ISSN 2347-1921

Ill-Conditioning in Matlab Computation of Optimal Control with Time-Delays

Eihab B.M. Bashier
Department of Applied Mathematics, Faculty of Mathematical Science, University of Khartoum, Sudan.

ABSTRACT

A direct transcription method transforms an optimal control problem (OCP) into a nonlinear programming problem (NLP). The resulting NLP can be solved by any NLP solver, such as the Matlab's optimization toolbox, the fsqp, etc.

On solving optimization problems using Matlab, the Matlab's optimization toolbox does not obtain an accurate Hessian matrix at the optimal solution due to the fact that the Hessian matrix is not being evaluated directly from the optimal solution.

In this paper we compute the condition numbers associated with the optimal control computation, where the classical fourth-order Runge-Kutta method is used for the discretization of the state equations. The computations of optimal solutions are done for different numbers of switching points and quadrature points per a switching interval.

A test example showed that the condition numbers of the active constraints, projected Hessian and the whole Lagrangian system are more likely to grow with the number of the switching intervals per a delay interval than by the number of the quadrature intervals per a switching interval. Also, the three medium scale optimization algorithm of the Matlab's optimization toolbox give almost similar condition numbers when used to solve an optimal control problem.

Indexing terms/Keywords

Ill-conditioning; KKT system; active constraints; Hessian matrix; QR-factorization.

Academic Discipline And Sub-Disciplines

Optimal control, Nonlinear programming, Optimization and Linear algebra.

AMS CLASSIFICATION (2010)

34H05, 49J15, 49K21, 49M05, 65K05

Council for Innovative Research
Peer Review Research Publishing System
Journal: JOURNAL OF ADVANCES IN MATHEMATICS
Vol .11, No.2
www.cirjam.com , editorjam@gmail.com

1. INTRODUCTION

Optimal control problems with time delays in state and control have wide applications in the real-life applications [1]. Some of these applications include the control of infectious diseases [2, 3], the continuous stirred tank reactor (CSTR) [4, 5], biological populations [6], population harvesting [1], etc.

We consider a general optimal control problem with a discrete time delay of the form:

$$\min_{\vec{u}\in U} \quad J(u(t)) = \phi(\vec{x}(t_f)) + \int_{t_0}^{t_f} L_0(t, \vec{x}(t), \vec{x}(t-\tau), \vec{u}(t), \vec{u}(t-\tau))dt, \tag{1}$$

subject to dynamics governed by the state equations

$$\dot{\vec{x}}(t) = \vec{f}(t, \vec{x}(t), \vec{x}(t-\tau), \vec{u}(t), \vec{u}(t-\tau)), \; t \in [t_0, t_f], \tag{2}$$

with initial data given by

$$\left.\begin{aligned} &\vec{x}(t_0) = \vec{x}^{\,0}, \\ &\vec{x}(t) = \vec{\varphi}(t), \; t \in [t_0 - \tau, t_0], \\ &\vec{u}(t) = \vec{\vartheta}(t), \; t \in [t_0 - \tau, t_0], \end{aligned}\right\} \tag{3}$$

where $\vec{x}, \vec{\varphi} : R \to R^n$, $\vec{u}, \vec{\vartheta} : R \to R^m$ and $\vec{\varphi}(t)$ and $\vec{\vartheta}(t)$ are given piecewise continuous functions.

The system is subject to continuous state inequality constraints

$$\vec{I}(t, \vec{x}(t), \vec{x}(t-\tau), \vec{u}(t), \vec{u}(t-\tau)) \le \vec{0}, \; t \in [t_0, t_f]. \tag{4}$$

It is subject to equality constraints

$$\vec{E}(t, \vec{x}(t), \vec{x}(t-\tau), \vec{u}(t), \vec{u}(t-\tau)) = \vec{0}, \; t \in [t_0, t_f], \tag{5}$$

and subject to terminal conditions:

$$\vec{\psi}(t_f, \vec{x}(t_f)) = \vec{\psi}_f \tag{6}$$

where the functions $\vec{I} : R \times R^n \times R^n \times R^m \times R^m \to R^p$ and $\vec{E} : R \times R^n \times R^n \times R^m \times R^m \to R^q$ are differentiable with respect to $\vec{x}$ and $\vec{u}$. The function $\vec{\varphi} : R^n \to R$ and $\vec{\psi} : R \times R^n \to R^l$ are differentiable with respect to each component of $\vec{x}$.

Direct transcription methods are well-known class for solving optimal control problems. Direct transciption methods are also known as discretize then optimize methods, because the problem discretization is a prior stage to the optimization process [7]. Methods of discretization such as the Runge-Kutta methods [8], splines [9], collocation methods [10], etc; are used for the discretization of the state equations, whereas some numerical quadrature method such as the trapezoidal rule or the Simpson's rule is used for the evaluation of the objective function.

One of the efficient direct transciption methods for solving the optimal control problems, is the control parameterization technique [11, 12, 13]. In a CPT, the time domain is partitioned by a number of switching points at which the control variables are evaluated, then each switching interval is partitioned by a number of quadrature points at which the state variables are evaluated. At each switching interval, the control variable is approximated by a constant or linear piece-wise continuous function [11].

At the end of the discretization stage, the optimal control problem is transformed into a large or medium scale finite dimensional nonlinear programming problem (NLP) [14]. The resulting NLP can be solved by using any nonlinear programming software, such as the Matlab's optimization toolbox [15], the SQP [16], the FSQP [17], etc.

The MATLAB's function *fmincon* returns the optimal solution $\vec{x}$, which contains the state and control variables, the value of the objective function *fval* at the optimal solution $\vec{x}$, the exit flag, the gradient vector, the Hessian matrix and the Lagrange multipliers.

The optimization toolbox in MATLAB uses the quasi Newton methods (BFGS, DFP) to solve the nonlinear programming problems [15]. If $\vec{x}^{\,0}$ denotes the initial guess for the optimal solution $\vec{x}^{\,*}$, the optimization process starts with the identity matrix I as an initial guess for the Hessian matrix H^*. Then the Hessian matrix is updated at each

ISSN 2347-1921

iteration, maintaining its positive-definiteness, to guarantee that, the direction of search $\vec{p}$ is always a descent direction. The optimization process terminates, when the directional derivative $\nabla\vec{f}^T\vec{p}$ is less than a given tolerance *FunTol*, and the maximum constraint violation is less than another given tolerance *ConTol*. The default value of both the tolerences *FunTol* and *ConTol* is 10^{-7}. Associated with the fmincon Matlab's function are three medium scale optimization algorithms, namely, the *active-set* algorithm, the *trust-region-reflective* algorithm and the *SQP* algorithm. A fourth algorithm that is used with the fmincon function for solving the large-scale nonlinear programming problems is the *interior-point* algorithm.

Generally, optimal control problems are by nature ill-conditioned problems [18]. That is if a small perturbation to the constraints of the problem or to the objective function is occured, then large variations on the parameters (Lagrange multipliers and search directions) of the system will follow. Methods for measuring condition numbers associated to optimal control computations without time delays were developed by Benyah and Jenenings [19, 20].

Matveev [21] showed that a small time delay cannot be neglected in a general optimal control problem, and at the same time it results in an ill-posed model. In this paper, we compute the condition numbers that are associated to Matlab's medium scale algorithms when solving an optimal control problem with a time delay using the control parameterization technique.

