J. Austral. Math. Soc. Ser. B 39(1997), 195–213

ON OPTIMAL CONTROL OF MULTI-LINK VERTICAL PLANAR ROBOT ARMS SYSTEMS MOVING UNDER THE EFFECT OF GRAVITY

H. W. J. LEE, K. L. TEO and L. S. JENNINGS[1]

(Received 22 March 1996; revised 24 April 1996)

Abstract

How to obtain a workable initial guess to start an optimal control (control parametrization) algorithm is an important question. In particular, for a system of multi-link vertical planar robot arms moving under the effect of gravity and applied torques (which can exhibit chaotic behaviour), a non-workable initial guess of torques may cause integration failure regardless of what numerical packages are used. In this paper, we address this problem by introducing a simple and intuitive "Blind Man" algorithm. Theoretical justification as well as a numerical example is provided.

1. Introduction

In the robotics literature, the closed-form model of the robot arm which is directly derived from the Lagrangian equations, is usually adopted to describe the dynamics of the system instead of using the state differential equations. However, models in terms of state differential equations are preferred in the optimal control literature. See [7, 12] for details. In this work, we focus mainly on the computations of the optimal control of multi-link planar robot arm systems. Hence, it is more natural to describe the dynamics of the robot arm using state differential equations.

One can design different cost functionals to achieve different objectives, be it minimum time, or quadratic regulator, or any more general cost functional. The problem of finding the control (torque), as a function of time, which minimizes this cost functional, subject to the state differential equations, with or without other constraints, is a standard optimal control problem.

There are a number of efficient computational techniques available nowadays to tackle this problem numerically, for example, see [2, 8–11, 13]. Software packages

[1]Department of Mathematics, The University of Western Australia, Nedlands, WA 6907, Australia

are also available now implementing these ideas such as MISER3.1 [3, 13] which was developed based on the method of control parametrization and nonlinear constrained optimization techniques.

In optimal control software packages, such as MISER3.1, an initial control is required to start the corresponding algorithm. In MISER3.1, the user needs to provide an initial time-parametrized control function to start the algorithm.

One of the shortcomings of many of these iterative algorithms is that the user has to supply a good guess. If the initial solution is too far from the global minimum, the iterative algorithms may be attracted to some local minimum. It is worse if the system exhibits some "chaotic" behaviour, in which the bad initial guess may cause integration failures.

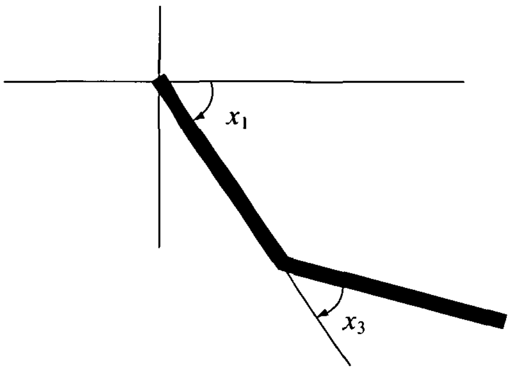

FIGURE 1. The planar two-link robot arm system (Double Pendulum).

Many robotic problems are inherently chaotic when controls (or torques) are not applied to the systems. For example, a vertical planar two-link robot arm (with gravity taken into account) is virtually a planar double pendulum when no controls (or torques) are applied at the joints (see Figure 1). This double pendulum is one of the early chaotic systems considered in the literature. The chaotic behaviour may not be significant when the time horizon is small. However, when the time horizon is large, the chaotic behaviour developed is no longer a matter that can be ignored.

For a certain fixed way of parametrization, with a "wild guess" of the control function as an initial solution to start the iterative algorithm, it is quite likely that the robot arms will spin chaotically and cause integration failure for the differential equation solver. So it may not even be able to complete the first iteration successfully, let alone compute the cost associated with that initial control function. We call this kind of initial control function non-workable. However, if an initial control function does not cause integration failures and the subsequent optimization task can be performed without any integration difficulties, we call it a workable initial control function.

Of course if one has enough experience with the dynamics of that particular robot arm system, or one has enough physical insight, a "wild guess" may not be necessary. However, such "physical insights" and "experiences" may not be easy to gain in general. Note that the concept of a workable initial control function is different from that of a feasible control function. An initial control function may be infeasible, however, it could as well be a workable one if the subsequent iterates iterate back to the feasible region and converge to the minimum.

This paper proposes a systematic way to obtain such a workable initial control function to start the iterative optimal control algorithm. This systematic scheme, is referred to as the Blind Man method (BM). Although this systematic method is rather simple and intuitive, it provides a practical approach in solving many highly complex optimal control problems involving robot arms under the effect of gravity.

A brief introduction to the formulation of the standard optimal control problem and the control parametrization method are given in the Appendix.

2. Problem formulation

Consider a vertical planar multi-link robot arms system moving under the effects of gravity. Its dynamics belongs to the following class of affine systems:

$$\dot{x} = f(x) + B(x)u, \tag{2.1}$$

$$x(0) = x_0, \tag{2.2}$$

where $u : [0, T) \mapsto U \subset \mathbb{R}^r$ is the control vector. The set U defines the control bounds and it is some compact subset of $\mathbb{R}^r$. Let $\mathscr{U}$ be the class of all such bounded measurable functions u. Elements of $\mathscr{U}$ are called admissible controls, and $\mathscr{U}$ is called the class of admissible controls. The functions $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ and $B : \mathbb{R}^n \mapsto \mathbb{R}^{n\times r}$ are given twice continuously differentiable functions. The state vector x consists of angles of the arms and their angular velocities. For example, the dynamics of the vertical planar two-link robot arm system are governed by (2.1) with $x = [x_1, x_2, x_3, x_4]^{\mathrm{T}}$ where x_1 and x_3 are the angles as indicated in Figure 1, and x_2 and x_4 their respective angular velocities. The corresponding forms of the functions f and B are given, respectively, by

$$f(x) = \begin{bmatrix} x_2 \\ \dfrac{(((x_4H_{2,2} + x_1H_{1,2})x_2 + (x_2 + x_4)x_4H_{2,2})h_c - (g_1H_{2,2} - g_2H_{1,2}))}{\det(H)} \\ x_4 \\ \dfrac{(((-x_4H_{2,1} + x_1H_{1,1})x_2 - (x_2 + x_4)x_4H_{2,1})h_c - (g_2H_{1,1} - g_1H_{2,1}))}{\det(H)} \end{bmatrix}$$

