

ON THE SIMULATION AND CONTROL OF SOME FRICTION CONSTRAINED MOTIONS ^{*}

ROLAND GLOWINSKI [†] AND ANTHONY J. KEARSLEY [‡]

Abstract. In this paper, some issues involved with numerical simulation and control of some elasto-dynamic systems are discussed. The motivation is the simulation of dry or *Coulomb* friction in the joints that link together remote manipulator systems used in aerospace operations (for example, space shuttle remote manipulator systems). The goal here is to develop numerical techniques to simulate and control these systems, while properly modeling the Coulomb friction. The numerical procedure described employs a finite difference time discretization in conjunction with a vector of multipliers that predicts the friction effect for all time. In addition to this discrete multiplier technique an associated regularization procedure that greatly improves the behavior of these multipliers is also presented. Numerical examples conclude the paper.

Key Words. Coulomb friction, direct search methods, nonsmooth optimization

AMS(MOS) subject classification. 49J15, 70Q05

Abbreviated title. Simulation and control of friction constrained motion
sectIntroduction

In this paper, we discuss the simulation and control of some elasto-dynamic systems with dry friction. The phenomenon of dry or *Coulomb* friction has been described and analyzed in [KikO88] and in [CamOK82] (see also [Ren92]). In [Cab81], Coulomb friction is analyzed in the motion of a string. These methods all approach these physical problems as time dependent variational inequalities (see [DuvL76]). The spatial semi-discretization of these problems gives rise to systems like the one examined here.

We consider the simple time dependent problem

$$(0.1) \quad M\ddot{x} + Ax + C\lambda = f, \quad t \in (0, T],$$

$$(0.2) \quad x(0) = x_0, \quad \dot{x}(0) = x_1,$$

$$(0.3) \quad \lambda_i(t) = 0 \text{ if } c_{ii} = 0, |\lambda_i(t)| \leq 1 \text{ if } c_{ii} > 0, \text{ and } C\lambda(t) \cdot \dot{x}(t) = \sum_{i=1}^d c_{ii} |\dot{x}_i(t)|.$$

We are using the standard inner product of \mathfrak{R}^d , i.e. for $y, z \in \mathfrak{R}^d$ we have $y \cdot z = \sum_{i=1}^d y_i z_i$. In this system, $x_i(t)$ denotes the displacement of the i th component at time t . The mass matrix $M \in \mathfrak{R}^{d \times d}$ is a symmetric and positive definite matrix. The stiffness matrix, $A \in \mathfrak{R}^{d \times d}$ is symmetric and positive semi-definite. The friction matrix $C \in \mathfrak{R}^{d \times d}$ is diagonal with only non-negative diagonal entries. Matrices M , A and C are all assumed to be constant with respect to time t . The vectors $x, \dot{x}, \ddot{x} \in \mathfrak{R}^d$ represent displacement, velocity, and acceleration, respectively. Here $\dot{x} = \frac{dx}{dt}$ and $\ddot{x} = \frac{d^2x}{dt^2}$. Also, $x_0, x_1 \in \mathfrak{R}^d$ are the initial displacement and velocity, respectively. The function f models the external forces acting on the system, and we will assume that $f \in L^2(0, T; \mathfrak{R}^d)$. Finally, $\lambda \in L^\infty(0, T; \mathfrak{R}^d)$ is the normalized coefficient of Coulomb friction.

^{*} Use of the Intel iPSC/860 was provided by the Center for Research on Parallel Computation under NSF Cooperative Agreement Nos. CCR-9120008 and CDA-8619893 with support from the Keck Foundation.

[†] Department of Mathematics, University of Houston, 4800 Calhoun Rd., Houston, TX 77204-3476

[‡] Department of Computational & Applied Mathematics, P. O. Box 1892, Rice University, Houston, Texas 77251-1892

The system described in (0.1) – (0.3) is equivalent to a second order time dependent variational inequality whose solution requires finding a function $x \in H^2(0, T; \mathfrak{R}^d)$ such that

$$(0.4) \quad \int_0^T (M\ddot{x} + Ax - f) \cdot (y - \dot{x})dt + \int_0^T (j(y) - j(\dot{x}))dt \geq 0, \quad \forall y \in L^2(0, T; \mathfrak{R}^d),$$

$$(0.5) \quad x(0) = x_0, \quad \dot{x}(0) = x_1.$$

Here we have

$$(0.6) \quad j(y) = \sum_{i=1}^d c_{ii}|y_i|, \quad \forall y \in \mathfrak{R}^d.$$

The space $H^2(0, T; \mathfrak{R}^d)$ is a Hilbert space with associated inner product

$$(0.7) \quad \langle y, z \rangle = \sum_{j=0}^2 \int_0^T \frac{d^j y}{dt^j} \cdot \frac{d^j z}{dt^j} dt.$$

The functional $j(\cdot)$ is convex and continuous. It is also non-smooth, unless $c_{ii} = 0$, for all $i = 1, 2, \dots, d$.

