ANZIAM J. 47(2006), 541–554

COMPUTATIONAL MODELS SATISFYING RELATIVE ANGLE CONSTRAINTS FOR 2-DIMENSIONAL SEGMENTED BODIES

S. MUNZIR✉ [1], L. S. JENNINGS[2] and M. T. KOH[3]

(Received 23 November, 2004; revised 6 February 2006)

Abstract

Planar hinged segmented bodies have been used to represent models of biomechanical systems. One characteristic of a segmented body moving under gravitational acceleration and torques between segments is the possibility that the body's segments spin through more than a revolution or past a natural limit, and a computational mechanism to stop such behaviour should be provided. This could be done by introducing angle constraints between segments, and computational models utilising optimal control are studied here. Three models to maintain angle constraints between segments are proposed and compared. These models are: all-time angle constraints, a restoring torque in the state equations and an exponential penalty model. The models are applied to a 2-D three-segment body to test the behaviour of each model when optimising torques to minimise an objective. The optimisation is run to find torques so that the end effector of the body follows the trajectory of a half-circle. The result shows the behaviour of each model in maintaining the angle constraints. The all-time constraints case exhibits a behaviour of not allowing torques (at a solution) which makes segments move past the constraints, while the other two show a flexibility in handling the angle constraints which is more similar to what occurs in a real biomechanical system.

2000 *Mathematics subject classification*: primary 92C10; secondary 49N15.
Keywords and phrases: biomechanics, optimal control, all-time constraints.

1. Introduction

The dynamic behaviour of multi-degree-of-freedom biomechanical systems is usually studied by computer simulation of their dynamic equations [1]. Planar hinged

[1] School of Mathematics and Statistics, The University of Western Australia, 35 Stirling Hwy, Crawley, WA 6009, Australia; e-mail: msaid@maths.uwa.edu.au.

[2] School of Mathematics and Statistics, The University of Western Australia, 35 Stirling Hwy, Crawley, WA 6009, Australia.

[3] Physical Education and Sports Science Group, National Institute of Education, 1 Nanyang Walk, Singapore.

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press

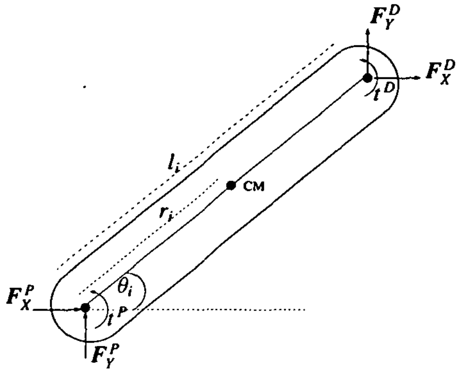

FIGURE 1. Body diagram of the i^{th} segment.

segmented body models have been used for analysis of biomechanical systems [6–8]. This is due to the need for simplified biomechanical models that represent a particular system accurately and which are capable of providing simulation results which satisfy the requirement for that system. This type of model has been used not only for dynamics simulation and analysis [3, 13] but also for control purposes, as shown in [10]. Therefore, studies in this area will provide advancement in the field of biomechanics and sport sciences.

Computational control of segmented body models under the effect of gravity has been consistently studied by many researchers, especially in the area of robotics [2]. However, certain improvements and modifications are needed since there are wide areas of application of this type of system with recent developments in robotics as well as in biomechanics. One of the most important characteristics of segmented rigid body models moving under the effect of gravity is the possibility that a body's segments spin past a natural limit and a computational mechanism to stop such behaviour should be provided.

In practice, for instance in robotics, this behaviour is simply stopped by introducing hard constraints to stop the segment from moving beyond its angle limit. Parallels can be drawn with what occurs in a real biomechanical system where the muscle and bone structure of a human body prevent its segments from moving past natural limits. However, hard constraint bounds are not fit for biomechanical systems, because large forces imposed at certain joints of the human body can cause injury. As a result, it is necessary to introduce different types of mechanisms into the computational model of this control problem, so that appropriate control signals (torques) can be generated to suit the corresponding real system. In this study, three different types of models to control angle constraints between segments are introduced. The first uses a restoring torque in the dynamic equations when the segments move past their constraints, the second uses an exponential penalty in the objective function, and the

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press

third uses additional all-time constraints.

Consider a 2-D body made up of n segments, each of which is a straight line, joined by revolute joints. The i^{th} segment has length ℓ_i, mass m_i and moment of inertia I_i about its centre of mass (CM), which is a distance r_i from its proximal end. Each segment position is known from one fixed point on the segment (for example, the proximal end) and the angle θ_i that the segment makes with the positive X-axis (Figure 1).

