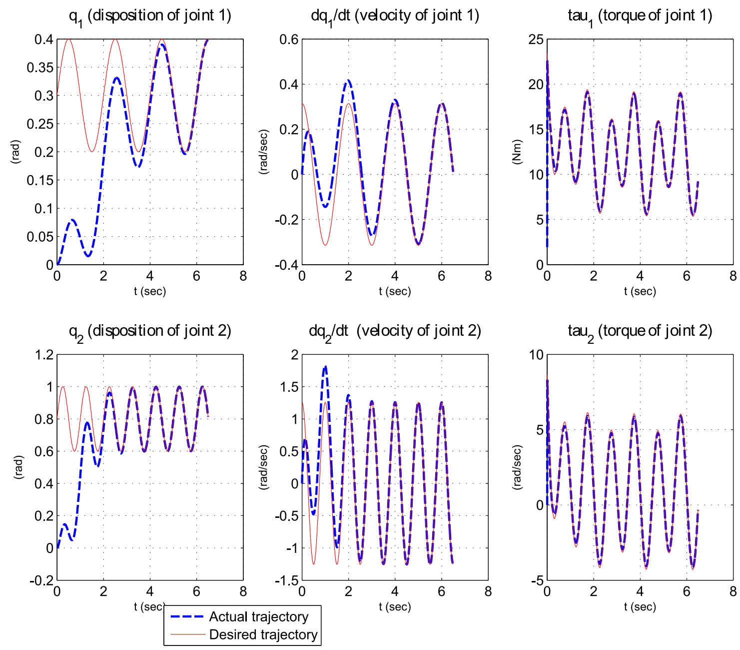

Figure 14: **Optimal trajectory of two links robot obtained by applying AOC**

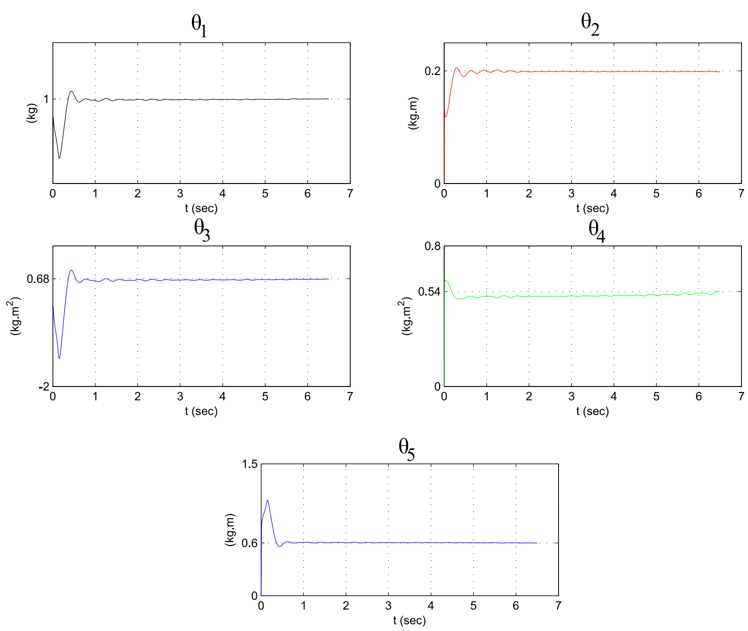

Figure 15: **Estimated parameters of two links robot**

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\boldsymbol{\tau},\ x(t_0) = x_0,\ x(t_f) = x_f \tag{48}$$

where

$$f(\mathbf{x}) = \begin{bmatrix} \mathbf{x_2} \\ -M^{-1}(\mathbf{x_1}) N(\mathbf{x_1}, \mathbf{x_2}) \end{bmatrix},\ g(x) = \begin{bmatrix} 0 \\ M^{-1}(\mathbf{x_1}) \end{bmatrix} \tag{49}$$

Now the optimal control of the robot manipulators may be formulated as follows:
find an admissible input torque vector which steers the robot system (48) at a finite time in such a way that the cost function (performance criterion) given by

$$J = \phi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \boldsymbol{\tau}(t), t)\, dt \tag{50}$$

is minimized, while the following constraints are met:

$$\begin{aligned} &\mathbf{x}(0) - x_0 = 0, \text{ initial state constraints} \\ &\mathbf{x}(t_f) - x_f = 0, \text{ final state constraints} \\ &C(\mathbf{x}(t), \boldsymbol{\tau}(t)) \leq 0, \text{ state \& control constraints} \end{aligned} \tag{51}$$

It must be noted that the final time t_f can be either fixed or free depending on the kind of the performance index used; for instance, in the case of time OCP, t_f must be free and in other cases it is usually fixed.

7.2 Proposed Method for Solving OCP of Robot Arms

The basic idea of the proposed method is to combine three techniques including Iterative Linearization (IL), Iterative Learning Control (ILC) and Parametric Optimization (PO) methods. Let a robot which is performing a repeated task such as pick and place parts in an assembly line. According to this method, the optimal control for nonlinear system (robot) is computed in several repetitions (trials). In other words, in each trial the IL method obtains a linear time varying (LTV) version of the nonlinear system (robot) and simultaneously, an optimal control input is computed for each LTV by the parametric optimal control technique. In each trial, also, the optimal solution of LTVs are stored in memory of the system because according to IL method they will be used in the next trial. In fact, the optimal control of nonlinear system is improved in each trial by means of optimal solution obtained from previous trial; hence, this procedure is compatible with the performance of ILC.