and

$$B(x) = \begin{bmatrix} 0 & 0 \\ -\dfrac{H_{2,2}}{\det(H)} & \dfrac{H_{1,2}}{\det(H)} \\ 0 & 0 \\ -\dfrac{H_{1,2}}{\det(H)} & \dfrac{H_{1,1}}{\det(H)} \end{bmatrix},$$

where

$$\begin{aligned} h_c &= m_2 l_1 c_2 \sin(x_3), \\ g_1 &= m_1 c_1 g \cos(x_1) + m_2 g (c_2 \cos(x_1 + c_3) + l_1 \cos(x_1)), \\ g_2 &= m_2 c_2 g \cos(x_1 + x_3), \\ H &= \begin{bmatrix} H_{1,1} & H_{1,2} \\ H_{2,1} & H_{2,2} \end{bmatrix}, \\ H_{1,1} &= m_1 c_1^2 + I_1 + m_2 (l_1^2 + c_2^2 + 2 l_1 c_2 \cos(x_3)) + I_2, \\ H_{1,2} &= m_2 l_1 c_2 \cos(x_3) + m_2 c_2^2 + I_2, \\ H_{2,1} &= H_{1,2}, \\ H_{2,2} &= m_2 c_2^2 + I_2. \end{aligned}$$

The constants l_j, m_j, I_j, c_j are the length, mass, moment of inertia, distance to the centre of gravity measured from the supporting joint, of link j, respectively, for $j = 1, 2$, and g is the gravitational acceleration.

The problem may also be subject to any or all of the inequality constraints and equality constraints as specified by (A.3) and (A.4) in the Appendix. These constraints correspond to the limitations imposed on the robotic systems due to various practical requirements. For example, obstacles for which the robot arm system are to avoid hitting. The software package MISER3.1 [3] is an optimal control software package that can handle this constrained optimal control problem efficiently (see [13] for details). However, it would have suffered from the same numerical difficulties when the initial guess of the control function is not workable. Thus, for simplicity, we consider the unconstrained case in this paper. The extensions to constrained cases are straightforward.

In view of the control parametrization method (see Appendix), it is clear that once the state differential equations (2.1) and cost functional (A.2) are fixed, one has to supply information on $u^0(t)$, the initial guess of the control to start the algorithm. If the initial guess $u^0(t)$ is a workable one, and all subsequent function evaluation are workable, there are three pieces of information from the computed results after the convergence of an outer iteration, namely:

(i) the optimal control function $u^*(t)$ of the same class of representation as $u^0(t)$ with a pre-specified accuracy,
(ii) the optimal trajectory (extremals) $x^*(t)$, and
(iii) the cost associated with them, $J(u^*)$.

However, if the initial guess $u^0(t)$ is not a workable one, numerical integration failures occur.

To summarize the above, we introduce the idea of a "selective rule" $M(x(0), u^0, i)$ for a fixed set of state differential equations and a fixed cost functional, where $i \in [1, 2, 3, 4]$. When $i = 1$, M gives the information about workability of u^0, i.e. u^0 is workable with respects to the initial condition $x(0)$ if and only if $M(x(0), u^0, 1) = 1$; otherwise, $M(x(0), u^0, 1) = 0$. If $M(x(0), u^0, 1) = 1$, then

$$M(x(0), u^0, 2) = u^*(t),$$
$$M(x(0), u^0, 3) = x^*(t),$$
$$M(x(0), u^0, 4) = J(u^*).$$

If $M(x(0), u^0, 1) = 0$, then $M(x(0), u^0, i)$ is undefined for $i = 2, 3, 4$.

Note that M itself is not a function. It is what we call a "selective rule". It is an integer when $i = 1$, is a real number when $i = 4$, and is a r or n dimensional vector-valued function of time when $i = 2$ or $i = 3$, respectively. The reason for introducing this "selective rule" $M(\cdot, \cdot, \cdot)$ is for the ease of notation in the following discussions, and thus makes the presentation of the "Blind Man" algorithm compact.

We can now formulate the problem of finding a workable initial guess as follows:

Subject to the dynamical system (2.1) and the cost functional (A.2) find a u^0 such that $M(x(0), u^0, 1) = 1$.

3. Mathematical preliminaries

The following theorems are well known in the theory of differential equations. For details, see [1].

3.1. Continuous dependence of solutions of Initial Conditions

THEOREM 1. *Consider the system*

$$\dot{x} = h(t, x) \qquad t \geq s \geq 0 \tag{3.1.a}$$

$$x(s) = \xi \tag{3.1.b}$$

where $h : [0, \infty) \times \mathbb{R}^n \mapsto \mathbb{R}^n$ *is a nonlinear function. Suppose that for each fixed but arbitrary* $\xi \in \mathbb{R}^n$, *the function* $t \mapsto h(t, \xi)$ *is measurable on* $[0, \infty)$. *Moreover,* *for almost all* $t \in [0, \infty)$, $\xi \mapsto h(t, \xi)$ *is continuous on* $\mathbb{R}^n$, *and that there exists a non-negative measurable function* $K \in L_1^{\mathrm{loc}}$ *such that*

$$|h(t, \xi)| \le K(t)[1 + |\xi|] \qquad \textit{for all} \quad \xi \in \mathbb{R}^n,$$
$$|h(t, \xi) - h(t, \eta)| \le K(t)|\xi - \eta| \qquad \textit{for all} \quad \xi, \eta \in \mathbb{R}^n.$$

Then $(s, \xi) \mapsto x(t)$ *is a continuous mapping from* $[0, t] \times \mathbb{R}^n$ *to* $\mathbb{R}^n$ *for each fixed* $t > 0$, *and* $t \mapsto x(t)$ *is continuous from* $[s, \infty)$ *to* $\mathbb{R}^n$ *for fixed* $(s, \xi) \in (0, \infty) \times \mathbb{R}^n$.

