Matematika, 2003, Jilid 19, Bil. 1, hlm. 1–13

The Newton-Like Properties of the Updating Mechanism of a Model-Reality Differences Algorithm

Rohanin Ahmad & Mohd Ismail Abd Aziz
Department of Mathematics
Universiti Teknologi Malaysia, Skudai
81310 Johor Bahru, Malaysia.

Abstract The Dynamic Integrated Systems Optimization and Parameter Estimation (DISOPE) algorithm is an algorithm for solving nonlinear optimal control problems and is of the gradient descent type. The updating step of DISOPE plays an important role in terminating the iterations of the algorithm and hence in determining its rate of convergence. In this paper, the mechanism was shown to have Newton-like properties and the order convergence established.

Keywords Composite maps, Newton's method, Error function, Initial solutions, Order of convergence.

Abstrak *Algoritma Dynamic Integrated Systems Optimization and Parameter Estimation (DISOPE) ialah suatu algoritma untuk menyelesaikan masalah kawalan optimum tak linear dan algoritma ini tergolong ke jenis penurunan gradien. Langkah pengemaskinian DISOPE memainkan peranan penting dalam pemberhentian lelaran algoritma ini dan seterusnya dalam penentuan kadar penumpuannya. Dalam kertas kerja ini, mekanisma tersebut ditunjukkan sebagai mempunyai ciri-ciri mirip kaedah Newton dan peringkat penumpuannya ditentukan.*

Katakunci *Peta gubahan, Kaedah Newton, Fungsi ralat, Penyelesaian awal, Peringkat penumpuan.*

1 Introduction

The Dynamic Integrated Systems Optimization And Parameter Estimation (DISOPE) algorithm is an iterative procedure for solving dynamic nonlinear optimal control problems. It was first developed by Roberts [10] and further improved by Becerra [3]. The algorithm specifically takes into account the model-reality differences in structure and parameters of the problem to be solved. Repeated solutions of optimization and estimation of parameters within the model is used for calculating the optimum [10]. The updating mechanism of this algorithm plays an important role in determining the convergence rate of the algorithm. In an earlier paper [1], the updating mechanism has been analyzed separately from the main algorithm by partitioning DISOPE into two distinct sub procedures based on theorems of composite mappings. One of the map is the main DISOPE algorithm and the other is the updating mechanism. The updating mechanism was treated as a full-fledged algorithm with an appropriate error function determined for it. The convergence of the updating mechanism was then established by comparing it with Newton's Method. The Newton's Method is one of the oldest methods for solving many root finding and optimization problems. In its simplest form, it converges only if the initial guess is sufficiently close to a solution [9]. Provided the initial guess is good enough, Newton's iteration will converge at least quadratically to the required solution [5,7]. If the initial estimate of the solution is very inaccurate, Newton's method converges with a very slow rate over the first few iterations [5] or it may even diverge [7]. This paper further investigates the convergence of DISOPE and its order of convergence. Its behavior with different set of initial solutions is also investigated.

2 The DISOPE approach

Consider the following unconstrained real optimal control problem (ROP), with given initial conditions:

$$\min_{u(t)} J^{*} = \varphi(x(t_f)) + \int_{t_0}^{t_f} L^*(x(t), u(t), t)dt,$$

subject to

$$\begin{aligned} \dot{x} &= f^*(x(t), u(t), t), \qquad (1)\\ x(t_0) &= x_0,\\ x_i(t_f) &= x_{if}; i \in [1, q], q < n, \end{aligned}$$

defined over $t \in [t_0, t_f]$, where $u(t) \in R^m$ and $x(t) \in R^n$ are the continuous control and state vectors, respectively. $L^* : R^n \times R^m \to R$ is the real performance measure function and $f^* : R^n \times R^m \to R^n$ represents the real plant dynamics.