Let us now consider the IL method in more detail. According to this method the state space representation of the nonlinear system (48) may be replaced by the following form, called state dependent coefficient (SDC) form

$$\dot{\mathbf{x}} = A(\mathbf{x})\mathbf{x} + B(\mathbf{x})\mathbf{u},\ \ \mathbf{x}(0) = x_0, x(t_f) = x_f \tag{52}$$

where $u = \tau$ and

$$A(x) = \nabla_x f(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix},\ B(x) = g(x) \tag{53}$$

where without loss of generality, it is assumed $x_e = 0$ is the equilibrium point of the system. Thus, the dynamic system (52) can be approximated by a sequence of LTV systems in the form

$$\begin{aligned} &\dot{x}^{[i]}(t) = A\left(x^{[i-1]}(t)\right) x^{[i]}(t) + B\left(x^{[i-1]}(t)\right) u^{[i]}(t), \\ &x^{[i]}(t_0) = x_0,\ x^{[i]}(t_f) = x_f,\ i = 1, 2, \ldots \end{aligned} \tag{54}$$

where $x^{[0]}(t)$ can be chosen as an arbitrary initial guess and for convenience it may be considered as initial condition, i.e., x_0. In fact, the LTV (54) is used as the robot dynamics in trial ith and its optimal control $u^{*[i]}(t)$ is computed by PO method. Also according to the proposed method, the optimal state of trial i, i.e. $x^{*[i]}(t)$ together with optimal control $u^{*[i]}(t)$ and an error signal as $e^{[1]}(t) = u^{*[1]}(t) - u^{*[0]}(t)$ are stored in memory of the system. In the next subsection, we will show how the optimal control input is computed for each LTV system (54) and then the proposed algorithm will be presented.

7.3 Parametric Optimization Method

In this subsection the parametric optimization method used in the proposed method is presented. The basic idea in the parametric optimal control method is that the original infinite dimension optimal control problem is converted into a finite dimension optimization problem (known as constrained nonlinear programming (CNP)) which can be solved by standard optimization algorithms like sequential quadratic programming (SQP). The parametrization is usually made using the spline functions which can be either polynomial spline or B-spline functions; since, they have sufficient attributes to converge to a good optimal solution in CNPs, as well as they cause faster convergence rate and are appropriate to approximate the practical trajectories for robot manipulators. Note that in the polynomial spline functions, the coefficients of polynomials are parameters and in the B-splines, the control points are used as parameters which need to be optimized.

In the case of OCP of robot manipulators stated in section 7.1, the parametrization procedure is performed by considering the following polynomial spline function, as each joint position of the robot:

$$x_{1j}(t) = q_j(t) = \begin{cases} s_0(t) & t_0 \le t < t_1 \\ s_1(t) & t_1 \le t < t_2 \\ \vdots & \vdots \\ s_\ell(t) & t_\ell \le t < t_{\ell+1} = t_f \end{cases} \qquad j = 1, 2, \dots, n \tag{55}$$

where s_i is a cubic spline function defined by

$$s_i(t) = p_{1i}(t - t_i)^3 + p_{2i}(t - t_i)^2 + p_{3i}(t - t_i) + p_{4i} \tag{56}$$

for $i = 0, 1, \dots, \ell$. According to (56), the joint velocity and acceleration trajectories can be expressed as

$$\begin{aligned} \dot{s}_i(t) &= 3p_{1i}(t - t_i)^2 + 2p_{2i}(t - t_i) + p_{3i} \\ \ddot{s}_i(t) &= 6p_{1i}(t - t_i) + 2p_{2i} \end{aligned} \tag{57}$$

In order to apply these cubic spline functions as the joint position trajectories of the robot, the following conditions should be satisfied:

- the initial and final state constraints represented in (51) should be met.
- $q_j(t)$ should be continuous on the interval $[t_0, t_{\ell+1} = t_f]$.
- $\dot{q}_j(t)$ should be continuous on the interval $[t_0, t_{\ell+1} = t_f]$.
- $\ddot{q}_j(t)$ should be continuous on the interval $[t_0, t_{\ell+1} = t_f]$.