In particular, if s is fixed at 0, the above theorem shows that the solution of the system (3.1) is continuously dependent on initial conditions. Consider the system (2.1). If the control $u(t)$ is fixed, then the system is in the form of (3.1). More precisely,

$$h(t, x) \equiv g(t, x(t), u(t)).$$

Hence by Theorem 1, the property of continuous dependence on initial condition remains valid for such systems. We summarize this idea by restating the theorem as follows.

COROLLARY 1. *Given a fixed control* $u \in \mathscr{U}[0, T)$, *suppose the system* (2.1) *satisfies all the conditions of Theorem* 1. *Then* $(0, \xi) \mapsto x(t)$ *is a continuous mapping from* $\mathbb{R}^n$ *to* $\mathbb{R}^n$ *for each fixed* $t \in (0, T]$ *and* $t \mapsto x(t)$ *is continuous from* $[0, T]$ *to* $\mathbb{R}^n$ *for fixed* $\xi \in \mathbb{R}^n$.

On the basis of Corollary 1 we consider system (2.1) with a fixed $u \in \mathscr{U}[0, T)$. Then for any $\epsilon > 0$, there exists a $\delta > 0$ such that

$$\max_{0 \le t \le T} |x(t|\xi_1) - x(t|\xi_2)| \le \epsilon \qquad \text{whenever} \qquad |\xi_1 - \xi_2| \le \delta.$$

This continuity property will be a main idea behind the Blind Man method to be proposed in this paper.

3.2. Continuous dependence of solutions on inputs (controls)

THEOREM 2. *Consider the controlled system*

$$\dot{x} = g(t, x, u) \qquad t \in [0, T], \tag{3.2.a}$$
$$x(0) = x_0, \tag{3.2.b}$$

where $u : [0, T) \mapsto U \subset \mathbb{R}^r$, *and the set* U *defines the control bounds and it is some compact subset of* $\mathbb{R}^r$. *Let* $\mathscr{U}$ *be the class of all such bounded measurable function* u, *and let $\rho : \mathcal{U} \times \mathcal{U} \mapsto [0, \infty)$ be the metric on $\mathcal{U}$. Thus $(\mathcal{U}, \rho)$ is a complete metric space. Suppose there exists a constant K_1 (possibly dependent on U) such that*

$$|g(t, \xi, w)| \le K_1[1 + |\xi|] \quad \textit{for all} \quad \xi \in \mathbb{R}^n,$$
$$|g(t, \xi_1, w) - h(t, \xi_2, w)| \le K_1|\xi_1 - \xi_2| \quad \textit{for all} \quad \xi_1, \xi_2 \in \mathbb{R}^n.$$

Then there exists a constant K_2, depending only on K_1, such that

$$\|x_u - x_v\| \equiv \sup_{t\in[0,T]} |x_u(t) - x_v(t)| \le K_2\rho(u, v)$$

for all $u, v \in \mathcal{U}$, where x_u and x_v denotes the responses of the system (3.2) *corresponding to controls u and v respectively.*

4. The Blind Man (BM) method

Although the method to be proposed is rather simple and intuitive, it is practical if one does not have any "physical insights" to get a workable initial guess.

A blind man, as we know, always checks from where he is standing before he moves. In other words, if he is at point A and he wants to go to point B, he starts off checking things at point A first. Then, he goes a bit further towards point B. Again, he will check this new point out until he is sure about the point before he moves on a bit further against towards point B.

In view of this "blind man approach", one can see that for problems with a given $x(0)$, a wild guess of u^0 may not be advisable. The first step of our approach is to free the initial condition $x(0) = x_0$ to a point denoted by ξ^0 for which a workable initial control denoted by μ^0 can be easily obtained. Then, we shall move the initial condition slowly towards the specified initial condition x_0 in view of Corollary 1. With this in mind, we have the following observation

If all the links of the robot arm hand at rest, pointing vertically downwards initially, without supplying any torque for the whole time horizon, the system will remain in this state, that is, $x(t)$ is constant for all $t \in [0, T]$. Of course there should be no integration failures in this starting configuration when we keep the applied torque to be constantly zero, or small enough in magnitude, through $[0, T)$. Thus, for a given vertical planar multi-link robot arm system moving under the effect of gravity together with a specified cost functional, if $\xi^0 = [\frac{-\pi}{2}, 0, 0, \dots, 0]^{\mathrm{T}}$, the $M(\xi^0, \mu^0, 1) = 1$ when μ^0 is constantly zero, and with the control search bounds that are small enough.

If the required bounds specified in $\mathcal{U}$ is too large to make $M(\xi^0, \mu^0, 1) = 1$, however, one can use smaller bounds on the control search space to compute $M(\xi^0, \mu^0, \cdot)$. If $M(\xi^0, \mu^0, 2)$ hits the bounds, we then widen the bounds a bit, and reset μ^0 as $M(\xi^0, \mu^0, 2)$, and compute $M(\xi^0, \mu^0, \cdot)$ again with this new μ^0. This widening process is repeated until the search bounds hits the required bounds, or $M(\xi^0, \mu^0, 2)$ does not hit the bounds anymore. After that, reset the search bounds as the required bounds specified in $\mathscr{U}$ in all the subsequent computations. We can formalize all these bound widening processes as follows.

Recall that $\mathscr{U}$ is the class of all bounded measurable functions $u : [0, T) \mapsto U \subset \mathbb{R}^r$, where U defined the control bounds. Let

$$u = \{w \in \mathbb{R}^r \mid \alpha_k \le w_k \le \beta_k \text{ for } k = 1, 2, \dots, r\}.$$