Existence and uniqueness for the solution of (0.1) – (0.3) can be proved using the methods from [DuvL76]. We will not prove these results here.

1. Simulation. Let $N, \Delta t, T$ be positive scalars denoting the number of time steps in the discretized problem, the length of the time step, and the final time, respectively. Here we have $\Delta t = \frac{T}{N}$. We seek an approximate solution to problem (0.1) – (0.3), say, x^n for every $n\Delta t$. Similarly, we approximate $f(n\Delta t)$ by f^n . In this article, we will assume that f is sufficiently smooth ($f \in C^0([0, T]; \mathfrak{R}^d)$) so that we can take $f^n = f(n\Delta t)$. We then approximate system (0.1) – (0.3) by

$$(1.1) \quad M \frac{1}{\Delta t^2} (x^{n+1} + x^{n-1} - 2x^n) + A(\alpha x^{n+1} + \alpha x^{n-1} + (1 - 2\alpha)x^n) + C\lambda^n = f^n, \quad \forall n \geq 0;$$

$$(1.2) \quad \lambda_i^n = 0 \quad \text{if} \quad c_{ii} = 0,$$

$$(1.3) \quad |\lambda_i^n| \leq 1 \quad \text{and} \quad \lambda_i^n (x_i^{n+1} - x_i^{n-1}) = |x_i^{n+1} - x_i^{n-1}| \quad \text{if} \quad c_{ii} > 0,$$

$$(1.4) \quad x^0 = x_0 \quad \text{and} \quad x^1 - x^{-1} = 2x_1 \Delta t,$$

$$(1.5) \quad \lambda_i^0 = 0 \quad \text{if either} \quad c_{ii} = 0 \quad \text{or} \quad \dot{x}_i^0(0) = 0 \quad \text{and} \quad c_{ii} > 0,$$

$$(1.6) \quad \lambda_i^0 = \text{sign}(\dot{x}_i(0)) \quad \text{if} \quad \dot{x}_i(0) \neq 0 \quad \text{and} \quad c_{ii} > 0.$$

Discretization (1.1) – (1.6) has been investigated in [DeaGKN90] for $\alpha \in [0, \frac{1}{2}]$ and stability properties of the scheme were established. Some convergence properties were also established, such as:

THEOREM 1.1. [Dean, Glowinski, Kuo, Nasser]. Suppose (x, λ) is the solution of (1.1)–(1.6). Let $\lambda_{\Delta t}$ be the function of t , defined by

$$(1.7) \quad \text{label2} - 7\lambda_{\Delta t}(t) = \lambda^0 \quad \text{when} \quad t \in [0, \frac{\Delta t}{2})$$

$$(1.8) \quad \lambda_{\Delta t}(t) = 0 \quad \text{when} \quad t \in (T - \frac{\Delta t}{2}, T]$$

$$(1.9) \quad \lambda_{\Delta t}(t) = \lambda^n \quad \text{when} \quad t \in (n\Delta t - \frac{\Delta t}{2}, n\Delta t + \frac{\Delta t}{2}).$$

Then (1.1)–(1.6) is an unconditionally stable scheme if $\alpha \in [\frac{1}{4}, \frac{1}{2}]$ and the following convergence results hold:

1. $\lim_{\Delta t \rightarrow 0} \max_{0 \leq n \leq N} \|x^n - x(n\Delta t)\| = 0$,
2. $\lim_{\Delta t \rightarrow 0} \max_{0 \leq n \leq N-1} \|\frac{1}{\Delta t}(x^{n+1} - x^n) - \dot{x}((n + \frac{1}{2})\Delta t)\| = 0$,
3. $\lim_{\Delta t \rightarrow 0} \lambda_{\Delta t} = \lambda$, weakly * in $L^\infty(0, T; \mathfrak{R}^d)$.

Remark 2.1: If $\alpha \in [0, \frac{1}{4})$ the above convergence results still hold, the stability condition in that case being

$$\Delta t < [(\frac{1}{4} - \alpha)\delta_M]^{-\frac{1}{2}}$$

with δ_M the largest eigenvalue of the matrix $M^{-1}A$.

Following [DeaGKN92], we can solve the nonlinear system (1.1)–(1.6) by observing that the conditions on λ in (1.1)–(1.6) can be rewritten as

$$(1.10) \quad \lambda^n = P_\Lambda \left(\lambda^n + \frac{r}{2\Delta t} C(x^{n+1} - x^{n-1}) \right), \quad \forall r \geq 0, \quad \forall n \geq 0.$$

The projector $P_\Lambda : \mathfrak{R}^d \rightarrow \Lambda$, is the orthogonal projection operator from \mathfrak{R}^d onto the set Λ , where Λ is a closed convex set defined by

$$(1.11) \quad \Lambda = \{\lambda : \lambda \in \mathfrak{R}^d, |\lambda_i| \leq 1 \text{ and } \lambda_i = 0 \text{ if } c_{ii} = 0\}.$$

We have for all $\mu \in \mathfrak{R}^d$, $P_\Lambda(\mu) = z$ with $z_i = 0$ if $c_{ii} = 0$ and $z_i = \min\{1, \max(\mu_i, -1)\}$ if $c_{ii} > 0$.