After fulfilling the conditions above, among the coefficients p_{1i}, p_{2i}, p_{3i} and p_{4i} some of them are dependent while the rest are independent. It can be shown that the number of independent coefficients is $n\ell$ which are used as the parameters in the CNP mentioned above. After applying the joint position, velocity and acceleration terms represented above, into each LTV (54), the control input can be obtained in terms of independent parameters. If we then substitute parametric joint, velocity and control input trajectories into the cost function (50), we obtain a parametric function $J(\mathbf{p})$ where $\mathbf{p}$ stands for independent parameters. Such substitutions can be repeated for constraints (51) and eventually we obtain a constrained optimization programming (CNP) which should be solved. One of the useful tools to solve CNPs is the optimization toolbox of MATLAB specifically the function *fmincon*. Although the parametrization in robot OPC usually is done by polynomial spline functions, however, they have a property which is significant in the case of moving of robot manipulator. This property is that the changing a parameter affects on entire shape of the spline curve, while we need to alter a part of curve. This problem can be resolved using B-spline functions which have the following form

$$q\left(t, \mathbf{P}\right) = \sum_{j=0}^{m} p_j B_{j,k}\left(t\right) \tag{58}$$

where $\mathbf{P} = \{p_0, \ldots, p_m\}$ are the control points (which are used as parameters in parametrization procedure) and k is the order of the B-spline. In (58) a knot space is defined as

$$I = \{t_0 \leq \cdots \leq t_{k-1} \leq t_k \leq \cdots \leq t_m \leq t_{m+1} \leq \cdots \leq t_{m+k}\} \tag{59}$$

as well as $B_{j,k}\left(t\right)$ is the *B-spline basis function* expressed as the following recursive formulation:

$$B_{j,k}\left(t\right) = \frac{t - t_j}{t_{j+k-1} - t_j} B_{j,k-1}\left(t\right) + \frac{t_{j+k} - t}{t_{j+k} - t_{j+1}} B_{j+1,k-1}\left(t\right) \tag{60}$$

where

$$B_{j,1}\left(t\right) = \begin{cases} 0 & \text{if } t_j \leq t < t_{j+1} \\ 1 & \text{otherwise} \end{cases} \tag{61}$$

In the case of robot manipulator's OCP, usually the cubic B-spline functions ($k = 4$) is used to parametrize the joint variables and the set of control points $\mathbf{P}$ are employed as parameters in the optimization problem [4].

7.4 The Proposed Algorithm

Let us now combine the mentioned techniques, i.e. IL, ILC and PO methods to solve the OCP of the robot manipulators.

First, consider a robot with following dynamics which performs a special task repeatedly

$$M\left(q\right)\ddot{q} + V\left(q, \dot{q}\right) + G\left(q\right) = \tau \tag{62}$$

with the boundary conditions

$$\begin{gathered} q\left(0\right) = q_0, \dot{q}\left(0\right) = qd_0 \\ q\left(T\right) = q_T, \dot{q}\left(T\right) = qd_T. \end{gathered} \tag{63}$$

By considering the following states

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \tag{64}$$

the state space representation of the robot can be written as follows

$$\dot{x}\left(t\right) = f\left(x\left(t\right)\right) + g\left(x\left(t\right)\right)u\left(t\right), \;\; x\left(0\right) = x_0, \;\; x\left(T\right) = x_T \tag{65}$$

which can be rewritten as the following SDC form

$$\dot{x} = A\left(x\right)x + B\left(x\right)u \tag{66}$$

where, without loss of generality, $x_e = 0$ is its equilibrium point and

$$A\left(x\right) = \nabla_x f\left(x\right), \; B\left(x\right) = g\left(x\right), \; u = \tau \tag{67}$$

Then the matrices A and B are achieved as follows

$$A\left(x\right) = \begin{bmatrix} Z_{n\times n} & I_{n\times n} \\ -\nabla_x\left(M^{-1}\left(x_1\right)N\left(x_1, x_2\right)\right) \end{bmatrix}, \;\; B = \begin{bmatrix} Z_{n\times n} \\ -M^{-1}\left(x_1\right) \end{bmatrix} \tag{68}$$

where Z and I are zero and identity matrices, respectively, with the specified dimensions. Now we are going to obtain the optimal control of the considered robot which is performing the desired repeated task. Therefore, the following linear dynamics is considered as the model of the robot in the first trial

$$\dot{x}^{[1]}(t) = A\left(x^{[0]}(t)\right)x^{[1]}(t) + B\left(x^{[0]}(t)\right)u^{[1]}(t), \quad x^{[1]}(0) = x_0, \ x^{[1]}(T) = x_T \tag{69}$$

where the cost functional considered in this trial is

$$J^{[1]} = \phi\left(x^{[1]}(T), T\right) + \int_0^T L\left(x^{[1]}(t), u^{[1]}(t), t\right) dt. \tag{70}$$

As explained in the previous subsection, the above OCP can be solved by parameterizing states of the system by spline functions $S(t;P)$. After solving the obtained parametric optimization problem, the optimal parameter matrix $P^{*[1]}$ is obtained for the first trial. Accordingly, the optimal control of the first trial is obtained as follows

$$u^{[1]}(t) = \begin{bmatrix} Z_{n\times n} & M\left(x_1^{[0]}(t)\right) \end{bmatrix} \left(\begin{bmatrix} \dot{x}_1^{[1]}(t) \\ \dot{x}_2^{[1]}(t) \end{bmatrix} - \begin{bmatrix} Z_{n\times n} & I_{n\times n} \\ -\nabla_x\left(M^{-1}\left(x_1^{[1]}\right)N\left(x_1^{[0]}, x_2^{[0]}\right)\right) \end{bmatrix} \begin{bmatrix} x_1^{[1]} \\ x_2^{[1]} \end{bmatrix} \right) \tag{71}$$