In this paper, we assume that $\alpha_k < 0$ and $\beta_k > 0$ for all $k = 1, 2, \dots, r$. The reason for this assumption is to ensure that the constant zero control is strictly in the interior of $\mathscr{U}$. We can see the relevance to this in the later part of this section. Nevertheless, this assumption is valid in most of the practical robot arm systems. Now we can define the set U_γ as

$$U_\gamma = \{w \in \mathbb{R}^r \mid \gamma\alpha_k \le w_k \le \gamma\beta_k \text{ for } k = 1, 2, \dots, r\},$$

and let the class $\mathscr{U}_\gamma$ be the class of all bounded measurable functions $u : [0, T) \mapsto U_\gamma \subset \mathbb{R}^r$. In the following discussions, we assume that the control function is parametrized by piecewise constant functions. Hence, for a certain fixed way of parametrization, a control function $\mu \in \mathscr{U}_\gamma$ takes the form

$$\mu^k = \sum_{j=0}^{M_k} h_{k,j} I_{[t_{k,j}, t_{k,j+1})}(t) \tag{4.1}$$

for $k = 1, 2, \dots, r$, $j = 0, 1, 2, \dots, M_k$, where $M_k + 1$ is the number of partitions of the time-axis of the k-th element of the control, and $I_{[t_{k,j}, t_{k,j+1})}(\cdot)$ is the indicator function defined as

$$I_{[t_{k,j}, t_{k,j+1})}(t) = \begin{cases} 1 & t \in [t_{k,j}, t_{k,j+1}) \\ 0 & \text{otherwise.} \end{cases}$$

Since $\mu \in \mathscr{U}$, for all $j = 0, 1, 2, \dots, M_k$, the coefficients $h_{k,j}$ are bounded by $\gamma\alpha_k \le h_{k,j} \le \gamma\beta_k$ for $k = 1, 2, \dots, r$. Now, let $\hat{\mu}$ be a particular control function in $\mathscr{U}_\gamma$ with its corresponding coefficients denoted by $\widehat{h_{k,j}}$. Thus, we can define $\widehat{\overline{h_{k,j}}}$ as

$$\widehat{\overline{h_{k,j}}} = \min\left\{\gamma\beta_k - \widehat{h_{k,j}}, \widehat{h_{k,j}} - \gamma\alpha_k\right\}.$$

We can then define the sets $V_{k,j}(\hat{\mu}, \mathscr{U}_\gamma, \vartheta)$ as

$$V_{k,j}(\hat{\mu}, \mathscr{U}_\gamma, \vartheta) = \left\{w \in \mathbb{R} \mid (\widehat{h_{k,j}} - \vartheta\widehat{\overline{h_{k,j}}}) \le w \le (\widehat{h_{k,j}} + \vartheta\widehat{\overline{h_{k,j}}})\right\}$$

for $k = 1, 2, \dots, r$ and $j = 0, 1, 2, \dots, M_k$. Hence, for each given $\hat{\mu} \in \mathcal{U}_\gamma$, with $\hat{\mu}$ not on the boundary of $\mathcal{U}_\gamma$, we can define a neighbourhood class of $\hat{\mu}$ as the class of controls given in the same form as in (4.1) with the coefficients $h_{k,j}$ restricted to the set $V_{k,j}(\hat{\mu}, \mathcal{U}_{\gamma,\vartheta})$ for each k and j. We denote this neighbourhood class of $\hat{\mu}$ as $\mathcal{U}_{\gamma,\vartheta}(\hat{\mu})$. Clearly, $\mathcal{U}_{\gamma,\vartheta}(\hat{\mu})$ is a subset of $\mathcal{U}_\gamma$. Let $\Gamma(\mathcal{U}_{\gamma,\vartheta}(\hat{\mu}))$ denote a real number defined by

$$\Gamma(\mathcal{U}_{\gamma,\vartheta}(\hat{\mu})) = \max_k \left\{ \max \left\{ \left| \frac{\min_j \left\{ (\widehat{h_{k,j}} - \vartheta \widehat{\overline{h_{k,j}}}) \right\}}{\alpha_k} \right|, \left| \frac{\max_j \left\{ (\widehat{h_{k,j}} + \vartheta \widehat{\overline{h_{k,j}}}) \right\}}{\beta_k} \right| \right\} \right\}.$$

Note that $\Gamma(\mathcal{U}_{\gamma,\vartheta}(\hat{\mu}))$ gives the smallest γ' such that $\mathcal{U}_{\gamma,\vartheta}(\hat{\mu}) \subset \mathcal{U}_{\gamma'}$, that is,

$$\Gamma(\mathcal{U}_{\gamma,\vartheta}(\hat{\mu})) = \arg \left\{ \min_{\gamma'} \{ \mathcal{U}_{\gamma,\vartheta}(\hat{\mu}) \subset \mathcal{U}_{\gamma'} \} \right\}.$$

Let $M(\cdot, \cdot, \cdot \mid \mathcal{W})$ denote the "selective rule" of the optimization process the same as in $M(\cdot, \cdot, \cdot)$ but with the control search space $\mathcal{U}$ replaced by $\mathcal{W}$.

Suppose $\xi^0 = [-\pi/2, 0, 0, \dots, 0]^{\mathrm{T}}$, and $\mu^0 = 0$ for all $t \in [0, T)$. Let $N > 1$ be a fixed large positive integer and ϵ be a small positive real number, we then have the following algorithm.

ALGORITHM 1.

```
step 0: Set $\gamma = 1$, and $\mu := \mu^0$.
step 1: If $M(\xi^0, \mu, 1 \mid \mathcal{U}_\gamma) = 1$, set $\mu := M(\xi^0, \mu, 2 \mid \mathcal{U}_\gamma)$, goto step 2;
        otherwise set $\gamma := \frac{\gamma}{2}$, goto step 1.
step 2: If $\mu$ hits the bounds of $\mathcal{U}_\gamma$, then goto step 3;
        otherwise, reset $\gamma = 1$, set $\mu^0 := \mu$, then stop.
step 3: If $\gamma = 1$, set $\mu^0 := \mu$, then stop;
        otherwise, set $\bar{\gamma} := \gamma$ and $\eta := \frac{1-\gamma}{N}$, then goto step 4.
step 4: Set $\gamma := \bar{\gamma} + \eta$, if $M(\xi^0, \mu, 1 \mid \mathcal{U}_\gamma) = 0$, set $\eta := \frac{\eta}{2}$, goto step 5,
        otherwise set $\mu := M(\xi^0, \mu, 2 \mid \mathcal{U}_\gamma)$, goto step 2.
step 5: If $\eta > \epsilon$, goto step 4
        otherwise set $\vartheta = 1$, goto step 6.
step 6: Set $\mathcal{W} := \mathcal{U}_{\gamma,\vartheta}(\mu)$. If $M(\xi^0, \mu, 1 \mid \mathcal{W}) = 1$, set $\gamma := \Gamma(\mathcal{W})$,
        set $\mu := M(\xi^0, \mu, 2 \mid \mathcal{W})$, goto step 3;
        otherwise, set $\vartheta := \frac{\vartheta}{2}$, goto step 6.
```

The use of the above algorithm is to compute an appropriate μ^0, such that $M(\xi^0, \mu^0, 1) = 1$ regardless of the size of the bounds on the control search space $\mathcal{U}$. Note that from step 5 to step 6, μ is not on the boundary of $\mathcal{U}_\gamma$. As long as the constant zero control μ^0 in step 0 is in the interior of $\mathcal{U}$, all subsequent μ computed just before executing step 6 are in the interior of each corresponding $\mathscr{U}_\gamma$, where γ has the value just before executing step 6. Hence, the neighbourhood classes $\mathscr{U}_{\gamma,\vartheta}(\mu)$ exist, and their constructions remain valid.