DISOPE does not work directly on ROP but rather modeled the reality into a Model Based Problem (MOP) which includes parameter estimates $\gamma(t) \in R$ and $\alpha(t) \in R^r$. These two estimates take into account the value differences between reality and model. This model is then expanded into another optimal control problem, which is equivalent to ROP called Expanded Optimal Control Problem (EOP). EOP ties ROP and MOP together by including the following equality expressions of states functions and performance index from ROP and MOP as constraints.

$$\begin{aligned} f(z(t), v(t), \alpha(t)) &= f^*(z(t), v(t), t),\\ L(z(t), v(t), \gamma(t)) &= L^*(z(t), v(t), t), \qquad (2)\\ u(t) &= v(t),\\ x(t) &= z(t), \end{aligned}$$

where $L^* : R^n \times R^m \rightarrow R$ and $f^* : R^n \times R^m \rightarrow R^n$ are, respectively, the performance index and the plant dynamics of the model. Both are approximates of the real performance index L^* and the plant dynamics f^*. $v(t) \in R^m$ and $z(t) \in R^n$ are introduced to distinguish between the state and control variables used in the optimization step and those used in the parameter estimation step. The convexification terms $\parallel u(t) - v(t) \parallel$ and $\parallel x(t) - z(t) \parallel$ are introduced in the performance index to aid convergence. r_1 and r_2 are given scalar convexification factors, with the following function H as the Hamiltonian:

$$H = L(x, u, \gamma) + p^T f(x, u, \alpha) - \lambda^T u - \beta^T x + \frac{1}{2} r_1 \parallel u(t) - x(t) \parallel^2 + \frac{1}{2} r_2 \parallel u(t) - x(t) \parallel^2 .$$

Instead of solving the EOP, the following modified model based optimal control problem (MMOP) is solved. The MMOP is:

$$\begin{aligned} \min_{u(t)_e} J &= \varphi(x(t_f)) + \int_{t_0}^{t_f} L^*(x(t), u(t), \gamma(t)) - \lambda(t)^T - \beta(t)^T x(t) \\ &\quad + \frac{1}{2} r_1 \parallel u(t) - x(t) \parallel^2 + \frac{1}{2} r_2 \parallel u(t) - x(t) \parallel^2 \, dt, \end{aligned} \tag{3}$$

subject to

$$\begin{aligned} \dot{x} &= f^*(x(t), u(t), \alpha(t)), \\ x(t_0) &= x_0, \\ x_i(t_f) &= x_{if}; i \in [1, q], q < n, \end{aligned}$$

together with its optimality conditions

$$\begin{aligned} \nabla_u H &= 0, \\ \nabla_x H + p(t) &= 0, \\ \nabla_p H &= 0, \end{aligned} \tag{4}$$

For details please refer Roberts [10], Becerra [3], and Ahmad and Mohd Ismail [1].

3 The standard DISOPE Algorithm

For clarification the following algorithm is reproduced from Ahmad and Mohd Ismail [1].

Algorithm 1

Data $f, L, \varphi, x_0, t_0, t_f$ and means for calculating f^* and L^*.

Step 0 Compute or choose a nominal solution $u^0(t), x^0(t)$, and $p^0(t)$.

Set $i = 0, v^0(t) = u^0(t), z^0(t) = x^0(t), \hat{p}^0(t) = p^0(t)$,$t \in [t_0, t_f]$.

Step 1 Compute the parameters $\alpha^i(t)$ and γ^i to satisfy (8).
This is called the parameter estimation step.

Step 2 Compute the multipliers $\lambda^i(t)$ and $\beta^i(t)$ from (9).

Step 3 With specified $\alpha(t)$, $\gamma(t)$, $\lambda(t)$ and $\beta(t)$ solve the MMOP to obtain $u^{i+1}(t), x^{i+1}$, and $p^{1+1}(t)$. This is called the system optimization step.

Step 4 This step tests the convergence and updates the estimate for the solution of the ROP.

$$v^{i+1}(t) = v^i(t) - k_v(v^i(t) - u^{i+1}(t)), \tag{5}$$
$$z^{i+1}(t) = z^i(t) - k_z(z^i(t) - x^{i+1}(t)), \tag{6}$$
$$\hat{p}^{i+1}(t) = \hat{p}^i(t) - k_p(\hat{p}^i(t) - p^{i+1}(t)), \tag{7}$$

where $k_v, k_z, k_p \in (0,1]$ are scalar gains. If $v^{i+1}(t) = v^i(t)$ within a given tolerance stop, else set $i = i+1$ and continue from step 1.