Using this geometric frame, the augmented equation of motion for 2-D segmented bodies resulting from the translational and rotational equations of motion is given by

$$\begin{bmatrix} \boldsymbol{J} & \boldsymbol{F}^X & \boldsymbol{F}^Y \\ \boldsymbol{J}^Y & \boldsymbol{S} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A} & \boldsymbol{0} \\ -\boldsymbol{J}^X & \boldsymbol{0} & \boldsymbol{S} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{f}^X \\ \boldsymbol{f}^Y \end{bmatrix} = \begin{bmatrix} \boldsymbol{T\tau} \\ -\boldsymbol{J}^X\boldsymbol{\omega}^2 \\ \boldsymbol{0} \\ \boldsymbol{m}g - \boldsymbol{J}^Y\boldsymbol{\omega}^2 \\ \boldsymbol{0} \end{bmatrix}. \tag{1.1}$$

The first row in Equation (1.1) is obtained from the rotational equations of motion, while the next two rows are horizontal translational equations and balances of horizontal forces respectively, and the last two rows are similar equations for the vertical direction. In the equation, $\boldsymbol{J}$ is the diagonal matrix of moments of inertia of the segments w.r.t. their centre of mass, $\boldsymbol{\alpha}$ is the angular acceleration vector, $\boldsymbol{\tau}$ is the vector of torques at the joints, $\boldsymbol{T}$ is the matrix with entries 1 or -1 depending on the influence of each torque on each segment, $\boldsymbol{\omega}$ is the angular speed vector, $\boldsymbol{m}$ contains the masses of the segments, g is gravitational acceleration, $\boldsymbol{f}^X$ consists of forces in the horizontal direction, $\boldsymbol{f}^Y$ consists of forces in the vertical direction, and $\boldsymbol{A}$ is a matrix with entries 1 and -1 reflecting the direction of translational forces.

The matrices $\boldsymbol{F}^X$, $\boldsymbol{F}^Y$, $\boldsymbol{J}^X$, $\boldsymbol{J}^Y$ and $\boldsymbol{S}$ are given by

$$\begin{aligned} \boldsymbol{F}^X &= \boldsymbol{D}_s\begin{bmatrix} -\boldsymbol{D}_r & \boldsymbol{0} & \boldsymbol{D}_l - \boldsymbol{D}_r \end{bmatrix}, & \boldsymbol{J}^X &= \boldsymbol{D}_m\boldsymbol{L}\boldsymbol{D}_c, & \boldsymbol{S} &= \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{m} & \boldsymbol{I} \end{bmatrix}, \\ \boldsymbol{F}^Y &= \boldsymbol{D}_c\begin{bmatrix} \boldsymbol{D}_r & \boldsymbol{0} & -(\boldsymbol{D}_l - \boldsymbol{D}_r) \end{bmatrix}, & \boldsymbol{J}^Y &= \boldsymbol{D}_m\boldsymbol{L}\boldsymbol{D}_s, \end{aligned}$$

where $\boldsymbol{D}_r = \operatorname{diag}(r_1, \dots, r_n)$, $\boldsymbol{D}_c = \operatorname{diag}(\cos\theta_1, \dots, \cos\theta_n)$, $\boldsymbol{D}_l = \operatorname{diag}(l_1, \dots, l_n)$, $\boldsymbol{D}_s = \operatorname{diag}(\sin\theta_1, \dots, \sin\theta_n)$, $\boldsymbol{D}_m = \operatorname{diag}(m_1, \dots, m_n)$, and $\boldsymbol{L}$ is given by

$$\boldsymbol{L} = \begin{bmatrix} r_1 & 0 & 0 & 0 & \dots & 0 \\ l_1 & r_2 & 0 & 0 & \dots & 0 \\ l_1 & l_2 & r_3 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ l_1 & l_2 & l_3 & l_4 & \dots & r_n \end{bmatrix}.$$

The complete derivation of the model is to be found in [5].

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press

The model of 2-D segmented bodies could be classified into two cases: the case of one contact at the proximal end of segment one and the case of free flight. For the case of one contact at the proximal end of segment one, using the special structure of matrices $\boldsymbol{S}$ and $\boldsymbol{A}$, the model can be simplified as

$$\boldsymbol{Q}(\boldsymbol{\theta})\boldsymbol{\alpha} = \boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{\tau}).$$

This simplified model is particularly of interest in this study, since it is readily used for the remainder of the paper.