where $x_1^{[1]}(t) = S\left(t; P^{*[1]}\right)$ and $x_2^{[1]}(t) = \dot{x}_1^{[1]}(t) = \dot{S}\left(t; P^{*[1]}\right)$. In addition, let us define an error variable as

$$e^{[1]}(t) = u^{*[1]}(t) - u^{*[0]}(t) \tag{72}$$

where it is assumed $u^{*[0]}(t) = 0$. The optimal state $x^{*[1]}$ and control $u^{*[1]}$ together with $e^{[1]}$ are stored in memory of the system. Other variable stored in memory of the system from first trial is first order optimality, denoted $\delta^{[1]}$. It is a variable produced by nonlinear programming algorithm which actually shows the variation of the cost functional, i.e. $\delta J^{[1]}$. If $u^{*[1]}$ is optimal solution, then $\delta^{[1]}$ must vanish on $u^{*[1]}$.

As such in the first trial, the above procedure is performed in the subsequent trials so that in the trial ith, the optimal state and control of trial $i-1$th is used

$$\dot{x}^{[i]}(t) = A\left(x^{[i-1]}(t)\right)x^{[i]}(t) + B\left(x^{[i-1]}(t)\right)u^{[i]}(t), \quad x^{[i]}(0) = x_0, \ x^{[i]}(T) = x_T \tag{73}$$

with considering the following cost functional in this trial

$$J^{[i]} = \phi\left(x^{[i]}(T), T\right) + \int_0^T L\left(x^{[i]}(t), u^{[i]}(t), t\right) dt \tag{74}$$

Let us now define two predetermined constants ε_1, ε_2 which are close to zero and are used as the stop criteria of the proposed algorithm.

Here, we present the proposed algorithm whose steps, as demonstrated in Figure 16, are listed as follows:

1. Obtain the state space representation and then the SDC form of the considered robot manipulator system.

2. Get the initial and final configurations (q_0, q_T);

3. Guess an arbitrary state $x^{[0]}(t)$, $t \in [0, T]$, and store it in memory of the system.

   Let the iteration index i be one.

4. Using $x^{[i-1]}(t)$ and utilizing spline-based optimal control technique explained in the previous subsection, compute the optimal force/torque vector $u^{*[i]}(t)$ and optimal state vector $x^{*[i]}(t)$ of the LTV system in step i represented in (73) given a cost functional, physical constraints of the robot and boundary conditions. Also store $x^{*[i]}(t)$ and $u^{*[i]}(t)$ in memory of the system together with $\delta^{[i]}$.