4 The Analysis of the Updating Mechanism

4.1 Decomposition of DISOPE

For the purpose of this analysis, DISOPE algorithm is decomposed as in Definition 1 [1] below.

Definition 1 (Map B and Map C)[1]

Let DISOPE algorithm be decomposed into two distinct maps; map B and map C, say. Then map B is steps 1 to 3 and map C is step 4 of Algorithm 1.

By viewing the algorithm as an application of the composite map **CB** [1], where B is known to be convergent and C corresponds to the set of intermediate steps of the complex algorithm, the overall convergence of such a scheme would be established [2].

4.2 The Updating Mechanism as a Special Case of Newton's Method

Out of the three equations representing map C, equation (5) plays the role of stopping criterion for DISOPE. For convenience (5) is rewritten here

$$v^{i+1}(t) = v^i(t) - k_v(v^i(t) - u^{i+1}(t)) \tag{8}$$

where $u^{i+1}(t)$ is the *computed* control variable produced by map **B** and $v^{i+1}(t)$ is the *estimated* control variable of the updating mechanism. The iterations of the algorithm will stop if the norm of the output of (8) is within a specified tolerance $\varepsilon > 0$. Let equation (8) be map **C(a)**. The norm of map **C(a)** is then $\| v^{i+1}(t) - v^i(t) \| < \varepsilon$. This is the crucial step where the convergence of the whole DISOPE algorithm is determined.

The following analysis of map **C(a)** is based on the assumption that map **B** converges, that is, map **B** produces the values of the control variable $u^{i+1}(t)$. Let $v^i(t) = v^i$ and $u^i(t) = u^i$ in the following analysis for ease of manipulation.

Proposition 1 *Let DISOPE algorithm be partitioned into two distinct maps as in Definition 1 and let map C(a) be equation (8). If map B converges and* $\varepsilon > 0$ *is given such that*

$$\| v^{i+1}(t) - v^i(t) \| < \varepsilon,$$

then map C(a) works at minimizing the total squared error of the output at each iteration; i.e. map C(a) works at minimizing the error function

$$E(v^i) = \frac{1}{2}\left[v^i - u^{i+1}\right]^T \left[v^i - u^{i+1}\right].$$

Proof

Consider the generation of values of the control variables $v^i(t)$ and $u^{i+1}(t)$ from equation (8). The updating mechanism takes in the values of $u^{i+1}(t)$ and updates the values of $v^i(t)$ using equation (8). The updated values are then compared to the current values of $v^i(t)$ and if

$$v^{i+1}(t) = v^i(t) \tag{9}$$

the algorithm stops, else the whole process is repeated. Since DISOPE solves a problem numerically, (9) is satisfied by taking the difference between the two to be less then a specified tolerance, that is, (9) is satisfied by taking $\varepsilon \to 0$,

$$\| v^{i+1} - v^i \| \to 0. \tag{10}$$

From step 4 of Algorithm 1, $k_v \in (0, 1]$, $k_v \neq 0$, hence (8) and (10) imply

$$\| v^i - u^{i+1} \| \to 0. \tag{11}$$

Let $E(v^i)$ be some vector function to be minimized by equation (8) at the ith iteration. The minimum of $E(v^i)$ happens at $\frac{\partial E(v^i)}{\partial v^i} = 0$. Thus let

$$\frac{\partial E(v^i)}{\partial v^i} = E'(v^i) = v^i - u^{i+1}. \tag{12}$$

Integrating (12), we get the following function, referred to as the *error function of the modifier equation*:

$$E(v^i) = \frac{1}{2}\left[v^i - u^{i+1}\right]^T \left[v^i - u^{i+1}\right]. \tag{13}$$

Hence the updating mechanism of DISOPE works at minimizing the total squared error of the output at each iteration.

DISOPE relies on equation (9) for its stopping criterion. Furthermore, for (9) to be satisfied, (8) must converge. Hence this convergence analysis is restricted only to (8) which makes up map **C(a)**. The following theorem proposes a characteristic of the algorithm, which implies its convergence.