Note also that infinite loopings will note occur in the above algorithm within step 1, and within step 6. The reasons for this follows readily from the continuous dependence property of the system trajectory on controls discussed in Section 3. Let $u = 0$ for all $t \in [0, T)$. For $0 < \gamma \leq 1$, there exists a metric $\rho(\cdot, \cdot)$ defined on $\mathscr{U}$, such that $\rho(u, v)$ decreases monotonically when γ decreases for any control $v \in \mathscr{U}_\gamma$. So, by Theorem 2, $\|x_u - x_v\|$ can be made arbitrarily small by reducing $\gamma > 0$ in order to avoid integration failures. Hence, the loopings within step 1 will terminate at a finite number of loops. In step 6, μ is in the interior of the neighbourhood classes $\mathscr{U}_{\gamma,\vartheta}(\mu)$. For $0 < \vartheta \leq 1$, $\rho(\mu, w)$ decreases monotonically when ϑ decreases for any control $w \in \mathscr{U}_{\gamma,\vartheta}(\mu)$. Again, by Theorem 2, $\|x_\mu - x_w\|$ can be made arbitrarily small by reducing $\vartheta > 0$ in order to avoid integration failures. Therefore, the loopings within step 6 will terminate at a finite number of loops as well.

The frequency of executing step 6 depends on the size of ϵ we choose. The larger it is, the more frequent the execute of step 6 becomes. From experience, there is hardly any execution of step 6 of Algorithm 1 in the whole process for some arbitrarily chosen small ϵ. The reasons for avoiding the execution of step 6 are: (i) it is computationally cumbersome, and (ii) any execution of this may increase the chances of finding a local minimum.

There is another way to avoid the execution of step 6 and at the same time avoiding infinite loopings of the Algorithm. Note that the optimization process $M(\cdot, \cdot, \cdot \mid \mathscr{W})$ is an iterating process by itself. Let $w \in \mathscr{W}$ denotes its search direction in each iteration and let $\sigma \in \mathbb{R}$ denotes the corresponding step size. Thus, if u is the control used in the previous iteration, $u := u + \sigma w$ is the control used for the current iteration. Let $\bar{\sigma}$ denotes the bounds on σ, that is, $|\sigma| \leq \bar{\sigma}$. Now, replace step 5 and step 6 by the following single step:

```
step 5:  If $\eta > \epsilon$, goto step 4
         otherwise, set $\bar{\sigma} := \frac{\bar{\sigma}}{2}$, goto step 4.
```

In addition, $\bar{\sigma}$ is reset to its original value every time step 2 is executed.

Note that infinite loopings can be avoided, again by Theorem 2. However, this modified algorithm is not recommended, as: (i) it is still possible to find a local minimum, (ii) it usually takes a longer computational time, and (iii) changing bounds on step size is usually not a built-in user option in optimal control software.

With Algorithm 1, we can now present the proposed BM method as follows.

ALGORITHM 2.

```
step 0:  Set $i = 0$, and fix the parametrization of the control.
step 1: Set $\xi^0 = [-\pi/2, 0, 0, \dots, 0]^{\mathrm{T}}$, and $\mu^0 = 0$ for all $t \in [0, T)$.
        Apply Algorithm 1.
step 2: Let $\mu^{i+1} = M(\xi^i, \mu^i, 2)$.
step 3: Let $\alpha = 1$, and let $\Delta$ be any non-zero $n$ dimensional vector small in magnitude
        that makes $|\xi^i + \alpha\Delta - x(0)| < |\xi^i - x(0)|$.
step 4: Let $\xi^{i+1} = \xi^i + \alpha\Delta$.
step 5 : If $M(\xi^{i+1}, \mu^{i+1}, 1) = 1$, goto step 6;
        otherwise, set $\alpha := \frac{\alpha}{2}$, goto step 4.
step 6: If $\xi^{i+1} = x_0$, then stop;
        otherwise, set $i := i + 1$, goto step 2.
```

Our required u^0 is given by μ^{i+1} upon the termination of the algorithm.

The remaining question is on the existence of such a non-zero term $\alpha\Delta$ in step 3 of the algorithm. The answer to this question follows readily from the continuous dependency property of system trajectory on initial conditions discussed in Section 3. To be more specific, let $\epsilon > 0$ represent the amount of trajectory shift that the differential equation solver can tolerate before encountering integration failures. Then for a fixed μ^{i+1}, there exists a $\delta > 0$ such that

$$|M(\xi^{i+1}, \mu^{i+1}, 3) - M(\xi^i, \mu^{i+1}, 3)| \le \epsilon$$

whenever $|\xi^{i+1} - \xi^i| \le \delta$. Hence such a non-zero $\alpha\Delta$ exists as long as $\epsilon > 0$. However, if there is no tolerance at all, ($\epsilon = 0$), the BM method will fail to work. In fact, nothing will work for such problems that do not allow any room for tolerance. As far as multi-link vertical planar robot arm systems are concerned, this does not happen.

It is interesting to note that gravity is, in fact, the cause of chaotic behaviour in the robot arm systems. However, a stable equilibrium point is known to be at $\xi^0 = [\pi/2, 0, 0, \dots, 0]^{\mathrm{T}}$, and it can be readily used as a starting point for the BM method.