We organize the rest of this paper as follows. In Section 2 we state the optimal control problem under consideration and describe the discretization of the problem which transcripts it into a nonlinear programming problem. In Section 3 is the description of the ill-conditioning in constrained optimization. In Section 4 are two test examples. In Section 5, are the conclusions.

2. ILL-CONDITIONING IN CONSTRAINED OPTIMIZATION

We consider an optimization problem of the form:

$$\min_{\vec{x}\in R^n} \vec{f}(\vec{x}), \tag{7}$$

subject to:

$$\vec{c}(\vec{x}) = \vec{0}, \tag{8}$$

where $\vec{c}(\vec{x}) = [c_1(\vec{x}), c_2(\vec{x}), \ldots, c_p(\vec{x})]^T$, and $p \le n$.

The Lagrangian system is given by:

$$L(\vec{x},\vec{\lambda}) = \vec{f}(\vec{x}) + \vec{\lambda}^T\vec{c}(\vec{x})$$

Then the first-order necessary conditions for $(\vec{x}^*,\vec{\lambda}^*)$ to be optimum are given by

$$\left.\begin{array}{l} \nabla_{\vec{x}} L(x,\lambda) = 0 = \vec{f}_{\vec{x}}(\vec{x}^*) + \nabla\vec{c}^T(\vec{x})\cdot\vec{\lambda}^*, \\ \nabla_{\vec{\lambda}} L(x,\lambda) = 0 = \vec{c}(\vec{x}), \end{array}\right\} \tag{9}$$

where $\nabla\vec{c}^T(\vec{x})$ is the Jacobian matrix, evaluated at $\vec{x}$.

The second-order necessary condition is that

$$\nabla^2_{\vec{x}\vec{x}} L(\vec{x}^*,\vec{\lambda}^*) = \vec{f}_{\vec{x}\vec{x}}(\vec{x}^*) + \sum_{i=1}^{p} \lambda_i^* \nabla^2_{\vec{x}\vec{x}}\vec{c}(\vec{x}) = H(\vec{x}^*) \in R^{n\times n} \tag{10}$$

be *positive semi-definite*.

The second order sufficient condition for $(\vec{x}^*,\vec{\lambda}^*)$ to be optimum is

$$\nabla^2_{\vec{x}\vec{x}} L(\vec{x}^*,\vec{\lambda}^*) = \vec{f}_{\vec{x}\vec{x}}(\vec{x}^*) + \sum_{i=1}^{p} \lambda_i^* \nabla^2_{\vec{x}\vec{x}}\vec{c}(\vec{x}^*) = H(\vec{x}^*) \in R^{n\times n} \tag{11}$$

be *positive definite*.

We assume that, $\nabla\vec{c}^T(\vec{x}) \in R^{n\times p}$ is of full rank. Then $\nabla\vec{c}^T(\vec{x})$ has a QR-factorization of the form:

$$\nabla \vec{c}^{T}(\vec{x}) = [Q' \quad Z]\begin{bmatrix} R \\ 0 \end{bmatrix} \tag{12}$$

where $Q' \in R^{n\times p}$, $R \in R^{p\times p}$ and $Z \in R^{n\times(n-p)}$. Q' is a basis for the column space of $\nabla \vec{c}^{T}(\vec{x})$, and Z is a basis for the null space of $\nabla \vec{c}^{T}(\vec{x})$.

At the optimum solution $\vec{x}^{*}$, the gradient of the objective function $\vec{f}^{*} = \vec{f}(\vec{x}^{*})$ is orthogonal to the constraints surface. That is, the projection of the gradient vector onto the constraint surface is zero. This can be expressed by $Z^{T}\nabla\vec{f}^{*}(\vec{x}) = 0$, which is an equivalent formula for the first order necessary conditions. The equivalent second order necessary condition for the optimality of $(\vec{x}^{*}, \vec{\lambda}^{*})$ is that, $Z^{*T} H Z^{*} \in R^{(n-p)\times(n-p)}$ be positive semi-definite, and the equivalent second order sufficient condition for the optimality of $(\vec{x}^{*}, \vec{\lambda}^{*})$ is that $Z^{*T} H Z^{*}$ be positive definite.

The vector $Z^{T}\nabla\vec{f}^{*}(\vec{x})$ is the *projected gradient* and the matrix $Z^{*T} H Z^{*}$ is the *projected Hessian*.

The Newton's methods work to find the couple $(\vec{x}^{*}, \vec{\lambda}^{*})^{T} \in R^{(n+p)\times(n+p)}$ such that the necessary conditions (9)-(10) of optimality are satisfied. Given $(\vec{x}^{(k)}, \vec{\lambda}^{(k)})$ at the k^{th} iteration, the Karush-Kuhn-Tucker (KKT) can be used to obtain $(\vec{x}^{(k+1)}, \vec{\lambda}^{(k+1)})$ as follows:

$$\begin{bmatrix} H(\vec{x}^{(k)}) & \nabla \vec{c}^{T}(\vec{x}^{(k)})^{T} \\ \nabla \vec{c}^{T}(\vec{x}^{(k)}) & 0 \end{bmatrix}\begin{bmatrix} p^{(k)} \\ \lambda^{(k+1)} \end{bmatrix} = \begin{bmatrix} -\nabla \vec{f}(\vec{x}^{(k)}) \\ -\vec{c}(\vec{x}^{(k)}) \end{bmatrix}, \tag{13}$$

where $\vec{p}^{(k)} = \vec{x}^{(k+1)} - \vec{x}^{(k)}$.

Let

$$\nabla \vec{c}^{T}(\vec{x}^{(k)}) = [Q'^{(k)} \quad Z^{(k)}]\begin{bmatrix} R^{(k)} \\ 0 \end{bmatrix} \tag{14}$$

be the QR-factorization of the matrix $\nabla \vec{c}^{T}(\vec{x}^{(k)})$. Define a matrix $Q^{(k)}$ by:

$$Q^{(k)} = \begin{bmatrix} [Q'^{(k)} \ Z^{(k)}] & 0 \\ 0 & I \end{bmatrix} \in R^{(n+p)\times(n+p)}, \tag{15}$$

where $I \in R^{p\times p}$ is the identity matrix. Partition $p^{(k)}$ into $p_1^{(k)}$ and $p_2^{(k)}$, where $p_2^{(k)} \in R^{n-p}$ and $p_1^{(k)} \in R^{p}$. Then,

$$Q^{(k)T}\begin{bmatrix} H(\vec{x}^{(k)}) & \nabla \vec{c}^{T}(\vec{x}^{(k)})^{T} \\ \nabla \vec{c}^{T}(\vec{x}^{(k)}) & 0 \end{bmatrix} Q^{(k)} Q^{(k)T}\begin{bmatrix} \vec{p}(k) \\ \vec{\lambda}^{(k+1)} \end{bmatrix} = Q^{(k)T}\begin{bmatrix} -\nabla \vec{f}(\vec{x}^{(k)}) \\ -\vec{c}(\vec{x}^{(k)}) \end{bmatrix}, \tag{16}$$

can be written as

$$\begin{bmatrix} Q'^{(k)T} H^{(k)} Q'^{(k)} & Q'^{(k)} H^{(k)} Z^{(k)} & R^{(k)} \\ (Q'^{(k)T} H^{(k)} Z^{(k)})^{T} & Z^{(k)T} H^{(k)} Z^{(k)} & 0 \\ R^{(k)T} & 0 & 0 \end{bmatrix}\begin{bmatrix} Q'^{(k)T} \vec{p}_1^{(k)} \\ Q'^{(k)T} \vec{p}_2^{(k)} \\ \vec{\lambda}^{(k+1)} \end{bmatrix} = \begin{bmatrix} -Q'^{(k)T}\nabla \vec{f}(\vec{x}^{(k)}) \\ -Z^{(k)T}\nabla \vec{f}(\vec{x}^{(k)}) \\ -\vec{c}(\vec{x}^{(k)}) \end{bmatrix} \tag{17}$$