Teorem 1 *Let $E(v^i) = \frac{1}{2}[v^i - u^{i+1}]^T[v^i - u^{i+1}]$ be the error function to be minimized by map C(a). If map B converges and $E"(v^i) \geq 1$, then equation (8) is a special case of the Newton's Method.*

***P**roof*

Given the error function

$$E(v^i) = \frac{1}{2}\left[v^i - u^{i+1}\right]^T \left[v^i - u^{i+1}\right] \tag{14}$$

to minimize (12) means to set (13) to zero, that is, $E'(v^i) = 0$. The minimizer of (14) is the root of $E'(v^i) = 0$. Let $v = g(v)$ be an iteration function and let $E'(v^i) = 0$ be equivalent to $v = g(v)$. It follows that, any solution to $E'(v^i) = 0$ is a fixed point of $g(v)$[5]. Being a fixed point, let $v^{i+1} = g(v^i)$. Rewrite

$$\begin{aligned} E'(v^i) &= 0 \\ &= \lambda(v^i)E'(v^i) \\ &= v - v + \lambda(v^i)E'(v^i) \end{aligned}$$

such that

$$v = v + \lambda(v^i)E'(v^i)$$

and

$$g(v^i) = v^i + \lambda(v^i)E'(v^i) \tag{15}$$

The following additional condition is required, that is, if $\bar{v}$ is the solution to $E'(v^i) = 0$, then $g'(\bar{v}) = 0$[5]. Then

$$g'(v^i) = 1 + \lambda'(v^i)E'(v^i) + \lambda(v^i)E''(v^i). \tag{16}$$

When $v^i = \bar{v}$

$$g'(\bar{v}) = 1 + \lambda'(\bar{v})E'(\bar{v}) + \lambda(\bar{v})E''(\bar{v}), \tag{17}$$

with $E'(\bar{v}) = 0$ and $g(\bar{v}) = \bar{v}$. Hence (17) becomes $g'(\bar{v}) = 1 + \lambda(\bar{v})E''(\bar{v})$, implying $\lambda(\bar{v}) = -\frac{1}{E''(\bar{v})}$. Thus (15) becomes

$$g(v^i) = v^i - \frac{E'(v^i)}{E''(v^i)}. \tag{18}$$

With $v^{i+1} = g(v^i)$, (18)becomes

$$v^{i+1} = v^i - \frac{E'(v^i)}{E''(v^i)}. \tag{19}$$

Because $E(v^i)$ is quadratic, it follows that $E''(v^i)$ is a constant and $E''(v^i) > 0$. Let $E''(v^i) \geq 1$, then $0 < \dfrac{1}{E''(v^i)} \leq 1$. Taking $\dfrac{1}{E''(v^i)} = k_v$, (19) becomes

$$v^{i+1} = v^i - k_v E'(v^i) \tag{20}$$

which is exactly equation (8). If $E'(v^i) = f(x)$ then (19) can be rewritten as

$$x^{i+1} = x^i - \frac{f(x)}{f'(x)} \tag{21}$$

which is the Newton's Method. Equation (20) is thus a special case of (21) with $f'(x) \geq 1$.

4.3 The Order of Convergence of the Updating Mechanism

Although convergence of an iterative process is desirable, it is the speed at which it converges that is important for practical purposes. Hence it is necessary to analyze the order of convergence of this algorithm to determine its speed and thus its practicality.

Definition 2(Convergence of order p)

One says that x_n converges to α with (at least) order $p \geq 1$ if $\| x_n - \alpha \| \leq \varepsilon_n$ holds with $\lim_{n\to\infty} \frac{\varepsilon_{n+1}}{\varepsilon_n^p} = c, c > 0$. Furthermore, if $c = 1$, we call this type of convergence sub linear; if $c = 0$, or if the limit does not hold for any $p > 1$, we call this type of convergence super linear.

Following Theorem 1, we state this corollary:

Corollary 1

Map C(a) of DISOPE algorithm converges super linearly.