2. Models to maintain angle constraints between segments

The purpose of this study is to explore models for the optimal control of 2-dimensional segmented bodies that maintain angle constraints between their segments while moving on a desired trajectory. Preserving angle constraints between segments is aimed especially at preventing a segment of the body from spinning around its transverse axis as the result of fast movement of its adjacent segments. (In computational models, large torques applied to heavier segments can make the lighter (foot or hand) segments spin.) In addition, it is also useful for determining a set of initial guesses of the control parameters for a multi-link robot, where convergence to the solution can only be achieved with a set of initial control parameters that are relatively close to the solution. Inverse analysis is usually used to obtain the initial torque estimates as shown in [5]. A particular study has also been conducted in [9] where an intuitive "Blind Man" algorithm was proposed for searching for a workable initial guess of control to avoid numerical integration failures, which usually occur due to extreme unrealistic velocities of lighter segments.

Taking z as the system parameters, the optimal control statement for this problem can be written in the following form:

$$\underset{\boldsymbol{\tau},z}{\text{minimise}}: \tilde{G}(\boldsymbol{\tau}, z) = \Phi_0(\boldsymbol{\theta}(T), \boldsymbol{\omega}(T), z) + \int_0^T \mathscr{L}(\boldsymbol{\theta}(t), \boldsymbol{\omega}(t), \boldsymbol{\tau}(t), z)\, dt$$

subject to the equations of motion for n-segmented bodies in the state space form

$$\dot{\boldsymbol{\theta}} = \boldsymbol{\omega}, \qquad \boldsymbol{Q}(\boldsymbol{\theta})\dot{\boldsymbol{\omega}} = \boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{\tau}, z),$$

with initial conditions $\boldsymbol{\theta}(0) = \boldsymbol{\theta}^0$ and $\boldsymbol{\omega}(0) = \boldsymbol{\omega}^0$. This is subject to the path constraints $\tilde{\boldsymbol{F}}(\boldsymbol{\theta}, t) = 0$, with constraints on the angle between segment i and $i - 1$

$$\zeta_1 \leq \boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1} \leq \zeta_2, \quad \text{for some } i \in \{2, \dots, n\}. \tag{2.1}$$

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press

Obviously, this problem could be solved through discretisation of state and control variables, resulting in a static optimisation problem in the form of finite difference equations. However, since extra constraints are necessary to represent the restriction of the angles between segments, this may produce an overdetermined system where the number of active constraints is larger than the number of variables near the solution. Another possible approach is to replace (2.1) by setting the lower and upper control bounds dependent on the state of the system. This approach can be implemented in the framework of full discretisation of state and control variables, as extra constraints. Again this may also produce an overdetermined system. Alternatively, this problem could be solved using control parametrisation approaches where optimal control problems are transformed into optimal parameter selection problems and gradients of the objective function as well as constraints w.r.t. the parameters are computed using costate variables. The latter approach is used here, since it has been successfully used in many previous works [6–8] and is proven to work very efficiently. The software for this method (MISER3.3 [4]) is also available and details of its mathematical background are given in [11].

Three different strategies for maintaining a small range of angles between segments are proposed. The first strategy is to introduce all-time constraints restricting the angle between segments in addition to the existing path constraints. Secondly, a type of restoring torque is introduced inside the state equation representing the dynamics of the segmented bodies. Finally, a penalty similar to a restoring torque is added to the objective function where its value increases very rapidly if the angle between segments increases beyond its limits.

2.1. All-time angle constraints The angle constraint between segment i and $i-1$,

$$\zeta_1 \leq \theta_i - \theta_{i-1} \leq \zeta_2, \tag{2.2}$$

is converted into the following two inequality constraints:

$$h_1(\boldsymbol{\theta}) = \theta_i - \theta_{i-1} - \zeta_1 \geq 0, \tag{2.3}$$

$$h_2(\boldsymbol{\theta}) = \theta_{i-1} - \theta_i + \zeta_2 \geq 0. \tag{2.4}$$

These constraints are then transformed into the standard canonical constraints to be added to the original standard optimal control problem. A method for solving these types of inequality constraints is readily available in MISER3.3 with a smoothing technique that provides a more accurate and stable computational result [12].

2.2. Restoring torque in the state equation The characteristics of the restoring torque are designed such that if the bounds on the relative angles are exceeded, the restoring torque will push the segments to restore the constraints. If $\theta_i - \theta_{i-1} > \zeta_2$,