5. Apply $u^{[i]}(t)$ to the ith trial.

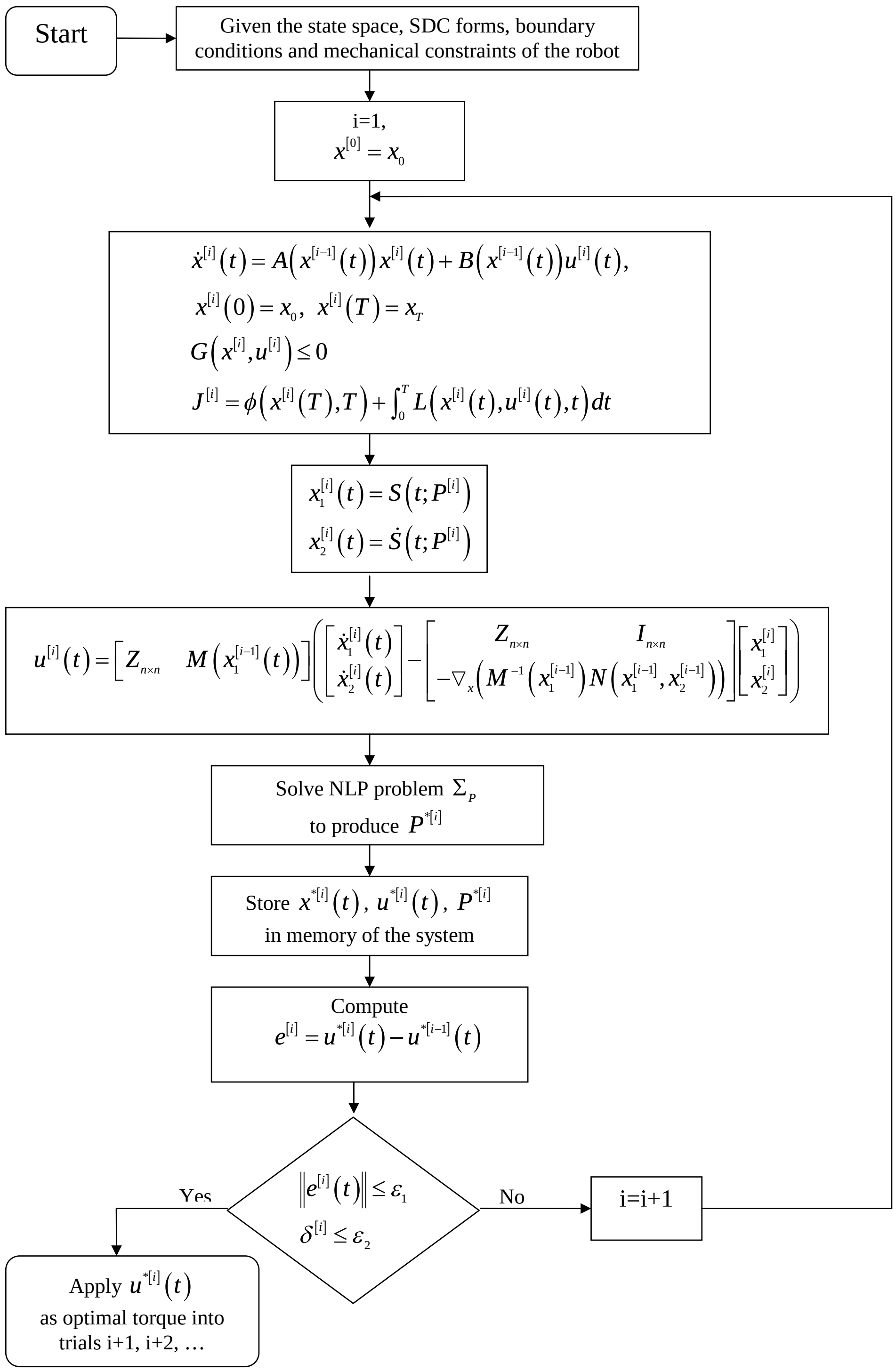

Figure 16: **Flowchart of the proposed method**

Table 4: Joint position, velocity and acceleration constraints of KUKA robot

Joint	$q_i\,[deg]$	$\lvert\dot{q}_i\rvert\,[deg/s]$	$\lvert\ddot{q}_i\rvert\,\left[deg/s^2\right]$	$\tau_i[N.m]$
1	±155	151	450	550
2	100 to -55	151	450	550
3	70 to -220	151	450	550

6. If

$$\left\| e^{[i]}\left(t\right)\right\| \le \varepsilon_1 \;\text{ and }\; \delta^{[i]} \le \varepsilon_2 \tag{75}$$

then terminate the computations and $u^{[i]}\left(t\right)$ can be used for the next trials. If stopping criteria given in (75) is not satisfied, then $i = i + 1$ and return to step 4.

7.5 Applying proposed method into KUKA robot

As explained in the previous sections, the dynamic model of the KUKA robot was obtained through an experimental identification. In this subsection we are going to obtain the optimal trajectories of this robot according to the joint position, velocity, acceleration and torque constraints of this robot listed in Table 4 [17]. For this problem, we consider the following cost function:

$$J_c = \frac{1}{2}\int_0^T \left(\boldsymbol{\tau}^T R\boldsymbol{\tau} + \dot{\mathbf{q}}^T Q\dot{\mathbf{q}}\right) dt \tag{76}$$

where R and Q are symmetric positive definite weighting matrices. In this case study we use the following diagonal matrices R, Q

$$R = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 1 \end{bmatrix},\; Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{77}$$

The optimal results are shown in Figure 17, considering the following boundary conditions:

$$\begin{aligned} & q_1\left(0\right) = q_2\left(0\right) = q_3\left(0\right) = 0, \\ & \dot{q}_1\left(0\right) = \dot{q}_2\left(0\right) = \dot{q}_3\left(0\right) = 0, \\ & q_1\left(T\right) = 50\left(deg\right),\; q_2\left(T\right) = 25\left(deg\right),\; q_3\left(T\right) = 30\left(deg\right), \\ & \dot{q}_1\left(T\right) = \dot{q}_2\left(T\right) = \dot{q}_3\left(T\right) = 0. \end{aligned} \tag{78}$$

This figure shows that the optimal controls of KUKA robot converge after 8 trials. Also Table 5 represents some information regarding different trails of the optimization procedure for the KUKA robot. As this table shows, in the first trial the minimum traversal time is 4.63 sec which results in a minimum cost function equals 148.78. However, in the subsequent trials a trade-off is made between these two values so that from trial forth the value of T_{min} is fixed. Moreover, we can obtain the necessary information about the rate of convergence in this case study by referring to Figure 18. We can use the sequence of error norms to obtain the rate of convergence. According to the value of elements of these three sequences (i.e. $\left\| e_j^{[i]} \right\|$ for $j = 1, 2, 3$), their convergence rates are $\mu_1 = \frac{1}{3}, \mu_2 = \frac{1}{6}, \mu_3 = \frac{1}{4}$, respectively, according to the following theorem and remark,

Theorem. Let the sequence $\left\{\zeta\,\mu^k\right\}_{k=0}^{\infty}$ where ζ is a constant. This sequence converges linearly to zero with rate μ if $|\mu| < 1$.

Remark. If $\{a_k\}_{k=0}^{\infty}$ be a sequence and $a_k \le \zeta\,\mu^k$, then $\{a_k\}$ converges to zero with at most rate μ.