Proof

Following the proof of Theorem 1, let $\bar{v}$ be a simple root of equation $E'(v^i) = 0$; i.e. $E'(\bar{v}) = 0$. Subtract $\bar{v}$ from both sides of equation (23).

$$v^{i+1} - \bar{v} = v^i - \bar{v} - \frac{E'(v^i)}{E''(v^i)} \tag{22}$$

with $E'(\bar{v}) = 0$, rewrite (1) as follows:

$$v^{i+1} - \bar{v} = (v^i - \bar{v})\left(1 - \frac{E'(v^i) - E'(\bar{v})}{(v^i - \bar{v})E''(v^i)}\right) \tag{23}$$

Using the notation of [7] for Newton's form of interpolation polynomial (see Appendix A), (23) becomes

$$v^{i+1} - \bar{v} = (v^i - \bar{v})\left(1 - \frac{[v^i, \bar{v}]E'}{[v^i, v^i]E'}\right) \tag{24}$$

where $[v^i, \bar{v}]E' = \frac{E'(v^i) - E'(\bar{v})}{v^i - \bar{v}}$ and $[v^i, v^i]E' = E''(v^i)$. Rewrite (24) as

$$\begin{aligned} v^{i+1} - \bar{v} &= (v^i - \bar{v})\left(\frac{[v^i, \bar{v}]E' - [v^i, \bar{v}]E'}{[v^i, v^i]E'}\right) \\ &= (v^i - \bar{v})\left(\frac{\frac{[v^i, v^i]E' - [v^i, \bar{v}]E'}{(v^i - \bar{v})}(v^i - \bar{v})}{[v^i, v^i]E'}\right) \\ &= (v^i - \bar{v})^2\left(\frac{[v^i, v^i, \bar{v}]E'}{[v^i, v^i]E'}\right) \end{aligned}$$

where

$$[v^i, v^i, \bar{v}]E' = \frac{[v^i, v^i]E' - [v^i, \bar{v}]E'}{v^i - \bar{v}}.$$

Therefore

$$\frac{v^{i+1} - \bar{v}}{(v^i - \bar{v})^2} = \frac{[v^i, v^i, \bar{v}]E'}{[v^i, v^i]E'}$$

If $v^i \to \bar{v}$, then

$$\lim_{n\to\infty} \frac{v^{i+1} - \bar{v}}{(v^i - \bar{v})^2} = \frac{[\bar{v}, \bar{v}, \bar{v}]E'}{[\bar{v}, \bar{v}]E'} = \frac{(E'(\bar{v}))''}{2(E'(\bar{v}))'} = \frac{E'''(\bar{v})}{2E''(\bar{v})}$$

where $[\bar{v}, \bar{v}, \bar{v}]E' = \frac{1}{2!}(E'(\bar{v}))''$ and $[\bar{v}, \bar{v}]E' = \frac{1}{1!}(E'(\bar{v}))'$. Since $E(v^i)$ is quadratic, $E'''(\bar{v}) = 0$, making

$$\lim_{n\to\infty} \frac{v^{i+1} - \bar{v}}{(v^i - \bar{v})^2} = 0$$

Hence DISOPE has a super linear convergence as is the case when $f(x)$ is quadratic in Newton's method

5 The Newton-like Properties

One of the attractive features of the algorithm is the integration of parameters from both the real problem and its model before arriving at the optimal solution. In view of the above analysis, DISOPE converges in just one iteration whenever no model-reality difference is introduced.

DISOPE requires an initial solution to start the iterations. A recommended one is the solution of the relaxed MMOP with $\alpha(t) = 0, r_1 = r_2 = 0$. Since DISOPE is composed of maps **B** and **C**, with map **B** supplying the input for map **C**, the initial solution is for map **B**. However, the initial solution of **B** affects its output and hence the input for map **C**. Thus it is imperative that a good initial solution to map **B** would translate into a good input to map **C**.

It is a well-known fact that the choice of initial solution is crucial in Newton's method to speed up convergence. The closer the initial guess, the faster it reaches the optimal solution. The same property is tested for DISOPE algorithm.