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press

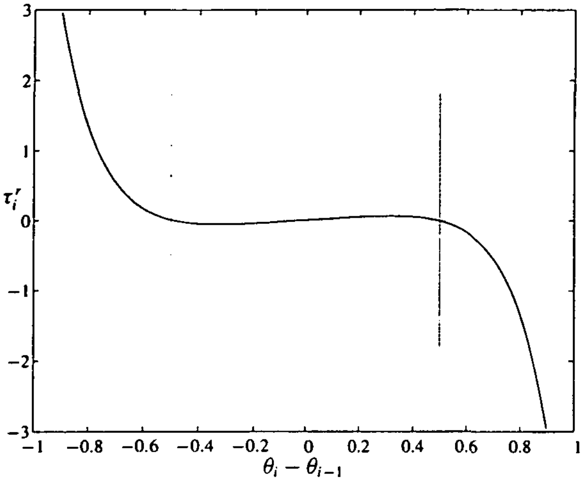

FIGURE 2. Graph of τ_i^r for $\zeta_1 = -0.5$, $\zeta_2 = 0.5$ and $\mu = 5$.

then the restoring torque forces segment i to decrease velocity and at the same time forces segment $i - 1$ to increase velocity to keep the angle difference less than ζ_2. On the other hand, if $\theta_i - \theta_{i-1} < \zeta_1$, then the restoring torque forces segment i to increase velocity and at the same time forces segment $i-1$ to decrease velocity to keep the angle difference greater than ζ_1. Such a design needs a relatively large amount of torque as soon as the differences between θ_i and θ_{i-1} hit the bounds. Here, the restoring torque is modelled using an exponential function, since this function increases very rapidly for positive arguments. The restoring torque for segment i is modelled as follows:

$$\tau_i^r = -h_1(\boldsymbol{\theta})e^{-\mu h_1(\boldsymbol{\theta})} + h_2(\boldsymbol{\theta})e^{-\mu h_2(\boldsymbol{\theta})}, \tag{2.5}$$

where μ is a relatively large positive value. If h_1 is negative (infeasible) the restoring torque on segment i is positive, while if h_2 is negative the restoring torque is negative. When h_1 and h_2 are positive the restoring torques are small and of opposite sign. It is assumed that μ is chosen large enough to make these small torques inconsequential compared to the $\boldsymbol{\tau}$ of the system. The restoring torque on segment $i - 1$ is opposite to that on segment i. A typical graph of τ_i^r and how it works on segment i are shown in Figures 2 and 3. Note that τ^r is zero at ζ_1 and ζ_2 and has slope -1 at these two points. Hence multiples of this torque or multiples of the two individual components could also be used, depending on how quickly larger restoring torques should come into action. The values of ζ_1 and ζ_2 can also be adjusted to be slightly 'inside' the real life limits.

As a result, when this one restoring torque is imposed on the model, the state

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press

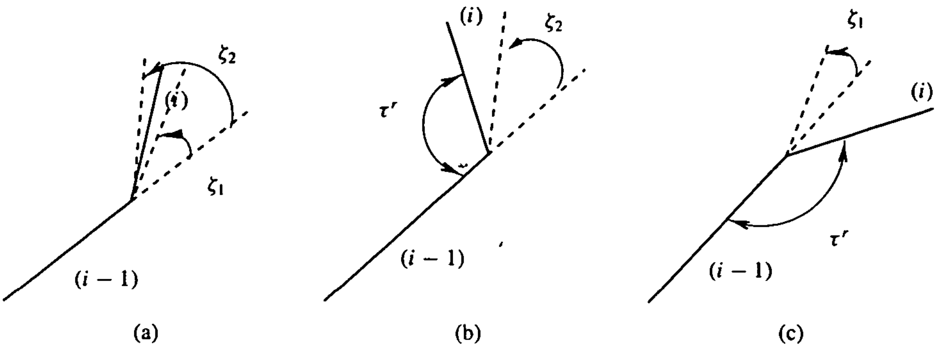

FIGURE 3. How τ^r works on segments i and $i-1$: (a) Lower and upper bound, (b) Exceed upper bound, (c) Exceed lower bound.

equations are typically changed to:

$$\begin{pmatrix} \dot{\theta} \\ Q(\theta)\dot{\omega} \end{pmatrix} = \begin{pmatrix} \omega \\ f_1(t,\theta,\omega,\tau,z) \\ \vdots \\ f_{i-1}(t,\theta,\omega,\tau,z) - \tau_i^r \\ f_i(t,\theta,\omega,\tau,z) + \tau_i^r \\ \vdots \\ f_n(t,\theta,\omega,\tau,z) \end{pmatrix}. \tag{2.6}$$

A state equation with all restoring torques replaces the original state equation, which is then combined together with the objective function and path constraints to form an optimal control problem.

To prevent the restoring torques from significantly distorting the optimal actual torques, the resulting optimal states in the optimisation should not, at any time interval, violate the bounds of the angle constraints. If the resulting optimal states violate the angle constraints, then the restoring torques should be added to the optimal torques to obtain the actual torques working on the corresponding joint. In this case, the solution may not be accepted as a viable solution.