Note that in step 3, the choice of Δ is infinite. Any Δ that is small enough in magnitude such that $|\xi^i + \alpha\Delta - x(0)| < |\xi^0 - x(0)|$ is a valid choice. In all iteration, if we choose Δ pointing in the same direction as $x(0) - [-\pi/2, 0, 0, \dots, 0]^{\mathrm{T}}$, this means ξ^i gets closer to $x(0)$ in a straight line when i increases. However, from experience, this choice of Δ is likely to end up with a lot of looping between step 4 and step 5. Although, there is no other natural restrictions for us to follow, if $x(0)_j = 0$ for some $j \in \{2, 3, \dots, N\}$, usually, we choose the corresponding $\Delta_j = 0$. Also, Δ is recommended to have only one non-zero element. That is to say, ξ^i gets closer to $x(0)$ in a zig-zag path with each segments parallel to one of the axes of $\mathbb{R}^n$. From experience, this scheme of choosing Δ is more natural to the user and loopings between step 4 and step 5 are less.

5. Illustrative example

Consider the vertical planar two-link robot arm system with gravity taken into account. The system of state differential equations is as stated in Section 2. The physical parameters are set as follows in their appropriate units:

$$l_1 = 0.45, \quad l_{c1} = 0.225, \quad m_1 = 10, \quad I_1 = 0.4,$$
$$l_2 = 0.25, \quad l_{c2} = 0.125, \quad m_2 = 5, \quad I_2 = 0.1956$$

and the gravitational acceleration $g = 9.81$.

The control bounds are $|u_1(t)| \leq 20$ and $|u_2(t)| \leq 10$.

Our aim is to move the robot arms from the initial configuration

$$x(0) = [x_1(0), x_2(0), x_3(0), x_4(0)]^{\mathrm{T}} = [-70, 0, 7, 0]$$

such that the following cost functional is minimized:

$$\begin{aligned} J(u) =& 10[[x_1(10) - 80]^2 + [x_2(10)]^2 + [x_3(10) - 80]^2 + [x_4(10)]^2 \\ &+ \int_0^{10} \frac{[x_1(t) - 80]^2 + [x_2(t)]^2 + [x_3(t) - 80]^2 + [x_4(t)]^2}{100} \\ &\qquad + \frac{[u_1(t)]^2 + [u_2(t)]^2}{1000}\, dt, \end{aligned}$$

where $x_1(0) = -70$, $x_2(0) = 0$, $x_3(0) = 7$, $x_4(0) = 0$.

This cost functional attempts to get the robot arms from the initial configuration into the configuration $x = [80, 0, 80, 0]^{\mathrm{T}}$ smoothly with a reasonably small amount of control (torque) in 10 units of time. The term

$$\frac{[x_1(t) - 80]^2 + [x_2(t)]^2 + [x_3(t) - 80]^2 + [x_4(t)]^2}{100}$$

in the integrand is to penalize the far positioning of the arms and its spinning and big swing motions, while $([u_1(t)]^2 + [u_2(t)]^2/1000$ is to penalize the amount of torque used.

Note that if the time horizon of the problem were just 1 unit of time instead of 10, there would not be much numerical difficulty for any initial guess of the control function. This is due to the fact that the chaotic behaviour developed in such a small time interval is not significant enough to cause integration failures.

We make use of the software package MISER3.1 [3] for the $M(\xi^i, \mu^i, \cdot)$ computations to obtain μ^i, and to perform the control parametrization refinements for u^j

thereafter. The software package MISER3.1 requires analytical partial derivatives of the right hand side of (2.1) w.r.t. the state and control. This could be a tedious job, especially when the dimensions of the system is high, or the state differential equation is complicated. However, with the automatic differentiation program PADRE2 [4] incorporated with MISER3.1, it is a simple task.

Since the required $x_2(0)$ and $x_4(0)$ are zeros, that is, the system should start from rest, ξ_2^i and ξ_4^i are continued to be zeros for all iterations i. The control is chosen from the class of piecewise constant functions on 4 equal partitions.

The iterates of ξ_1^i are shown in Figure 2, whereas the iterates of ξ_3^i are shown in Figure 3. Note that the increase of both ξ_1^i and ξ_3^i are slow when i is ranging from 4 to 50. Certainly, there is some numerical difficulties when ξ^i is around $[-74, 0, 3, 0]^{\mathrm{T}}$. The system becomes "very chaotic" in this region. Numerical integration within MISER3.1 can tolerate only a very slight change in the initial condition. After passing that region, both ξ_1^i and ξ_3^i move quickly to $\xi^i = [-70, 0, 7, 0]^{\mathrm{T}}$. The algorithm terminated at the 73rd iteration. The plots of μ^1, and μ^{73} are shown in Figures 4 and 5 respectively, and their associated costs are shown in Table 1. Note that both μ^1 and μ^{73} hit the bounds. From the look of μ^{73} itself, it is hard to guess an initial control "close enough" to it to avoid numerical integration failures.

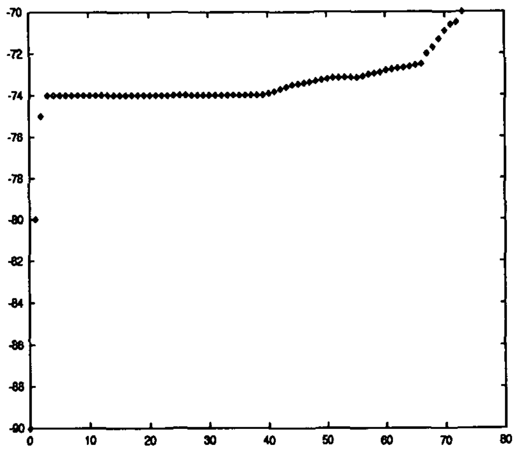

FIGURE 2. The plot of ξ_1^i. Rapid increases at the beginning and at the end.