In DISOPE the real problems are modeled as linear quadratic regulators. It is customary for DISOPE to use the identities for Q and R; weights for the performance index. It is noticed however that other choices for Q and R; hence the initial solutions; have different effects on the rate of convergence. In fact the right choice of weights would tremendously cut down on the number of iterations as shown in the numerical examples below.

5.1 Numerical Examples

Each of the two examples below is simulated with two different initial solutions by giving different values to the weighting matrices of the performance index.

Example 1

Consider the continuous stirred tank reactor problem taken from Kirk [8]. The real optimization problem (ROP) is as follows:

$$\min_{u(x)} J* = \int_0^{0.78} (x_1^2 + x_2^2 + 0.1u^2)dt$$

subject to

$$\begin{aligned} \dot{x_1} &= -(x_1 + 0.25) + (x_2 + 0.5)\exp\left(\frac{25x_1}{x_1 + 2}\right) - (1 + u)(x_1 + 0.25) \\ \dot{x_1} &= 0.5 - x_2 - (x_2 + 0.5)\exp\left(\frac{25x_1}{x_1 + 2}\right) \\ x(0) &= [0.05\ 0]^T \end{aligned}$$

The modified model (MOP) used in DISOPE algorithm is:

$$\min_{u(x)} J = \int_0^{0.78} (x^T Q x + u^T R u + \gamma(t))dt$$

subject to

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 4.25 & 1 \\ -6.25 & -2 \end{bmatrix} x(t) + \begin{bmatrix} -0.25 \\ 0 \end{bmatrix} u(t) + \alpha(t) \\ x(0) &= [0.05\ 0]^T \end{aligned}$$

where $x(t) \in R^2, u(t) \in R, \gamma(t) \in R$ and $\alpha(t) \in R^2$. Two different initial solutions are used in the simulations of this proble. The first uses the value of the weighting matrix $Q = 2I_2$. Teh second initial solution uses the value of $Q = \begin{bmatrix} 22.40 & 4.480 \\ 4.480 & 0.896 \end{bmatrix}$. In both cases the value of R is kept constant at $R = 0.2$. During the iterations of DISOPE, no turning was done to the values of the parameters r_1 and r_2 and k_v, k_z and k_p, that is, r_1 and r_2 are set to zero and k_v, k_z and k_p are set to one. The integration step taken is $h = 0.01$, and the tolenrance considered for the convergence is $tol = 0.01$.

Table 1: Results of simulations with different values of Q

	No. of iteration	$\lVert x_{init} - x_{opt} \rVert$	Values of J
$Q = 2I_2$	10	0.1977	0.0281
$Q = \begin{bmatrix} 22.40 & 4.480 \\ 4.480 & 0.896 \end{bmatrix}$	4	0.0332	0.0280

The 2-norms between the initial solutions and the optimal solution are calculated to gauge the ‘closeness’ of the initial guess to the converged solution. The results of the simulations are summarized in Table 1 above. Figures 1(a) and (b) below show the related results.

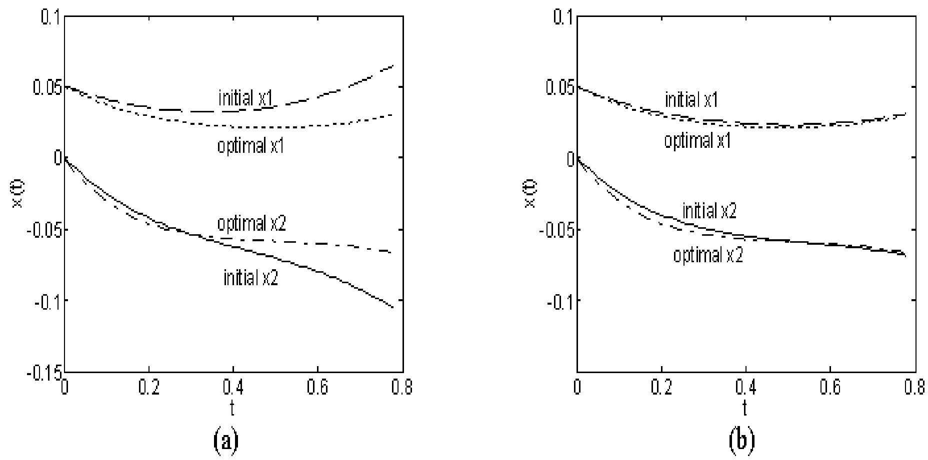

Figure 1: : Comparisons of closeness between two different initial solutions and the optimal solution (a) Initial solution for $Q = 2I2$; (b) Initial solution for $Q = [22.404.480; 4.4800.896]$.