2.3. Exponential penalty in the objective function With the exponential penalty, the segment is allowed to move freely according to τ. However, a penalty in the objective function is imposed if the difference between angles of segment i and $i-1$ is not in the interval (ζ_1, ζ_2). Therefore the penalty function is quite similar to the restoring torque, with the following form:

$$P = -h_1(\theta)e^{-\mu h_1(\theta)} - h_2(\theta)e^{-\mu h_2(\theta)}. \tag{2.7}$$

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press

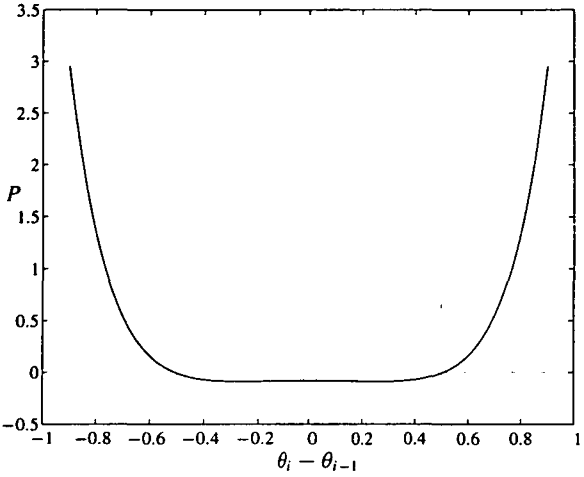

FIGURE 4. Graph of exponential penalty for $\zeta_1 = -0.5$, $\zeta_2 = 0.5$ and $\mu = 5$.

If $h_1 < 0$ then the first term of the penalty function produces a big positive value in the objective function, which through the optimisation algorithm will be pushed back to stay within the limit. On the other hand, if $h_2 < 0$ then the second term of the penalty function produces a big positive value in the objective function, which through the optimisation algorithm will be pushed back to stay within the limit. A typical graph of the exponential penalty function is shown in Figure 4.

The 'shallow' form of the graph near the upper and lower bound of the angle constraint is important, since the optimisation software is expected to converge quickly once the angle constraint is achieved. Although the exponential penalty function appears to have two local minima, the function is very flat in between the lower and upper bounds of the angle constraint. Therefore, it can only dominate the overall objective if the optimal control objective is very flat. If this is the case, almost any point in the flat region is a good enough solution.

3. Results and discussion

Each of the approaches is applied to a three-segment body of lengths 4, 2 and 1 metres with a mass of 6, 3 and 1.5 kgs respectively. The algebraic path constraint is the trajectory of a half-circle centred at (5, 0) with a radius of 1 m, starting from rest at a point close to (5, 1) moving in a counter-clockwise direction to move to the point (5, −1). The end effector of the body is required to move along the trajectory in 1 second. Hence, it is expected that the optimal control software will generate a large torque to the first two segments, which in turn will cause the adjacent segment, which

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press

is smaller in mass, to move extremely quickly and hit the angle constraint between segments. Here, the model is specifically intended to prevent the last segment from spinning, with the angle constraint $|\theta_3 - \theta_2| \le 0.5$. The optimal control problem is run using MISER3.3, where the algebraic path constraint is treated as integral constraints by introducing additional state variables. The additional state variables are formulated as

$$\dot{x}_a = (x_3(t) - x_{\text{traj}}(t))^2, \quad x_a(0) = 0,$$
$$\dot{x}_b = (y_3(t) - y_{\text{traj}}(t))^2, \quad x_b(0) = 0,$$

where $(x_3(t), y_3(t))$ are the coordinates of the distal end of segment three, while $(x_{\text{traj}}(t), y_{\text{traj}}(t))$ is the planned trajectory defined as:

$$(x_{\text{traj}}(t), y_{\text{traj}}(t)) = (5 - \sin(\pi t), \cos(\pi t)).$$

The additional state variables are combined with terminal state constraints $x_a(1) = 0$ and $x_b(1) = 0$ to replace the path constraint.

The objective of the optimal control is to minimise

$$\tilde{G}(\tau, z) = \int_0^1 \left((\tau_1)^2 + (\tau_2)^2 + (\tau_3)^2\right) dt.$$

The initial state position values $\theta(0)$, were allowed free (and hence were parameters of the optimisation) while the initial angular velocities were set at zero. Second-order regularisation is added to the objective as a penalty for deviation of torques from a straight line. This tends to give a smooth torque in time, instead of a stop-start torque.