Let $\vec{w}_1^{(k)} = Q'^{(k)T} \vec{p}_1^{(k)}$ and $\vec{w}_2^{(k)} = Q'^{(k)T} \vec{p}_2^{(k)}$, then solving the linear system $R^{(k)T} w_1^{(k)} = -\vec{c}(\vec{x}^{(k)})$ will have the will have the condition number $\chi(R^{(k)})$. As $Q'^{(k)}$ is orthogonal, then $\chi(R^{(k)}) = \chi(\nabla \vec{c}^{T}(\vec{x}^{(k)}))$ gives the condition number of the active constraints. Also, solving the linear system

ISSN 2347-1921

$Z^{(k)^T} H^{(k)} Z^{(k)} \vec{w}_2^{(k)} = -Z^{(k)^T} \nabla \vec{f}(\vec{x}^{(k)}) Z^{(k)} - (Q^{'(k)^T} H^{(k)} Z^{(k)})^T \vec{w}_1^{(k)}$ for $\vec{w}_2^{(k)}$ will have a condition number $\chi(R^{(k)}) \chi(Z^{(k)^T} H^{(k)} Z^{(k)})$. Since $(\vec{w}_1^{(k)}, \vec{w}_2^{(k)}) = Q\vec{p}^{(k)}$, then $\chi(R^{(k)}) \chi(Z^{(k)^T} H^{(k)} Z^{(k)})$ gives the condition number of the system parameters, which can be affected by perturbations in the matrix $R^{(k)^T}$. Finally, the new values of the Lagrange multipliers $\vec{\lambda}^{(k+1)}$ are obtained by solving the system

$$R^{(k)} \vec{\lambda}^{(k+1)} = -Q^{'(k)^T} \nabla \vec{f}(\vec{x}^{(k)}) - Q^{'(k)^T} H^{(k)} Q^{'(k)} w_1^{(k)} - Q^{'(k)^T} H^{(k)} Z^{(k)} w_2^{(k)} \tag{18}$$

and will have a condition number $\chi(R^{(k)})^2 \chi(Z^{(k)^T} H^{(k)} Z^{(k)})$. This gives the condition number of the Lagrange multipliers. The two sub-matrices $Q^{'(k)^T} H^{(k)} Q^{'(k)}$ and $Q^{'(k)} H^{(k)} Z^{(k)}$ can lessen or worsen the condition number of the Lagrange multipliers. This tells that the Lagrange multipliers are most likely to be effected by perturbations than the solution vector, see [22].

If the accuracy level for the Kuhn-Tucker conditions is chosen to be $\varepsilon > 0$ and for the constraints is chosen to be $\delta > 0$, then, in the neighbourhood of the solution, $\vec{x}^{(k)}$ is expected to change as much as

$$\left\| \vec{x}^{(k+1)} - \vec{x}^{(k)} \right\| < \chi(R^{(k)^T}) \left(\delta + \chi(Z^{(k)^T} H^{(k)} Z^{(k)}) \varepsilon \right) \tag{19}$$

and the Kuhn-Tucker gradient $\nabla \vec{f}(\vec{x}^{(k)}) + H\vec{p}^{(k)} = \zeta$ can have a norm as large as

$$\left\| \zeta \right\| < \delta \chi \left((R^{(k)^T})^2 \cdot \chi(Z^{(k)^T} H^{(k)} Z^{(k)}) \right) \left\| R^{(k)^T} \right\| \tag{20}$$

This shows three scenarios for the KKT system, going from bad to worse to worst [22]:

(1) $R^{(k)^T}$ is well conditioned and $Z^{(k)^T} H^{(k)} Z^{(k)}$ is ill conditioned, which is the best case.

(2) $R^{(k)^T}$ is ill conditioned but $Z^{(k)^T} H^{(k)} Z^{(k)}$ is well conditioned.

(3) both $R^{(k)^T}$ and $Z^{(k)^T} H^{(k)} Z^{(k)}$ are ill-conditioned which gives the worst case.

The quantity $\chi(Z^{(k)^T} H^{(k)} Z^{(k)})$ defines the condition number of the projected Hessian matrix, and if M_{KKT} is the matrix in the left-hand side of equation (17) then $\chi(M_{KKT})$ defines the condition number for the whole Lagrangian system.

3. DESCRIPTION OF THE DIRECT TRANSCRPTION METHOD

In this section, we present the discretization method of the optimal control problem, which results in an nonlinear programming problem.

3.1 Discretization of the Optimal Control Problem

In this section, we describe the evaluation of the objective function, discretization of the state equations using a general Runge-Kutta method and the evaluations of the equality and inequality constraints. The whole technique is referred to as the control parameterization method. This method is used in [20, 21, 22].

We will assume that $t_f - t_0 = K\tau$ for some positive integer K. Then,

$$[t_0, t_f] = \bigcup_{k=0}^{K-2} [t_0 + k\tau, t_0 + (k+1)\tau) = [t_0, t_0 + \tau) \cup [t_0 + \tau, t_f]$$

For the discretization of the control variables, each delay interval $[t_0 + k\tau, t_0 + (k+1)\tau)$ is divided into S equally spaced switching points $\{s_j : j = 0,1,\ldots,S\}$, the distance between any two successive switching points s_j and s_{j+1} is $h_c = s_{j+1} - s_j = \tau / S$. Then, $N_c = S \cdot K$ is the number of all the switching points in the closed interval $[t_0, t_f]$ and $s_j = t_0 + j \cdot h_c,\ j = 0,1,\ldots,N_c$.

For $t \in [s_i, s_{i+1})$ the control $u_j, j = 1, \ldots m$ is approximated by a constant value $v_j^i, i = 0, \ldots, N_c; j = 1, \ldots, m$, with $v_j^{N_c} = v_j^{N_c - 1}$ (no control jump at the final time).

Now, the control $u_j(t), t \in [t_0, t_f]$ is given by

$$u_j(t) = \bigcup_{i=0}^{N_c} v_j^i$$

For $t \in [s_i, s_{i+1})$, we see that $s_i - \tau \le t - \tau < s_{i+1} - \tau$, or equivalently, $s_i - S \cdot h_c \le t - S \cdot h_c < s_{i+1} - S \cdot h_c$. This is equivalent to

$$s_{i-S} \le t - \tau < s_{i+1-S}$$

Now,

$\vec{u}(t - \tau) = \vec{u}(s_{i-S}) = \vec{v}^{i-S}$ for $t \in [s_i, s_{i+1})$.