Table 5: Optimal data obtained for KUKA robot

Trial	Number of SQP iterations	Number of math operations	Time of computation	J_{min}	T_{min}	First order Optimality (δ)
1	22	302	2.85	148.78	4.643	151
2	17	271	2.68	122.63	5.38	38.7
3	15	258	2.61	124.87	5.42	23.9
4	15	259	2.6	126.83	5.647	2.88
5	15	262	2.62	126.8	5.647	0.43
6	14	257	2.5	125.792	5.647	0.0636
7	12	245	2.38	125.78	5.647	0.0022
8	11	235	2.34	125.627	5.647	00.000127

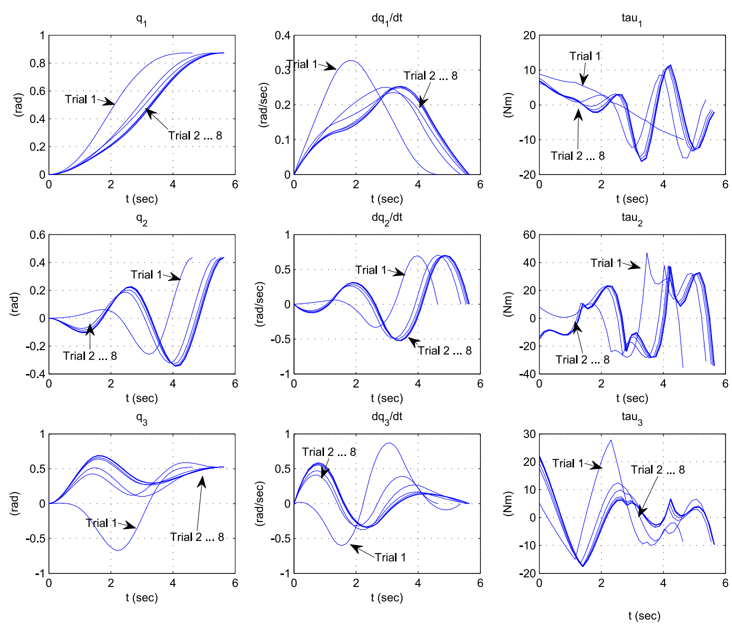

Figure 17: **Optimal profiles of KUKA robot**

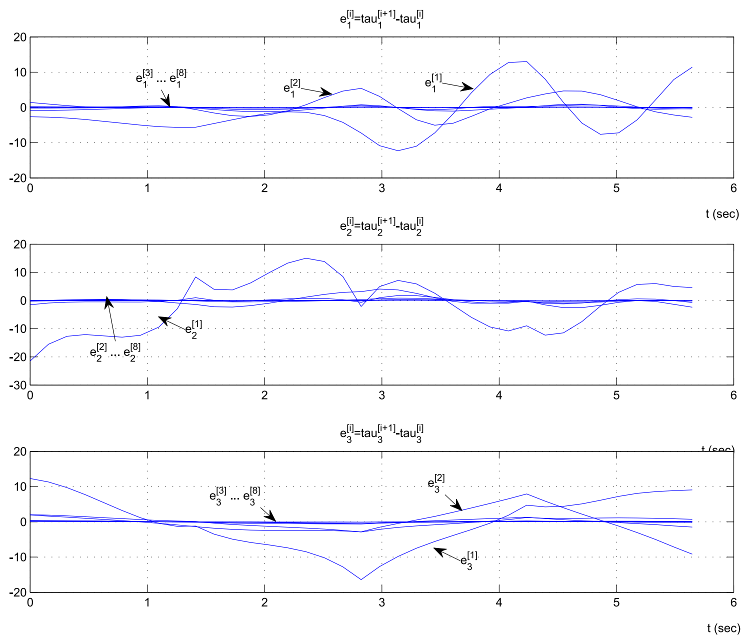

Figure 18: **Successive errors of KUKA robot joints**

8 Conclusion

Motivation of this study is to propose a new method to optimal control of serial robot manipulators. Let us now consider the fulfillment of the considered objectives, stated in the section 3 as follows:

Objective 1: The first objective is to obtain the kinematic and dynamic models of our main case study, i.e., KUKA IR 364/10 robot manipulator existed in robotic laboratory of Mechatronic faculty of TUL. The kinematic model of this robot was derived employing modified Denavit-Hartenberg (MDH) notation. In the context of dynamic model of this KUKA robot, we first develop an algorithm using recursive Newton-Euler formulation. This algorithm was used as the main core of a GUI by which user can derive the dynamic model of either 3 or 6 degrees of freedom robot manipulators by entering just robot MDH parameters. A writing task was used to verify the validation of obtained kinematic and dynamic models of the KUKA robot in comparison with these models produced by Robotic Toolbox of MATLAB (RTM).

Objective 2: The second objective is KUKA robot identification. In doing so, in the first stage a new (regression) model of the robot dynamics which is linear in terms of a new set of parameters, so-called base parameter set (BPS), which are compound of dynamic and friction parameters of the robot is derived. In the second stage an excitation trajectory is calculated. This trajectory which is calculated from an optimization problem has a considerable affect on the identification result and hence this stage must be carried out with high attention. Eventually the elements of BPS for this KUKA robot which contains 21 parameters are estimated and a validation stage is accomplished to verify the obtained model.