Different methods applied in the treatment of angle constraint between segments will cause an additional computation for each iteration during optimisation, compared to the original problem. The software consists of two main computations, that is, computation of states, costates, objective, constraints and the gradient of the objective and constraints in one group and optimisation routine computation in another group. The computation of the optimisation routine depends on the number of control parameters and constraints, therefore only the all-time constraints method will change the amount of computation in the optimisation routine per iteration, since it introduces additional constraints.

In the first group of computation in the software, the number of computation for each iteration changes for all methods. For the restoring torque, the change occurs in the computation of the state and costate equations with the additional term in the state equation, and this causes a significant increase in the amount of computation. For the all-time constraints, the computation change is more significant than the restoring torque case, since extra computation is needed for the gradient of the extra constraints and their costate variables as well. For the penalty method, the difference only

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press

TABLE 1. Violation of angle constraint. Abbreviation used: pwc: Piecewise constant control; pwl: Piecewise linear control; pwc reg: Piecewise constant control with regularisation; pwl reg: Piecewise linear control with regularisation.

Methods	Type of Control	No. Times Violated	Maximum Violation	Average Violation
All-time Constraints	pwc	2	0.0377	0.0238
	pwc reg	9	0.0097	0.0058
	pwl	7	0.1542	0.0888
	pwl reg	17	0.0264	0.0169
Restoring Torque	pwc	85	0.6260	0.3968
	pwc reg	96	0.9035	0.3802
	pwl	102	0.8432	0.6496
	pwl reg	91	0.9327	0.3609
Penalty	pwc	30	0.2854	0.0485
	pwc reg	51	0.2227	0.1076
	pwl	41	0.3306	0.1019
	pwl reg	57	0.2399	0.1193

appears in the computation of the costate of the objective which can be considered as an insignificant change. Finally, the amount of computation for solving the state and costate equations also depends on the stiffness of the corresponding ode, where a stiff ode is more difficult and takes a longer time to integrate. In this study, more frequent stiffness occurred in the restoring torque method, as expected.

One of the most important issues in this simulation is how well each model keeps the relative angle inside the interval of $[-0.5, 0.5]$. This result could be used as a hint as to which type of application each of the approaches is suitable for. From the result in Table 1, it is very obvious that the all-time constraints approach is the best in stopping the segments from moving beyond their bounds. From 130 quadrature points, the segments violate their angle constraint only a few times and only a relatively small distance. The final torques generated here are such that the segments do not 'crash' into their bounds. The restoring torque approach is quite flexible in holding the angle constraint inside the interval. This method allows the system to violate the angle constraint very frequently with a relatively large distance from its bound. This characteristic suits certain biomechanics systems in the human body in which flexibility is needed when large forces are applied to the system due to external contacts. It can indicate dangerous types of 'optimal' movement where joints are put at risk of injury. The exponential penalty method behaves moderately compared to the other two, which is good for the less flexible biomechanical system. There is flexibility built into the restoring torque and penalty approach in that parameters μ, ζ and an overall scale parameter can be used to adjust the violations of any solution. For static optimisation, too large a value of μ in any penalty method may cause ill-

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press

conditioning for the Hessian matrix of the penalty objective which in turn may make the optimisation routine unstable. Some eigenvalues of the Hessian are large like μ, while others are small. The exponential penalty here does not contain the control parameters directly, so the effect of μ may be spread over all eigenvalues of the Hessian when integration in the objective is taken into account. The big μ value may contribute to the costate ode integration failure due to the increasing of its stiffness. Similarly for the restoring torque method, if the value of μ is too big, then for some parameter values in the line searches both the state and costate ode may become very stiff and impossible to integrate.

During optimisation, the penalty approach allows the segment to exceed the angle bound since this is a very common behaviour of penalty methods. The small segment may spin for a particular setting of parameter values during optimisation and integration failure may occur if the optimisation is permitted to try inappropriate torque values in its line search. Therefore, upper and lower bounds of control parameters for this method have to be set properly. The final solution may also exceed the bound by a small quantity. The restoring torque method allows the body to generate torques which do care for the angle limits and does prevent the smaller segments from spinning whenever the optimisation tries an evaluation with inappropriate torque values in its line search. The behaviour of the all-time constraint method during optimisation depends on the optimisation algorithm used within the software. If the optimisation algorithm is of a type which begins with an infeasible point, then the small segments may spin and integration failure may occur without a proper lower and upper bound of the control parameters. On the other hand, if the optimisation algorithm is of a type which maintains feasibility during optimisation, then the optimisation may prevent small segments from spinning.