Finally, the delayed control variables $\vec{u}(t - \tau)$ are evaluated at the time $t \in [s_i, s_{i+1})$ by the rule:

$$\vec{u}(t - \tau) = \begin{cases} \vec{v}^{i-S}, & t \ge \tau \\ \vec{\vartheta}(t - \tau) & t < \tau \end{cases} \tag{21}$$

The optimal control problem (1)-(6) takes the form

$$\min_{\vec{v} \in U} \phi(\vec{x}(t_f)) + \sum_{i=0}^{S} L_0(t, \vec{x}(t), \vec{x}(t - \tau), \vec{\vartheta}(t), \vec{\vartheta}(t - \tau)) + \sum_{i=S+1}^{N_c - 1} L_0(t, \vec{x}(t), \vec{x}(t - \tau), \vec{v}^i, \vec{v}^{i-S})$$

subject to the dynamics

$$\dot{\vec{x}}(t) = \begin{cases} \vec{f}(t, \vec{x}(t), \vec{x}(t - \tau), \vec{v}^i, \vec{\vartheta}(t - \tau)), & s_i < t_0 + \tau \\ \vec{f}(t, \vec{x}(t), \vec{x}(t - \tau), \vec{v}^i, \vec{v}^{i-S}), & s_i \ge t_0 + \tau \end{cases}$$

Subject to the initial condition and the preconditions

$$\vec{x}(t_0) = \vec{x}^0,$$
$$\vec{x}(t) = \vec{\varphi}(t), t \in [t_0 - \tau, t_0],$$
$$\vec{u}(t) = \vec{\vartheta}(t), t \in [t_0 - \tau, t_0],$$

subject to the equality constraints

$$\begin{cases} \vec{E}(t, \vec{x}(t), \vec{x}(t - \tau), \vec{v}^i, \vec{\vartheta}(t - \tau)) = 0, & s_i < t_0 + \tau \\ \vec{E}(t, \vec{x}(t), \vec{x}(t - \tau), \vec{v}^i, \vec{v}^{i-S}) = 0, & s_i \ge t_0 + \tau \end{cases}$$

subject to the continuous state inequality constraints

$$\begin{cases} \vec{I}(t, \vec{x}(t), \vec{x}(t - \tau), \vec{v}^i, \vec{\vartheta}(t - \tau)) \le 0, & s_i < t_0 + \tau \\ \vec{I}(t, \vec{x}(t), \vec{x}(t - \tau), \vec{v}^i, \vec{v}^{i-S}) \le 0, & s_i \ge t_0 + \tau \end{cases}$$

and subject to terminal conditions:

$$\vec{\psi}(t_f, \vec{x}(t_f)) = \vec{\psi}_f$$

For the discretization of the state variables, each switching interval $[s_i, s_{i+1})$ is divided into Q quadrature sub-intervals, where Q is a fixed positive integer. The time interval $[t_0, t_f]$ is divided into $N = N_c \cdot Q$ subintervals. Let

$h = (t_f - t_0)/N$, then the time t_i is given by $t_i = t_0 + i \cdot h$. Now at time t_i the state variable x_j is approximated by the value x_j^i; $j = 1,...,n, i = 0,...,N$. It is clear that $h = (t_f - t_0)/N = (t_f - t_0)/(N_c Q) = h_c / Q.$

For the discretization of the delayed optimal control problem, we have to find approximations to the delayed state variables $\vec{x}(t-\tau)$ at the mesh points $\{t_i; i = 0,\ldots,N\}$. Let x_j^i denote the approximation of $x_j(t_i)$ for all $j = 1,\ldots n$ and $i = 1,\ldots N$. If $t_i < \tau$, then $\vec{x}(t_i - \tau) = \vec{\varphi}(t_i - \tau)$. But, if $t_i > \tau$, one can see that

$$\tau = S \cdot h_c = S \cdot Q \cdot h = D \cdot h,$$

where $D = S \cdot Q$; and therefore, $\vec{x}(t_i - \tau) = \vec{x}(t_i - D \cdot h) = \vec{x}(t_{i-D}) = \vec{x}^{i-D}$.

The delayed state variables $\vec{x}(t_i - \tau)$ is finally approximated as:

$$\vec{x}(t_i - \tau) = \begin{cases} \vec{\varphi}(t_i - \tau), & t_i < \tau \\ \vec{x}(t_{i-D}) & t_i \geq \tau \end{cases}$$

And set

$$\vec{\Phi} = [\vec{x}(t_i - \tau), i = 0,\ldots,N]^T$$

To evaluate the control variable $\vec{u}(t)$ and the delayed control variable $\vec{u}(t-\tau)$ at $\{t = t_i; i = 0,\ldots,N\}$, It is clear that $s_k \leq t_i < s_{k+1}$ for some positive integer k. The index k is given by the relation $k = \lfloor t_i / h_c \rfloor = \lfloor i/Q \rfloor$, then it follows that, $\vec{u}(t_i) = \vec{v}^k$.

Then from equation (21) we have,

$$\vec{u}(t_i - \tau) = \begin{cases} \vec{\vartheta}(t_i - \tau), & t_i \in [t_0, t_0 + \tau) \\ \vec{v}^{k-S}, & t_i \geq t_0 + \tau \end{cases}$$

And set

$$\vec{\Psi} = \{\vec{u}(t_i - \tau), i = 0,\ldots,N\}$$

3.2 Transcription of the OCP into A NLP

Let $L_0^j, j = 0,\ldots,N$ be the value of the integrand part of the objective function at time $t_j, j = 0,\ldots,N$. The objective function is approximated by the Simpson rule as:

$$\min_{\vec{v} \in U} \phi(\vec{x}(t_N)) + \frac{h}{3}\left(L_0^0 + L_0^N\right) + \frac{4h}{3}\sum_{i=1}^{\left\lfloor \frac{N}{2} \right\rfloor} L_0^{2i-1} + \frac{2h}{3}\sum_{i=1}^{\left\lfloor \frac{N}{2} \right\rfloor} L_0^{2i} \tag{22}$$

For the discretization of the state equations, we use an M-stage Runge-Kutta method. On each subinterval $[t_i, t_{i+1}], i = 0,\ldots,N-1$, let $t_i^j = t_i + c_j h, j = 1,\ldots,M$, where $0 = c_0 \leq c_1 \leq \cdots \leq c_M = 1$. Then,

$$\eta^{i+c_j} = \vec{u}(t_i^j - \tau) = \begin{cases} \vec{\vartheta}(t_i^j - \tau), & t_i^j \in [t_0, t_0 + \tau) \\ \vec{v}^{k-S}, & t_i^j \geq t_0 + \tau \end{cases} \tag{23}$$

And

$$\vec{\sigma}^{i+c_j} = \vec{x}(t_i^j - \tau) = \begin{cases} \varphi(t_i^j - \tau), & t_i^j < \tau \\ x(t_{i-D}^j) & t_i^j \geq \tau \end{cases} \tag{24}$$

The state equations become:

$$\vec{x}(t_i^M) = \vec{x}(t_i^1) + h\sum_{j=1}^{M} b_j \vec{f}(t_i^j, x(t_i^1) + h\sum_{l=1}^{M} a_{jl} \vec{f}(t_i^l, \vec{x}(t_i^l), \vec{\sigma}^{i+c_l}, \vec{u}(t_i^l), \vec{\eta}^{i+c_l}), \sigma^{i+c_j}, \vec{u}(t_i^j), \eta^{i+c_j}) \quad (25)$$

The above equation can be written as:

$$\vec{x}(t_i^M) - \left(\vec{x}(t_i^1) + h\sum_{j=1}^{M} b_j \vec{f}(t_i^j, x(t_i^1) + h\sum_{l=1}^{M} a_{jl} \vec{f}(t_i^l, \vec{x}(t_i^l), \vec{\sigma}^{i+c_l}, \vec{u}(t_i^l), \vec{\eta}^{i+c_l}), \sigma^{i+c_j}, \vec{u}(t_i^j), \eta^{i+c_j}) \right) = 0 \quad (26)$$