Example 2

Consider a fourth order non-linear system representing a horizontal planar revolute/prismatic two degrees of freedom robot manipulator taken from Craig [6]. ROP is defined as:

$$\min_{u(t)} \frac{1}{2}\int_{t_0}^{4}[x_1^2 + x_2^4 + x_3^2 + x_4^4 + u_1^2 + 0.1u_2^4]dt$$

subject to

$$\begin{aligned}
\dot{x_1} &= x_2; x_1(0) = 2, x_1(4) = 0\\
\dot{x_2} &= \frac{u_1 - 4x_2x_4(x_3+0.5)}{1+2(x_3+0.5)}; x_2(0), x_2(4) = 0\\
\dot{x_3} &= x_4; x_3(0) = 1, x_3(4) = 0\\
\dot{x_4} &= (x_3+0.5)x_2^2 + 0.5u_2; x_4(0) = 0, x_4(4) = 0
\end{aligned}$$

where $x_1(t)$ and $x_2(t)$ are the angular position and velocity of link 1, $x_3(t)$ and $x_4(t)$ are the angular position and velocity of the prismatic link 2. $u_1(t)$ and $u_2(t)$ are the driving torque and force of the two links.
MOP is taken as a linear quadratic model representing small perturbations about the equilibrium point at the origin:

$$\min_{u(t)} \frac{1}{2}\int_{0}^{4}(x^TQx + u^TRu + \gamma(t))dt$$

subject to

$$\begin{aligned}
\dot{x} &= Ax + Bu + \alpha(t)\\
x(0) &= [2\ 0\ 1\ 0]^T, x(4) = [0\ 0\ 0\ 0]^T
\end{aligned}$$

with

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0.5 \end{bmatrix}$$

where $x(t) \in R^4, u(t) \in R^2, \gamma(t) \in R$ and $\alpha(t) \in R^4$. In this example the value of the weighting matrix Q is kept constant at

$$Q = \begin{bmatrix} 0.015 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The weighting matrix R on the other hand is given two different values; $R = I_2$ and $R = 1.5I_2$.
With the choices of Q and $R = I_2$ together with $r_1 = 1, r_2 = 0, k_v = k_z = 0.25, k_p = 1$ DISOPE converged in 91 iterations in the first simulation. The numerical integration step used was $h = 0.05$ and a tolerance of 0.01 was specified for convergence. To see the effect of different initial solution on the convergence rate of the algorithm, we simulated next $R = 1.5I_2$. The value of r_1 is now zero and the values of the other inputs are kept the same. The result is a faster convergence. The algorithm converged in 77 iterations. The 2-norms between the two initial solutions and the optimal solution are calculated and summarized in Table 2 below.

Table 2: Results of simulations with different values of R

	No. of iteration	$\|x_{init} - x_{opt}\|$	Value of J
$R = I_2$	91	4.6735	6.3788
$R = 1.5I_2$	77	3.7426	6.3788

The two examples above show that the closer the initial solution to the optimal solution, the faster the convergence.

6 Conclusion

A convergence analysis of the updating mechanism cum stopping criterion of DISOPE algorithm was presented. An appropriate error function was proposed. Based on the proposition, a theorem on the likeness of the mechanism to Newton's method was proved. The convergence of the mechanism was establish based on the convergence of Newton's method. The order of convergence of the algorithm was also analyzed. The algorithm has a super linear convergence equating that of Newton's whenever the function involved is quadratic.

The algorithm converges in only one iteration if no model-reality difference is introduced. Furthermore DISOPE possesses the Newton-like property in the choice of the initial solution used to start the iterations. It is noticed from the two examples given, the closer the initial solution to the optimum, the faster the convergence.