Objective 3: The third objective considered in this thesis is to solve unconstrained OCP of robot manipulators. So as to achieve this objective we present a completely innovative and new approach to solve this problem in the case of point to point motion and trajectory tracking tasks. Unlike the existing methods which yield a local optimal solution, the proposed method solves the considered optimal control problem with obtaining a global optimal solution so that the computation time to find this solution is less than 0.01 sec. noting that the robot dynamics is highly nonlinear and coupled. However, this method can not support any physical constraints on a robot arm. The proposed method which is a model-based controller was extended into a more general case in which an exact model of the robot is not available, namely designing an *adaptive optimal control* scheme for robot manipulators.

Objective 4: The fourth objective is to propose a new method to solve the constrained time-energy optimal control problem of serial robot manipulators. we propose a combined method which contains Iterative Linearization (IL), Iterative Learning Control (ILC) and Parametric Optimization (PO). In this method it is assumed that the robot is performing a repeated task which is usual for robot arms in their applications. In accordance with this method, in each repetition (trial) a linear time varying (LTV) version of robot dynamics is derived by IL with the original considered cost functional. Then PO is used to solve the optimal control problem in this trial and its solution is stored in memory of the system to use in the next trial (ILC). The above procedure is repeated in the next trials so that after a finite number of trials the sequence of optimal solutions converge to the optimal solution of the original nonlinear system (robot dynamics). Then, the limit of the sequence is used to control of the next trials. The corresponding developed algorithm was applied into all standard types of robot arm structures, i.e. SCARA, spherical, cylindrical and angular robots (such as Puma 560, ABB IRB140 and KUKA IR 364/10 manipulators) for the different case of cost functionals. For having a better insight regarding the proposed method, the optimal solution of the considered optimal control problem for SCARA, spherical and cylindrical robots are obtained by direct multiple shooting and spline-based optimal control methods as well. Then, a series of comparisons are made between the proposed method and the other two methods. According to these comparisons, the following results were obtained for the proposed method:

- In each trial a linear version of highly nonlinear robot dynamics is dealt with.
- Optimization problem is solved gradually during the the successive trials. In other words, as shown by the optimal data given in tables of different case studies in chapter 5, the number of math operations and computation time to find the optimal solution are divided on successive trials.
- The convergence rate of the sequence of optimal solutions is too fast, as shown in various case studies.
- It supports any type of cost functions (quadratic, non-quadratic, linear, nonlinear and so on) and any kind of constraints.
- It generates the smooth trajectory for robot motions causing reduction the stresses to the actuators and to the manipulator structure.
- The possibility to set the initial and final joint accelerations and jerks a priori by the user.
- Unlike the multiple shooting method which produces a constant piecewise control, the proposed method provides a continuous optimal control which can be implemented in practice.
- The structure of the proposed optimal control system is almost simple and it can be implemented easily.

9 References

References

[1] B. Armstrong. On finding exciting trajectories for identification experiments involving systems with non-linear dynamics. *Proceeding IEEE International Conference on Robotics and Automation*, pages 1131–1139, 1987.

[2] A. Astolfi, D. Karagiannis, and R. Ortega. *Nonlinear and Adaptive Control with Applications.* Springer, London, 2008.

[3] T. Balkan. A dynamic programming approach to optimal control of robotic manipulators. *Mechanics research Communication*, 25:225–230, 1998.

[4] J. E. Bobrow. Optimal robot path planning using the minimum time criterion. *IEEE Journal of Robotics and Automation*, 4:443–450, 1988.

[5] J. E. Bobrow, S. Dubowsky, and J. S. Gibson. Time optimal control of robot manipulators along specified paths. *The International Journal of Robotic research*, 4(3):3–17, 1985.

[6] J.E. Bobrow. *Optimal Control of Manipulators.* PhD thesis, University of California, Los Angeles, 1982.

[7] H.G. Bock, , and K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceeding of 9th IFAC World Congress*, pages 243–247, Budapest, 1984.

[8] G. Calafiore, M. Indri, and B. Bona. Robot dynamic calibration: Optimal excitation trajectories and experimental parameter estimation. *Journal of Robotic Systems*, 18(2):55–68, 2001.

[9] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Trans ASME Journal of Applied Mechanics*, 23:215–221, 1955.

[10] M. Diehl, H.G. Bock, H. Diedam, and P.B Wieber. Fast direct multiple shooting algorithms for optimal robot control. *NCIS, Fast Motions in Biomechanics and Robotics*, 340:65–93, 2006.

[11] C.R. Dohrmann and R.D. Robinett. Robot trajectory planning via dynamic programming. In *Proceedings of the Fifth International Symposium on Robotics and Manufacturing, and Applications*, pages 75–81, 1994.

[12] E. Dombre and W. Khalil. *Modeling, Performance Analysis and Control of Robot Manipulators.* London, 2007.