The initial data and conditions can be written as:

$$\left.\begin{array}{l} \vec{x}(t_0) - \vec{x}^0 = 0, \\ \vec{x}(t_i - \tau) - \vec{\varphi}(t_i - \tau) = 0, t \in [t_0, t_0 + \tau], \\ \vec{u}(t_i - \tau) - \vec{\vartheta}(t_i - \tau) = 0, t \in [t_0, t_0 + \tau] \end{array}\right\}, \quad (27)$$

The equality constraints are given by:

$$\begin{cases} \vec{E}(t, \vec{x}(t), \vec{x}(t-\tau), \vec{v}^i, \vec{\vartheta}(t-\tau)) = 0, & s_i < t_0 + \tau \\ \vec{E}(t, \vec{x}(t), \vec{x}(t-\tau), \vec{v}^i, \vec{v}^{i-S}) = 0, & s_i \geq t_0 + \tau \end{cases} \quad (28)$$

And the inequality constraints are given by:

$$\begin{cases} \vec{I}(t, \vec{x}(t), \vec{x}(t-\tau), \vec{v}^i, \vec{\vartheta}(t-\tau)) \leq 0, & s_i < t_0 + \tau \\ \vec{I}(t, \vec{x}(t), \vec{x}(t-\tau), \vec{v}^i, \vec{v}^{i-S}) \leq 0, & s_i \geq t_0 + \tau \end{cases} \quad (29)$$

Finally both the control variables $\vec{u} = [u_1, \ldots, u_m]^T$ and the state variables $\vec{x} = [x_1, \ldots, x_n]^T$ are placed in one vector $\vec{Y} \in R^{n(1+N)+m(1+N_c)}$. The jth component of the ith state variable x_i^j is mapped to $\vec{Y}(i + (i-1)\cdot N + j); i = 1, \ldots, n, j = 0, \ldots, N$ and u_k^l is mapped into $\vec{Y}(k + n + (n-1)\cdot N + (k-1)N_c + \lfloor (l-1)/Q \rfloor); k = 1, \ldots, m, l = 0, \ldots, N$.

The total length of the vector $\vec{Y}$ is $n\cdot(1+N) + m\cdot(1+N_c)$.

The problem described by (22), with the constraints (23)-(29) then becomes:

$$\min_{\vec{Y} \in R^{n(1+N)+m(1+N_c)}} f_0(\vec{Y}, \vec{\Phi}, \vec{\Psi}) \quad (30)$$

subject to:

$$h(\vec{Y}, \vec{\Phi}, \vec{\Psi}) = 0 \quad (31)$$

And

$$g(\vec{Y}, \vec{\Phi}, \vec{\Psi}) \leq 0 \quad (32)$$

subject to the initial conditions:

$$\vec{Y}(i + (i-1)N) - \vec{x}^{0^i} = 0, i = 1, \ldots, N \quad (33)$$

Subject to equality constraints

$$E(\vec{Y}, \vec{\Phi}, \vec{\Psi}) = 0 \quad (34)$$

and inequality constraints:

$$I(\vec{Y}, \vec{\Phi}, \vec{\Psi}) \leq 0 \quad (35)$$

The NLP given by (30)-(35) can be solved using the *fmincon* matlab's function.

4. COMPUTING THE CONDITION NUMBERS ASSOCIATED WITH THE COMPUTATIONS OF OPTIMAL CONTROL

In this section, we compute the condition numbers that are associated with the use of Matlab's fmincon function to compute the optimal controls. The condition numbers associated with the three medium scale optimization algorithms (active-set, trust-region-reflective and the SQP) will be used.

The Hessian matrix $H^{(k)}$, which is given by the optimization routine fmincon, is not exact. It is obtained by updating the Hessian matrix in every iteration, and is not computed directly from the solution vector $\vec{Y}$. Hence, the projected Hessian matrix $H_{22}^{(k)} = Z^{(k)^T} H^{(k)} Z^{(k)}$ given by the optimization toolbox does not has the exact condition number as same as the true projected Hessian. If $\vec{z}$ is the computed solution vector at iteration k, the Hessian matrix is to be recomputed at $\vec{z}$. Let I_n I be the $n \times n$ identity matrix, and let I_n^j denotes the j^{th} column of the identity matrix I_n. Let $\varepsilon > 0$, be a small real number, then using the central differences, the Hessian matrix $H^{(k)}$ is given by:

$$(H^{(k)})_{ii} = \frac{\vec{f}_0(\vec{z} + \varepsilon \cdot I_n^i) - 2\vec{f}_0(\vec{z}) + \vec{f}_0(\vec{z} - \varepsilon \cdot I_n^i)}{4\varepsilon^2}$$

$$(H^{(k)})_{ij} = \frac{\vec{f}_0(\vec{z} + \varepsilon \cdot I_n^i + \varepsilon \cdot I_n^j) - \vec{f}_0(\vec{z} - \varepsilon \cdot I_n^i + \varepsilon \cdot I_n^j) + \vec{f}_0(\vec{z} + \varepsilon \cdot I_n^i - \varepsilon \cdot I_n^j) + \vec{f}_0(\vec{z} - \varepsilon \cdot I_n^i - \varepsilon \cdot I_n^j)}{4\varepsilon^2}, j \neq i$$

Given a tolerance *ConTol* in the constraints violation, it is possible to compute the active constraints. If given a set $\vec{h}(\vec{x}) \leq 0$, of inequality constraints, then a constraint $h_i(\vec{x})$ is active at $\vec{z}$ if $\left|\vec{h}(\vec{z})\right| \leq ConTol$.

Let $\vec{A}$ be the set of active constraints at $\vec{z}$. That is $\vec{A} = \{h_i(\vec{z}) : |h_i(\vec{z})| < ConTol\}$.

Let $\vec{g} = \begin{bmatrix} \vec{A}(\vec{z}) \\ \vec{c}(\vec{z}) \end{bmatrix}$. Then the i^{th} column of the Jacobian matrix $\nabla \vec{c}^T$ at $\vec{z}$ is given by:

$$(\nabla \vec{c}^T)_i = \frac{g(\vec{z} + \varepsilon \cdot I_n^i) - g(\vec{z} - \varepsilon \cdot I_n^i)}{2 \cdot \varepsilon}$$

5. AN ILLUSTRATIVE EXAMPLE

$$\min_{u(t)} \quad J(u) = \frac{x^2}{2} + \frac{1}{2}\int_0^2 (x^2(t) + u^2(t))dt$$

subject to:

$$\dot{x}(t) = x(t)\sin(x(t)) + x(t-1) + u(t), t \in [0,2]$$
$$x(t) = 10, t \in [-1,0]$$

and subject to the continuous inequality constraint

$$x(t) + 8t - 114.6 \leq 0, t \in [0,2]$$

and subject to the terminal equality constraint

$$x^2(2) - 22x(2) - 119.854 = 0$$

We used the classical fourth-order Runge-Kutta method for the discretization of the state equation. For 1, 2, 4, 8, 16, 32, 64 and 128 switching points per a delay interval, with 1, 2, 4 and 8 quadrature points per a switching interval, the active set method is used to compute the optimal control. The Hessian matrix and gradient vector obtained by the Matlab are used to compute the condition numbers associated with the computation of the optimal control. Then, we recomputed the Hessian matrix and the gradient vector by the laws from the past section. We obtained the following condition numbers:

ISSN 2347-1921

Table 1. The condition numbers (approximated for 3 decimal places) that are obtained by both our computations and the Matlab's computations with 1, 2, 4, 8, 16, 32, 64 and 128 switching points per delay interval (SPs Per DI) and 1, 2, 4 and 8 quadrature points per switching interval (QPs Per SI)

SPs Per DI	QPs Per SI	Condition numbers given by our computations				Condition numbers given by Matlab			
		Active Constraints	Projected Hessian	System Parameters	Lagrangean System	Active Constraints	Projected Hessian	System Parameters	Lagrangean System
1	1	5.484E+00	1.800E+00	9.871E+00	2.416E+01	5.484E+00	5.437E+00	2.982E+01	9.091E+01
1	2	1.592E+01	1.651E+00	2.629E+01	1.150E+02	1.592E+01	2.456E+00	3.909E+01	7.492E+01
1	4	3.023E+01	6.993E+00	2.114E+02	6.850E+02	3.023E+01	1.153E+00	3.486E+01	1.065E+02
1	8	5.916E+01	1.904E+01	1.127E+03	4.911E+03	5.916E+01	1.543E+00	9.127E+01	1.561E+02
2	1	1.260E+01	1.817E+00	2.289E+01	1.326E+02	1.260E+01	4.093E+00	5.156E+01	4.010E+02
2	2	4.113E+01	3.519E+00	1.447E+02	1.446E+03	4.113E+01	1.842E+00	7.578E+01	2.054E+02
2	4	6.390E+01	1.323E+01	8.452E+02	6.027E+03	6.390E+01	2.917E+00	1.864E+02	2.710E+02
2	8	1.199E+02	3.508E+01	4.206E+03	4.023E+04	1.199E+02	1.403E+00	1.682E+02	1.159E+05
4	1	1.622E+01	1.947E+00	3.158E+01	1.502E+03	1.622E+01	1.849E+01	2.999E+02	2.534E+02
4	2	3.754E+01	8.045E+00	3.020E+02	2.014E+03	3.754E+01	1.319E+01	4.950E+02	6.099E+02
4	4	3.338E+01	2.684E+01	8.960E+02	3.153E+03	3.338E+01	8.351E+02	2.787E+04	8.649E+04
4	8	3.672E+01	6.830E+01	2.508E+03	7.547E+03	3.672E+01	1.545E+02	5.672E+03	2.520E+03
8	1	4.172E+01	1.805E+00	7.529E+01	1.787E+04	4.172E+01	2.549E+01	1.063E+03	1.089E+03
8	2	8.874E+01	1.631E+01	1.447E+03	2.186E+04	8.874E+01	5.695E+01	5.054E+03	9.901E+03
8	4	7.132E+01	5.287E+01	3.771E+03	2.811E+04	7.132E+01	5.364E+02	3.826E+04	1.908E+05
8	8	7.582E+01	1.331E+02	1.009E+04	6.325E+04	7.582E+01	4.137E+02	3.137E+04	2.323E+04
16	1	9.186E+01	1.803E+00	1.656E+02	2.583E+05	9.186E+01	4.671E+01	4.291E+03	6.857E+03
16	2	1.862E+02	3.305E+01	6.153E+03	1.905E+05	1.862E+02	6.943E+01	1.293E+04	3.877E+04
16	4	1.451E+02	1.062E+02	1.541E+04	2.310E+05	1.451E+02	1.079E+02	1.565E+04	1.449E+04
16	8	1.529E+02	2.669E+02	4.080E+04	5.117E+05	1.529E+02	7.976E+01	1.219E+04	7.174E+04
32	1	1.882E+02	1.866E+00	3.512E+02	4.097E+06	1.882E+02	1.178E+02	2.216E+04	4.409E+04
32	2	3.813E+02	6.628E+01	2.528E+04	1.594E+06	3.813E+02	1.644E+03	6.269E+05	5.118E+07
32	4	2.917E+02	2.114E+02	6.167E+04	1.862E+06	2.917E+02	2.507E+02	7.315E+04	3.754E+05
32	8	3.065E+02	5.299E+02	1.624E+05	4.107E+06	3.065E+02	1.207E+02	3.699E+04	1.223E+06
64	1	3.789E+02	1.951E+00	7.393E+02	6.708E+07	3.789E+02	1.523E+03	5.771E+05	3.185E+06
64	2	7.627E+02	1.332E+02	1.016E+05	1.272E+07	7.627E+02	1.159E+02	8.837E+04	1.781E+06
64	4	5.842E+02	4.231E+02	2.472E+05	1.492E+07	5.842E+02	5.619E+02	3.283E+05	5.912E+06
64	8	6.134E+02	1.060E+03	6.504E+05	3.287E+07	6.134E+02	8.825E+02	5.413E+05	3.027E+06
128	1	7.587E+02	2.207E+00	1.674E+03	1.200E+09	7.587E+02	7.628E+03	5.787E+06	9.207E+08
128	2	1.525E+03	2.665E+02	4.065E+05	1.017E+08	1.525E+03	4.440E+02	6.773E+05	3.556E+06
128	4	1.169E+03	8.478E+02	9.909E+05	1.193E+08	1.169E+03	5.996E+03	7.008E+06	9.817E+07
128	8	1.227E+03	2.125E+03	2.608E+06	2.630E+08	1.227E+03	5.396E+03	6.621E+06	1.043E+08

We also plotted the condition numbers of the Lagrangean System by fixing the number of switching points per delay interval one time, and by fixing the number of quadrature points per switching interval another time; for both our computations and the Matlab computations.

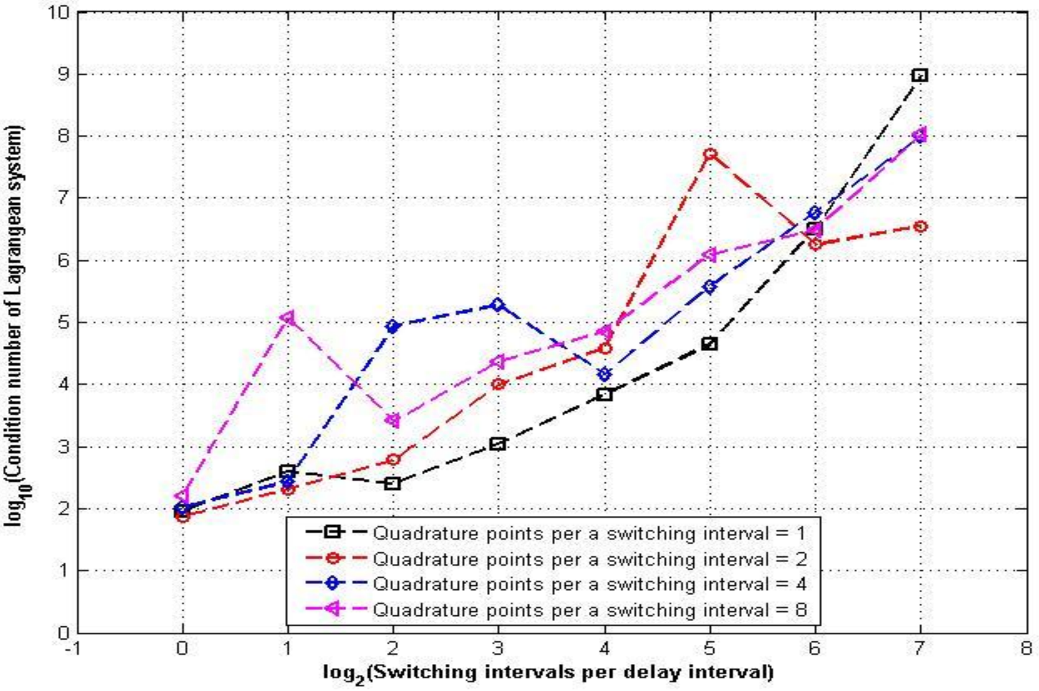

Figure 1. Condition numbers of the Lagrangean system, obtained by fixing the number of switching intervals per delay interval from the Hessian matrix and gradient vector obtained from Matlab computations.