Now let the time horizon be partitioned into 63 equal subintervals. Then, by using $u^0 = \mu^{73}$ as a workable initial solution, MISER3.1 converged successfully. The corresponding optimal state trajectory and the controls u_1 and u_2 are shown in Figures 6, 7 and 8 respectively. Note that x_1 and x_3 behave like a turnpike solution. It swings with a small amplitude until close to the end of the time horizon, where it moves to the desired state. The turnpike behaviour also appears in u_1 and u_2 where most of

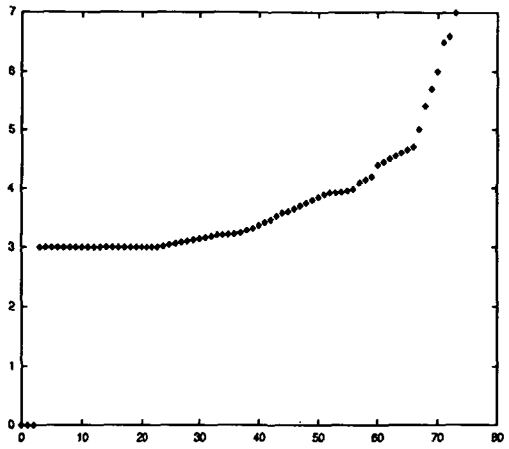

FIGURE 3. The plot of ξ_3^i. Rapid increases at the beginning and at the end.

the control effort are spent close to the end. Moreover, it is interesting to note that, although μ^{73} hit the bounds, the optimal control u_1 and u_2 are inactive with respect to the bounds.

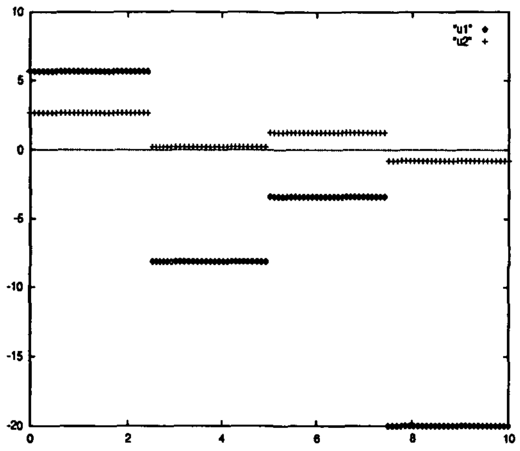

FIGURE 4. The plot of $\mu_1^1(t)$ and $\mu_2^1(t)$. Control hits the bounds.

6. Conclusions

We consider an optimal control problem involving a vertical planar multi-link robot arm system moving under the effect of gravity. For such an optimal control problem, it is not an easy task to guess an initial control function such that the numerical integration of the system would not fail. Thus, none of the gradient

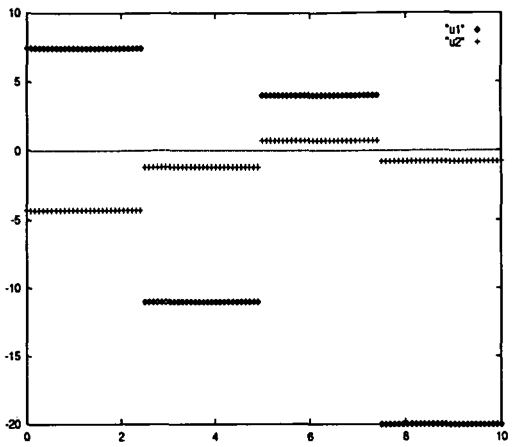

FIGURE 5. The plot of $\mu_1^{73}(t)$ and $\mu_2^{73}(t)$. Control hits the bounds.

based optimal control algorithms, such as the control parametrization algorithm, will work in this situation. In this paper, we have developed a simple and intuitive method for constructing a workable initial guess for the control function of the control parametrization algorithm. This method is referred to as the Blind Man (BM) method. It is developed based on the properties of continuous dependence of solutions on initial conditions and on inputs (controls). These properties are well known and elementary in the theory of differential equation. Theoretical justification is provided and an illustrative example is given to demonstrate the usefulness of the method.

TABLE 1. The table lists the costs associated with various iterations and the optimal cost (with number of partitions set to 63).

$i = 1$	46.933
$i = 6$	47.263
$i = 16$	47.262
$i = 73$	47.194
optimal cost	1.547

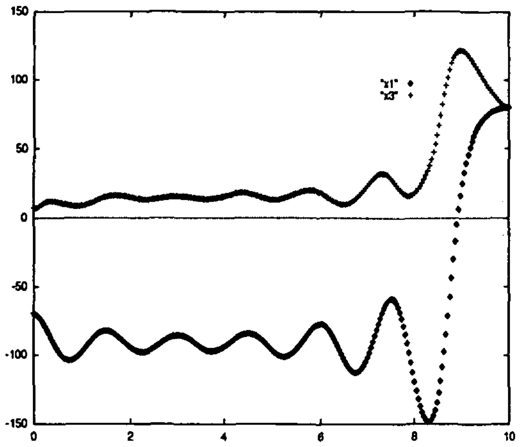

FIGURE 6. The plot of $x_1(t)$ and $x_3(t)$ after MISER3.1 converged, with u set as a 63 equally partitioned piecewise constant function. It exhibits a turnpike-like behaviour.

Appendix

Optimal control and control parametrization Consider a process described by the following system on nonlinear differential equations on $(0, T]$:

$$\dot{x} = g(t, x, u), \tag{A.1.a}$$

where $x \in \mathbb{R}^n$ is the state vector and $u : [0, T) \mapsto U \subset \mathbb{R}^r$ is the control vector. The set U defines the control bounds and it is some compact subset of $\mathbb{R}^r$. Let $\mathscr{U}$ be the class of all such bounded measurable functions u. Elements of $\mathscr{U}$ are called admissible controls, and $\mathscr{U}$ is called the class of admissible controls. The function $g : [0, T] \times \mathbb{R}^n \times \mathbb{R}^r \mapsto \mathbb{R}^n$ is given a nonlinear function which is continuously differentiable with respect to all its arguments. The initial condition for the system of differential equations (A.1.a) is

$$x(0) = x_0. \tag{A.1.b}$$

The standard optimal control problem is then formulated as:

Subject to the dynamical systems (A.1), find an admissible control $u \in \mathscr{U}$ such that the cost functional

$$J_0(u) = \Phi_0(x(T)) + \int_0^T \mathscr{L}_0(t, x, u)\, dt \tag{A.2}$$

is minimised over $\mathscr{U}$, and such that for $i = 1, 2, \dots, M_e$, the inequality constraints

$$J_i(u) = \Phi_i(x(T)) + \int_0^{\tau_i} \mathscr{L}_i(t, x, u)\, dt \geq 0 \tag{A.3}$$