References

[1] R. Ahmad, and B.A.A Mohd Ismail, *"The Convergence Analysis of the Updating Mechanism of the Dynamic Integrated Systems Optimization Algorithm,"*, Prosiding Simposium Kebangsaan Sains Matematik Ke-10,2002, pg 326-332.

[2] M.S. Bazaraa, H.D. Sherali and C.M. Shetty, *"Nonlinear Programming: Theory and Algorithms"*, John Wiley and Sons, Inc., New York, 1993.

[3] V.M. Becerra, *"Development and Applications of Novel Optimal Control Algorithms"*, PhD Thesis, Dept. of Electrical, Electronics and Information Engineering, City University, London, 1994.

[4] A.E. Bryson and Y.C. Ho, *"Applied Optimal Control"*, Hemisphere Publishing Corporation, Washington D¿C., 1975.

[5] J.L Buchanan and P.R. Turner, *"Numerical Methods and Analysis"*, McGraw-Hill, Inc., New York, 1992.

[6] J.C. Craig, *", Introduction to Robotics"*, Addison-Wesley, Reading, USA, 1989.

[7] W.Gautschi, *"Numerical Analysis"*, Birkhuser, Boston, 1997.

[8] D.E.Kirk, *", Optimal Control Theory"*, Prentice-Hall Inc., Englewood Cliffs, 1970.

[9] E. Polak, *"Optimization: Algorithms and Consistent Approximations"*,Springer-Verlag New York Inc., New York, 1997.

[10] P.D. Roberts, *"An Algorithm for Optimal Control of Nonlinear Systems with Model-Reality Differences"*, Proceedings of 12th IFAC World Congress on Automatic Control, 8(1993), 407-412.

[11] P.D. Roberts, *"Two-Dimensional Analysis of an Iterative Nonlinear Optimal Control Algorithm"*, IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications. 49: No. 6, (2002), 872-878.

Appendix A

Newton's Form of the Interpolation Polynomial [6]

Given $n+1$ points $x_0, x_1, ..., x_n$ and values of $f_i = f(x_i)$, of some function at these points, a polynomial $p \in \mathcal{P}_n$ such that $p(x_i) = f_i; i = 0, 1, ..., n$ interpolate f(x) at these points. We denote the unique polynomial $p \in$ interpolating f at the distinct points $x_0, x_1, ..., x_n$ by

$$\begin{aligned} p_n &= (f; x_0, x_1, ..., x_n; x) = p_n(f; x), \quad n = 0, 1, 2, ... \\ p_n(f; x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + ... + a_n(x - x_0)(x - x_1)...(x - x_{n-1}) \end{aligned}$$

The constants involved can be determined by

$$\begin{aligned} f_0 &= a_0 \\ f_1 &= a_0 + a_1(x_1 - x_0) \\ f_2 &= a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \\ \vdots &= \vdots \end{aligned}$$

Thus the constants are

$$a_0 = f_0 \tag{25}$$

$$a_1 = \frac{f_1 - f_0}{x_1 - x_0} \tag{26}$$

$$a_2 = \frac{f_2 - f_1}{(x_2 - x_0)(x_2 - x_1)} \tag{27}$$

The following notation is used to denote the constants:

$$a_n = [x_0, x_1, ..., x_n]f, \quad n = 0, 1, 2, \ldots \tag{28}$$

Hence (25),(26) and (27)respectively look like the following

$$\begin{aligned} a_0 &= f_0 \\ a_1 &= [x_0, x_1]f \\ a_2 &= [x_0, x_1, x_2]f \end{aligned}$$

The right hand side of (28) is called the *nth divided difference of* f relative to the nodes $x_0, x_1, ..., x_n$. The name comes from the property

$$[x_0, x_1, ..., x_n]f = \frac{[x_0, x_1, ..., x_k]f - [x_0, x_1, ..., x_{k-1}]f}{x_k - x_0}. \tag{29}$$

Another property that will be useful in this analysis is

$$[x_0, x_0, ..., x_0]f = \frac{1}{n!} f^{(n)}(x_0). \tag{30}$$

(28) is a symmetric function; that is, the permutation of variables does not affect the value of the function.