[13] P. Dutkiewicz, K. R. Kozlowski, and W. S. Wroblewski. Experimental identification of robot and load dynamic parameters. *Second IEEE Conference on Control Applications*, 2:767–776, 1993.

[14] A. Gasparetto and V. Zanott. A technique for time-jerk optimal planning of robot trajectories. *Robotics and Computer-Integrated Manufacturing*, 24:415426, 2008.

[15] M. Gautier. Dynamic identification of robots with power model. *International Conference on Robotics and Automation*, 3:1922 – 1927, 1997.

[16] M. Gautier and P. Poigent. Extended kalman filtering and weighted least squares dynamic identification of robot. *Control Engineering Practice*, 9:1361–1372, 2001.

[17] KUKA Schweissaniagen-Roboter GmbH. *KUKA IR 364/10.0-VK 10 Technical Data*, 1995.

[18] Y. Guan, K. Yokoi, O. Stasse, and A. Kheddar. On robotic trajectory planning using polynomial interpolations. *2005 IEEE International Conference on Robotics and Biomimetics*, pages 111–116, 2006.

[19] W. Khalil and J.F. Kleinfinger. A new geometric notation for open and closed loop robots. *IEEE International Conference on Robotics and Automation.*, pages 1174–1179, 1986.

[20] F.L. Lewis, D.M. Dawson, and C.T. Abdallah. *Robot Manipulator Control Theory and Practice.* Marcel Dekker Inc., 2004.

[21] L. Ljung. *System Identification, Theory for the User.* Prentice-Hall, New Jersey, second edition edition, 1999.

[22] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, California, 1994.

[23] R. P. Paul. *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press, Massachusetts, 1981.

[24] P. Pedel and Y. Bestaoui. Actuator constraints in optimal motion planning of manipulators. In *IEEE International Conference on Robotics and Automation*, pages 2427–2432, Nagoya, Japan, 1995.

[25] Z. Shiller. On singular points and arcs in path constrained time optimal motions. *ASME Dynamic System Control*, 42:141–147, 1992.

[26] Z. Shiller. Time-energy optimal control of articulated systems with geometric path constraints. In *IEEE International Conference on Robotics and Automation*, San Diego, USA, 1994.

[27] Z. Shiller and S. Dubowsky. On the optimal control of robot manipulators with actuator and end-effector constraints. *IEEE International Conference in Robotic and Automation*, pages 614–620, 1985.

[28] Z. Shiller and H. Lu. Computation of path constrained time optimal motions with dynamic singularities. *Journal of Dynamic Systems*, 114:34–40, 1992.

[29] K.G. Shin and N.D. Mckay. A dynamic programming approach to trajectory planning of robotic manipulators. *IEEE Trans. On Automatic Control*, 6:491–500, 1986.

[30] J. J. E. Slotine and W. Li. *Applied Nonlinear Control.* Prentice Hall, New Jersey, 1991.

[31] S. F. P. Sraramago and V. Steffen. Optimization of the trajectory planning of robot manipulators taking into account the dynamic of the system. *Mechanism and Machine Theory*, 33(7):141–147, 1998.

[32] S. F. P. Sraramago and V. Steffen. Optimization of the trajectory planning of robot manipulators taking into account the dynamic of the system. *Mechanism and Machine Theory*, 35(8):1079–1094, 2000.

[33] V. Záda. Optimal control of robots. In *9th International Conf. on Computer, Communication and Control Technology CCCT'03*, pages 330–335, 2003.

Author Publications

- Conference:

[1]- **A. Chatraei**, Vision based control of a robot manipulator to swing-up and stabilizing control of an inverted pendulum, *International Conference on Automation, Robotics and Control Systems*, Orlando, USA, pp. 97-102, 2009.

[2]- V. Záda, **A. Chatraei**, and D. Linder, Learning control for robot manipulators, *Engineering Mechanics Conference*, Svratka, Czech Republic, pp. 1489-1494, 2009.

[3]- V. Záda,**A. Chatraei**, and D. Linder, Analytical solution of inverse task with sixth rotational joints and its concrete utilization, *Engineering Mechanics Conference, Svratka*, Czech Republic, pp. 1495-1502, 2009.

[4]- **A. Chatraei**, M. A. Zanjani, M. Montazeri, and M. Sanatizadeh, Prototype design and fabrication Of an intelligent surveillance and guard robot, *Second National Electrical Conference (NEEC2010)*, Iran, Najafabad , February, 2010.

[5]- **A. Chatraei**, Solving position kinematics Of ABB IRB 140 industrial robot and its motion planning, *Second National Electrical Conference (NEEC2010)*, Iran, Najafabad , February, 2010.

- Journal:

[6]- **A. Chatraei**, and V. Záda, A combined optimal control technique for robot manipulators, *International Scientific Journal Acta Technica*, reviewing (will be published in Apr. 2011).

[7]- **A. Chatraei**, V. Záda, and D. Lindr, Modeling and identification of positioning part of the KUKA robot, *International Scientific Journal Acta Technica*, reviewing.

[8]- **A. Chatraei**, and V. Záda, Global optimal control of robots, International Asian Journal of Control, (Submitted recently).