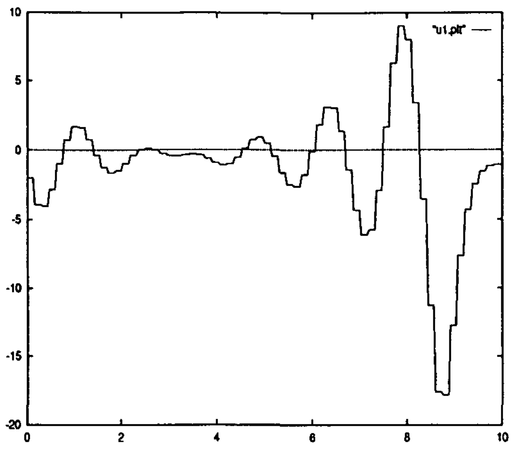

FIGURE 7. The plot of $u_1^*(t)$. Turnpike behaviour and within the bounds.

and for $i = M_e + 1, M_e + 2, \dots, M_e + M_i$, the equality constraints

$$J_i(u) = \Phi_i(x(T)) + \int_0^{\tau_i} \mathscr{L}_i(t, x, u)\, dt = 0 \tag{A.4}$$

are satisfied. For $i = 0, 1, 2, \dots, M_e + M_i$, Φ_i and $\mathscr{L}_i$ are given functions which are continuously differentiable with respect to all their arguments.

In the control parametrization method, the control function u is approximated as a linear combination of some basis functions partitioning the time horizon $[0, T]$. For example, if these basis functions are zero order splines, this approximate representation turns out to be a piecewise constant function. The coefficients of these basis functions will completely characterize the control function within this particular representation. Hence, finding an optimal control within this representation is to minimize (A.2) with respect to the corresponding coefficients. It is, therefore, an optimal parameter selection problem, which can be viewed as a mathematical programming problem. The first outer iteration of the method is said to be complete, after the optimal coefficients are obtained.

Then, a refinement of the partition of the time horizon is taken. This refinement is taken in such a way that it contains the previous partition, resulting in a larger set of coefficients. We can now start the second outer iteration, where, the previously calculated coefficients are used in the initial guess for the new set of coefficients. Optimization is performed to obtain the new set of optimizing coefficients. The second outer iteration is said to be complete after new optimal coefficients are obtained.

This refining and the subsequent optimizing process are repeated until a certain stopping criteria is satisfied.

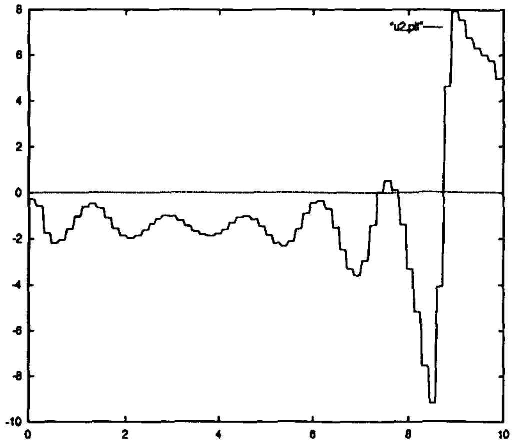

FIGURE 8. The plot of $u_2^*(t)$. Turnpike behaviour and within the bounds.

Note that from the second and other subsequent outer iterations, initial guesses are obtained from the results of the previous iteration. However, how do we obtain an initial guess of the first outer iteration remains a question.

For more details on control parametrization methods see [2, 13].

For a vertical planar multi-link robot arms system moving under the effect of gravity, its chaotic behaviour associated with a non-workable initial guess will cause numerical integration failures. Thus, our aim in this paper is to develop a systematic approach to construct an initial guess, which is workable, for the optimal control algorithms, in particular, control parametrization algorithms.

Acknowledgements

The authors are grateful to Dr Bruce D. Craven for his most valuable suggestions in improving this work.

This paper is partially supported by a research grant fro the Australian Research Council.

References

[1] N. U. Ahmed, "Elements of finite-dimensional systems and control theory", *Pitman Monographs and Surveys in Pure and Applied Mathematics* 37 (Longman Sciences and Technical, 1988).

[2] B. D. Craven, *Control and optimization* (Chapman and Hall, 1995).

[3] L. S. Jennings, M. E. Fisher, K. L. Teo and C. J. Goh, "MISER3 Optimal Control Software" : *Theory and User Manual* (EMCOSS Pty Ltd, 1990).

[4] Koichi Kubota and Masao Iri, *PADRE2, version 1 – User's Manual Research Memorandum* (RMI 90-01, 1990).

[5] H. W. J. Lee, K. L. Teo and V. Rehbock, "Sub-optimal local feedback control for a class of nonlinear control problems", *Dynamics of Continuous, Discrete and Impulsive systems* **1** (1995).

[6] H. W. J. Lee, K. L. Teo and V. Rehbock, "Sub-optimal local feedback control for a class of constrained discrete time nonlinear control problems", *Computers and Mathematics and Applications*, to appear.

[7] John J. Murray, *Computational robot dynamics*, Ph. D. Thesis, Carnegie Mellon University, 1986.

[8] A. Miele, "Gradient Algorithms for the Optimization of Dynamic Systems", in: *Control and Dynamic Systems Advances in Theory and Applications* 16 (ed. C. T. leondes) (Academic Press, New York, 1980) 1–52.

[9] E. Polak and D. Q. Mayne, "A feasible directions algorithm for optimal control problems with control and terminal inequality constraints", *IEEE Trans. Aut. Control* **AC-22** (1977) 741–751.

[10] Y. Sakawa, "On local convergence of an algorithm for optimal control", *Numerical Funct. Anal. Opt.* **3** (1981) 301–319.

[11] H. R. Sirisena and F. S. Chou, "Convergence of the control parametrization ritz method for nonlinear optimal control problems", *J. Optimaization Theory and Applications* (November 1979).

[12] Jan M. Skowronski, *Control Theory of Robotic Systems* (World Scientific, 1989).

[13] K. L. Teo, C. J. Goh and K. H. Wong, *A Unified Computational Approach to Optimal Control Problems* (Longman Sciences and Technical, 1991).