Other common issues explored in this study are accuracy in following the trajectory and time per iteration needed in the optimal control software to obtain a fairly accurate final set of control parameters. These results are summarised in Table 2, which shows the average amount of time needed for one iteration of the optimal control software. Accuracy is defined as $x_a(1) + x_b(1)$, which is obtained from the integral constraints. As appears in the table, without regularisation, the amount of time for the exponential penalty method is relatively small compared to the other two methods and this is consistent with the previous analysis on the amount of computation. The restoring torque method requires more time than the all-time constraints approach, although it needs less computation. This is due to the frequent stiffness which occurred in the state and costate equations resulting from this approach. Introducing regularisation also appears to make the states and costates ODEs more difficult to integrate since it increases the amount of time for all methods.

From these results, the restoring torque method in general appears to be the best in following the designed trajectory of the segmented bodies. This advantage of the

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press

TABLE 2. Time per iteration and accuracy. Abbreviation used: pwc: Piecewise constant control; pwl: Piecewise linear control; pwc reg: Piecewise constant control with regularisation; pwl reg: Piecewise linear control with regularisation.

Methods	Type of Control	Time (s)	Accuracy
All-time Constraints	pwc	0.382	7.3×10^{-4}
	pwc reg	2.824	1.24×10^{-3}
	pwl	0.241	9.3×10^{-4}
	pwl reg	2.233	1.05×10^{-3}
Restoring Torque	pwc	0.466	6.2×10^{-4}
	pwc reg	1.058	8.4×10^{-4}
	pwl	0.345	3.16×10^{-3}
	pwl reg	1.577	6.5×10^{-4}
Penalty	pwc	0.126	2.21×10^{-3}
	pwc reg	0.306	1.74×10^{-3}
	pwl	0.187	2.09×10^{-3}
	pwl reg	2.655	4.15×10^{-3}

method is also complemented by a smaller number of iterations (304 iterations on average) needed to complete the optimal control computation. On the other hand, introducing all-time constraints gives a less accurate result in following the trajectory compared to the restoring torque. This weakness is also followed with a larger number of iterations (1125 iterations on average) in completion of the optimal control computation. In the exponential penalty method, a less accurate path in following the designed trajectory is found, however this is a result of a much smaller number of iterations (247 iterations on average) required to complete the optimisation.

Sample plots of the actual and desired trajectory of the segmented bodies for each method are shown in Figures 5 and 6. Sources of inaccuracy in the all-time constraints method appear at the beginning and the end of motion, while in between it is fair in following the designed trajectory. On the other hand, the restoring torque approach does not result in much violation near the end of the motion. This is possibly because it allows more flexibility for the segment to move. Similar behaviour is also found in the exponential penalty method with much more inaccuracy shown for a longer time at the beginning of the motion.

4. Conclusion

Three models to maintain angle constraints between segments in the optimal control of a segmented body are proposed. These models are: restoring torque in the state equations, all-time constraints and an exponential penalty model. The models are applied to a 2-D three-segment body to exhibit the behaviour of each model. Simulations

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press

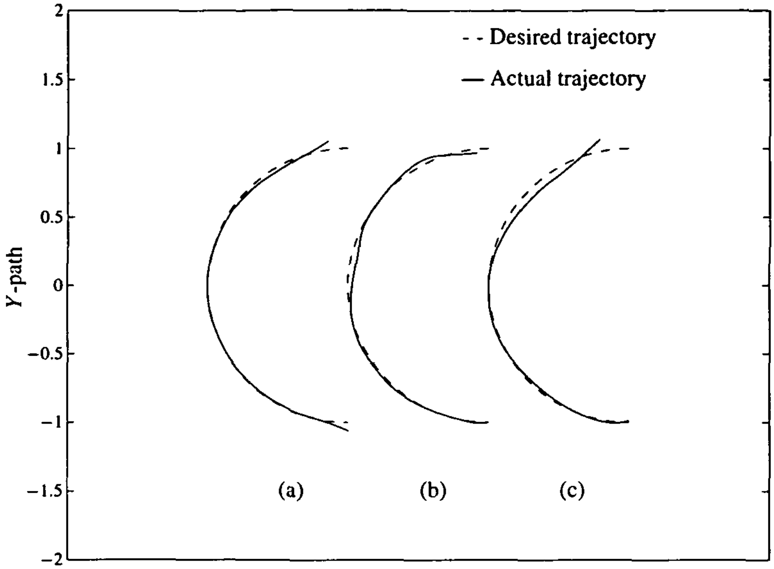

FIGURE 5. Actual and desired path of the body using piecewise constant control: (a) All-time constraints, (b) Restoring torque and (c) Penalty method.