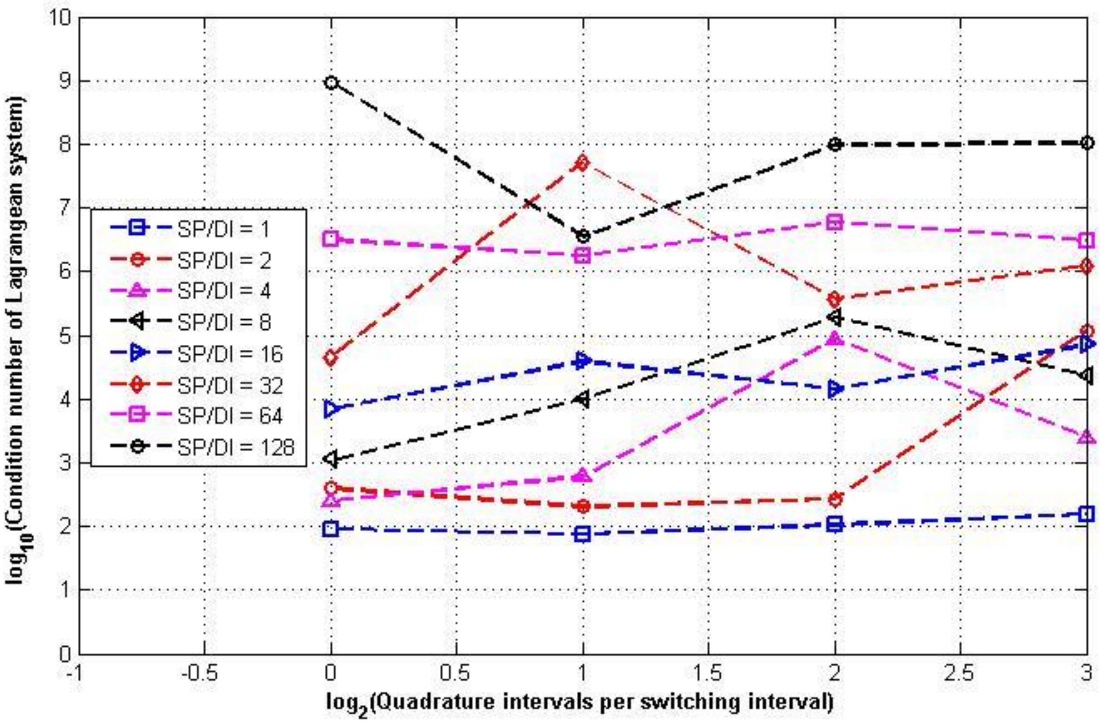

Figure 2. Condition numbers of the Lagrangean system, obtained by fixing the number of quadrature points per a switching interval from the Hessian matrix and gradient vector obtained from Matlab computations.

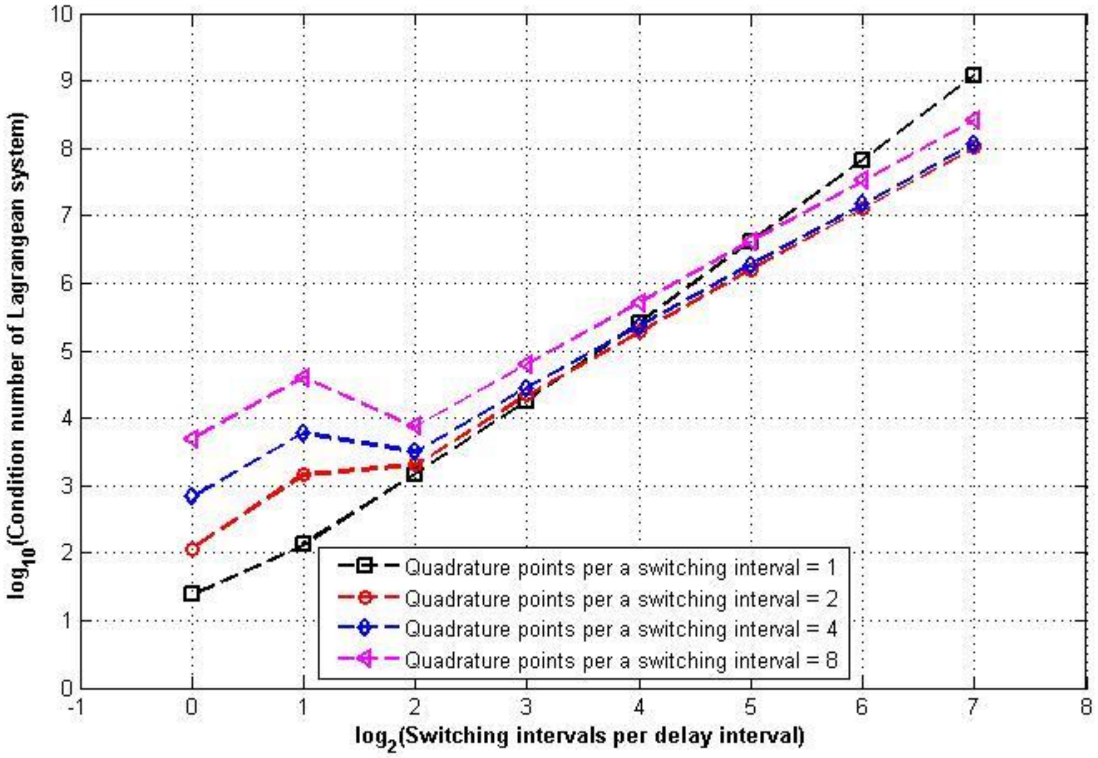

Figure 3. Condition numbers of the Lagrangean system, obtained by fixing the number of switching intervals per delay interval from the Hessian matrix and gradient vector obtained by our computations.