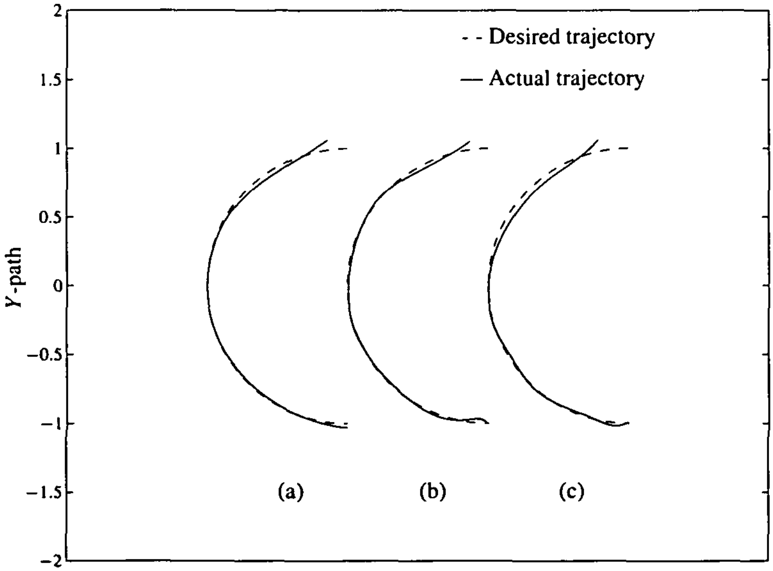

FIGURE 6. Actual and desired robot path of the body using piecewise linear control: (a) All-time constraints, (b) Restoring torque and (c) Penalty method.

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press

are run where the end effector of the manipulator is designed to follow the trajectory of a half-circle. The results show the behaviour of each model in maintaining the angle constraints, which can be concluded as:

(1) The all-time constraints model exhibits a behaviour of computing torques so as not to 'crash' into constraints. Hence it is the best method to prevent the violation of angle constraints.

(2) The restoring torque method allows more violation of angle constraints, however it follows the path constraint more accurately compared to other methods as it treats the angle constraints inside the state equations.

(3) The exponential penalty is the least favourable method both for violation of angle constraints and for not achieving the most accurate optimum result.

References

[1] S. K. Agrawal, "Inertia matrix singularity of planar series-chain manipulators", in *Proceedings of the 1991 IEEE conference on robotics and automation, Sacramento, California*, Vol. 1 (IEEE Computer Society, Los Alamos, CA, 1991), pp. 102–107.

[2] F. Alonge, F. D'Ippolito and F. M. Raimondi, "An adaptive control law for robotic manipulator without velocity feedback", *Control Eng. Prac.* **11** (2003) 999–1005.

[3] C. Azevedo, N. Andreff and S. Arias, "Bipedal walking: from gait design to experimental analysis", *Mechatronics* **14** (2004) 639–665.

[4] L. S. Jennings, M. E. Fisher, K. L. Teo and C. J. Goh, *MISER3: Optimal Control Software, Theory and User Manual, Version 3* (http://www.maths.uwa.edu.au/u/les, 2004).

[5] M. T. Koh and L. S. Jennings, "Dynamic optimization: A solution to the inverse dynamic problem of biomechanics using MISER3", *Dyn. Contin. Discrete Impulsv. Syst. Ser. B Appl. Algorithms* **9B** (2002) 369–386.

[6] M. T. Koh and L. S. Jennings, "Dynamic optimization: inverse analysis for the Yurchenko layout vault in women's artistic gymnastics", *J. Biomech.* **36** (2003) 1177–1183.

[7] M. T. Koh, L. S. Jennings and B. Elliott, "Role of joint torques generated in an optimized Yurchenko layout vault", *Sports Biomech.* **2** (2003) 177–190.

[8] M. T. Koh, L. S. Jennings, B. Elliott and D. Lloyd, "A predicted optimal performance of the Yurchenko layout vault in women's artistic gymnastics", *J. Appl. Biomech.* **19** (2003) 187–204.

[9] H. W. J. Lee, K. L. Teo and L. S. Jennings, "An optimal control of multi-link vertical planar robot arms systems moving under the effect of gravity", *J. Austral. Math. Soc. Ser. B* **39** (1997) 195–213.

[10] L. L. Menegaldo, A. D. T. Fleury and H. I. Weber, "Biomechanical modeling and optimal control of human posture", *J. Biomech.* **36** (2003) 1701–1712.

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

[12] K. L. Teo and L. S. Jennings, "Nonlinear optimal control problems with continuous state inequality constraints", *J. Optim. Theory Appl.* **63** (2003) 1–22.

[13] L. Tian and L. G. Gilbertson, "The study of control methods for the robotic testing system for human musculoskeletal joints", *Computer Meth. Prog. Bio.* **74** (2004) 211–220.

https://doi.org/10.1017/S1446181100010129 Published online by Cambridge University Press