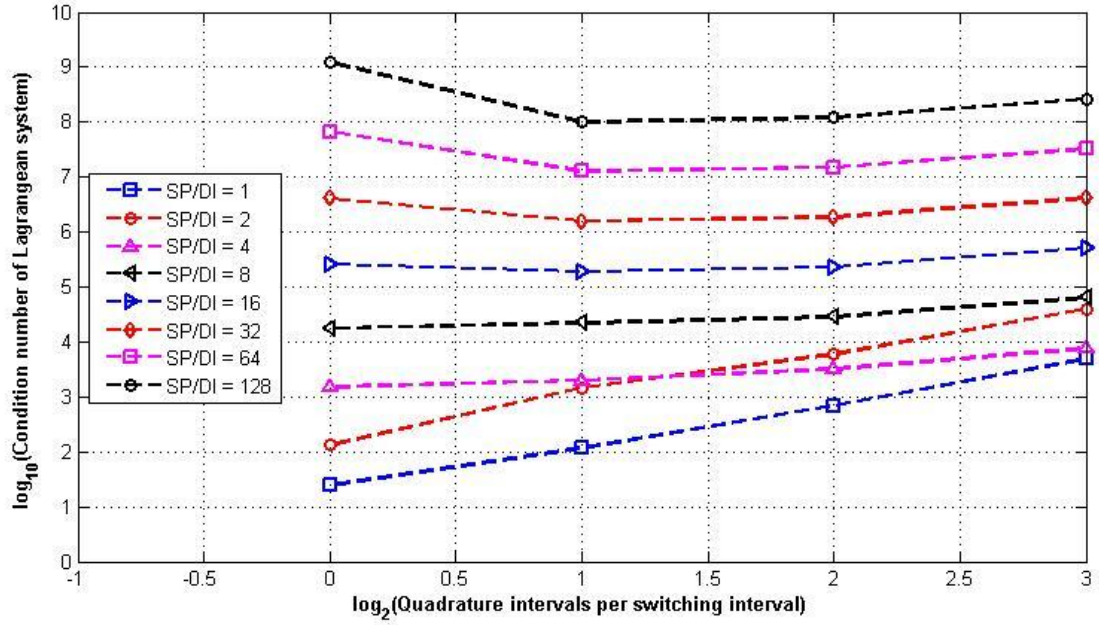

Figure 4. Condition numbers of the Lagrangean system, obtained by fixing the number of quadrature points per a switching interval from the Hessian matrix and gradient vector obtained by our computations.

ISSN 2347-1921

6. CONCLUSION

The main purpose from this paper was to measure the condition numbers associated to the solution of a constrained optimization problem that is resulted from the discretization of an optimal control problem using a Runge-Kutta method. Because the Hessian matrices obtained from the Matlab's optimization toolbox are not accurate, we re-evaluated the Hessian matrices directly from the optimal solution of the optimization problem. These computations have been made for different numbers of switching points and quadrature points per a switching interval.

From figures 1 and 2, we see that the condition numbers resulting from the Hessian matrix and gradient vector obtained by the Matlab's *fmincon* do not show how the selection of the number of switching point per delay interval (quadrature points per switching interval) can affect those condition numbers as seen in Table 1 and figures 1 and 2.

We found that, as the number of switching intervals per a delay interval increases, the condition numbers of the active constraints, projected Hessian and the whole Lagrangian system increase as seen in Table 1 and Figure 3. Also, as the number of quadrature intervals per a switching interval increases, the condition numbers of the active constraints, projected Hessian and the whole Lagrangian system increase as seen in Table 1 and Figure 4.

Figure 3 shows that when fixing the number of quadrature points per a switching interval to one, then the condition numbers of the Lagrangian system jump high as the number of the switching points per delay interval increases. Therefore, choosing 2, 4 and 8 quadrature points per a switching interval might be much stable. On the other hand, Figure 4 shows that when fixing the number of quadrature points per a switching interval to either one or two, the condition numbers of the Lagrangian system jump high as the number of quadrature points per a switching interval increases. Therefore, choosing 4, 8, 16 or 32 switching points per a delay interval might be much stable.

Finally, the three medium scale Matlab's optimization algorithms give almost similar results when are used to compute the condition numbers associated with the optimal control computations.

REFERENCES

[1] Gölmann, L., Kern, D., and Maurer, H. (2009). Optimal control problems with delays in state and control variables subject to mixed control-state constraints. Optimal Control Applications and Methods, 30(4):341–365.

[2] Elhia, M., Rachik, M., and Benlahmar, E. (2013). Optimal control of an SIR model with delay in state and control variables. ISRN Biomathematics, 2013:1–7.

[3] Hattaf, K. and Yousfi, N. (2012). Optimal control of a delayed hiv infection model with immune response using an effficient numerical method. ISRN Biomathematics, 2012:1–7.

[4] Lin, J. (1996). Optimal control of time-delay systems by forward iterative dynamic programming. Ind. Eng. Chem. Res., 35(8):2795–2800.

[5] Luus, R. (2000). Iterative Dynamic Programming. Charman and Hall/CRC.

[6] Williams, B. (2007). Optimal management of non-Markovian biological population. Ecological Modelling, 200:234–242.

[7] Betts, J.T. (2001). Practical Methods for Optimal Control Using Nonlinear Programming. Society for Industrial and Applied Mathematics.

[8] Shwartz, A. (1996). Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems. Phd thesis, Electronic Research Laboratory, UC Burkeley.

[9] Inanc, T. and Bhattacharya, R. (2003). Numerical solution of optimal control problems using splines.

[10] Von Stryk, O. (1993). Numerical solution of optimal control problems by direct collocation. International Series of Numerical Mathematics, 111:129–143.

[11] Wong, K. H. (1988). A control parametrization algorithm for nonlinear time-lag optimal control problems. Opsearch, 25(E):177–184.

[12] Wong, K. H., Jennings, L., and Benyah, F. (2001). Control parametrization method for free planning time optimal control problems with time-delayed arguments. Journal of Nonlinear Analysis, 47:5679–5689.

[13] Wong, K. H., Jennings, L., and Benyah, F. (2002). The control parametrization enhancing transform for constrained time-delayed optimal control problems. ANZIAM J. The Australian and New Zealand Industrial and Applied Mathematics Journal, 43(E):E154–E185.

[14] Betts, J. T. and Hoffman, W. (1999). Exploring sparsity in the direct transcription method for optimal control. Computational Optimization and Applications, 14:179–201.

[15] Coleman, T., Branch, M., and Grace, A. (1999). Optimization Toolbox for Use With Matlab. The Mathworks Inc.

[16] Boggs, P. T. and Tolle, J. W. (1996). Sequential quadratic programming. Acta Numerica, pages 1–48.

[17] Lawrence, C. T. and Tits, A. L. (2001). A computationally efficient feasible sequential quadratic programming algorithm. SIAM Journal on Optimization, 11:1092–1118.

[18] Benyah, F. and Jenning, L. (2000). A comparison of the ill-conditioning of two optimal control computation methods. Technical report, the University of the Western Australia, Dept. of Mathematics and Statistics, Nedlands 6907 WA.

[19] Benyah, F. and Jennings, L. S. (1998). Ill-conditioning in optimal control computation. In Proceedings of the Eighth International Colloquium on Differential Equations (ed. D. Bainov), pages 81–88, VSP BV, Utrecht, The Netherlands.

[20] Benyah, F. and Jennings, L. S. (2001). A review of ill-conditioning and regularization in optimal control computation. In X., E., Yang, Teo, K. L., and Caccetta, L., editors, Optimization Methods and Applications, pages 23–44. Kluwer Academic Publishers, Dordrecht, The Netherlands.

[21] Matveev, A. (2005). The instability of optimal control problems to time delay. SIAM J. Control Optimization, 43(5):1757–1786.

[22] Jennings, L. S., Teo, K. L., Goh, C. J., and Fisher, M. E. (1990). Miser3: A Fortran program for solving optimal control problems. A Manual Book.

Author' biography with Photo

Dr. Eihab Bashier obtained his PhD in 2009 from the University of the Western Cape in South Africa. He is currently working as an associate professor, faculty of Mathematical Sciences, University of Khartoum, Sudan.

The research interests of Dr. Bashier are mainly in developing numerical methods for differential equations with applications to mathematical biology problems. He has also research interests in information and computer security. He has supervised many MSc and PhD research projects at the University of Khartoum, most of them are in developing classes of high-order numerical methods for solving PDEs with time delays.

Dr. Bashier published many research papers in international journals. In 2011, Dr. Bashier won the African Union and the Third World Academy of Science (AU-TWAS) young scientists' national award in basic sciences, technology and Innovation.