We can numerically solve the class of problems defined by (1.1)–(1.6) using the techniques developed and described in [Glo84].

In the particular case of problem (1.1) –(1.6) and for $\alpha = \frac{1}{4}$, these techniques yield the following algorithm for computing the pair (x^{n+1}, λ^n) :

$$(1.12) \quad \lambda^{n,0} \in \Lambda;$$

for $k \geq 0$, $\lambda^{n,k}$ being known in Λ , solve the linear system

$$(1.13) \quad (M + \frac{\Delta t^2}{4}A)x^{n+1,k} = \Delta t^2(f^n - C\lambda^{n,k}) + M(2x^n - x^{n-1}) - \frac{\Delta t^2}{4}A(2x^n + x^{n-1})$$

and update $\lambda^{n,k}$ by

$$(1.14) \quad \lambda^{n,k+1} = P_\Lambda \left(\lambda^{n,k} + \frac{r}{2\Delta t} C(x^{n+1,k} - x^{n-1}) \right).$$

We stop iterating when

$$(1.15) \quad \frac{\|x^{n+1,k} - x^{n+1,k-1}\|}{\|x^{n+1,k-1}\| + 1} \leq \eta$$

for some well chosen norm $\|\cdot\|$ and parameter η .

When applying algorithm (1.12) – (1.15) to the solution of system (1.1) – (1.6), for some test problem (0.1) – (0.3), the numerical experiments show very good convergence properties for the approximate displacement and velocity (see, for example, Figures 1, 2, 5 and 6). However, the approximate multiplier $\lambda_{\Delta t}$ displays violent oscillations (as shown in Figures 3 and 7); naturally, this behavior is compatible with the convergence Theorem 2.1 which only guarantees weak convergence for $\lambda_{\Delta t}$. Some applications, however, require more accurate discrete multipliers since, after all, these multipliers measure friction forces. It is for this reason that a regularization procedure has been developed in [DeaGKN92]. This procedure is based on observing first that equation (1.14) is a discrete form of

$$(1.16) \quad \lambda = P_\Lambda (\lambda + rC\dot{x}),$$

then by regularizing (1.16) via

$$(1.17) \quad \epsilon \dot{\lambda} + \lambda = P_{\Lambda}(\lambda + rC\dot{x}),$$

with $\epsilon > 0$, and finally by discretizing (1.17) by

$$(1.18) \quad \epsilon \left(\frac{\lambda^n - \lambda^{n-1}}{\Delta t} \right) + \lambda^n = P_{\Lambda} \left(\lambda^n + rC \frac{x^{n+1} - x^{n-1}}{2\Delta t} \right).$$

This procedure can be viewed as a dynamical Tychonoff regularization procedure. It is quite clear that relation (1.17) implies that the regularized multiplier is a Lipschitz continuous function of t . It can be shown that if

$$\lim_{\Delta t \rightarrow 0} \frac{\epsilon}{\Delta t} = 0,$$

then the convergence results of Theorem 2.1 still hold.

In practice, inspired by (1.18), we shall use the regularized variant of algorithm (1.12) – (1.15) obtained by replacing (1.14) by

$$(1.19) \quad \lambda^{n,k+1} = \frac{\Delta t}{\epsilon + \Delta t} P_{\Lambda} \left(\lambda^{n,k} + \frac{r}{2\Delta t} C(x^{n+1,k} - x^{n-1,k}) \right) + \frac{\epsilon}{\epsilon + \Delta t} \lambda^{n-1};$$

the parameter ϵ has to be chosen so that $\lim_{\Delta t \rightarrow 0} \frac{\epsilon}{\Delta t} = 0$.

The above regularization procedure has a double effect:

- It substantially improves, for the same value of r , the convergence of algorithm (1.12) – (1.15).
- It improves the convergence of $\lambda_{\Delta t}$ to λ without affecting the convergence of the approximate displacement and velocity to their respective limit, this is seen in Table 5. Indeed from the figures and Table 5 we can conjecture that the weak-* convergence of the discrete multipliers has been replaced by a strong convergence of the discrete regularized multipliers $L^s(0, T; \mathfrak{R}^d)$ for all $s \in [1, +\infty)$. Proving this result would be of interest.

Let δ be the largest eigenvalue of the matrix $([M + \frac{\Delta t^2}{4}A]^{-1}C^2)$. Numerical observations suggest that a value of $r \approx \frac{2}{\delta \Delta t}$ yields the best results. While apparently this value of r is optimal, $r \in (0, \frac{4}{\delta \Delta t})$ is necessary to guarantee the important fixed point property that yields convergence of the pair $(x^{n+1,k}, \lambda^{n,k})$.

2. Control. There are many practical applications where we are interested in controlling the time evolution of the system modelled by (0.1) – (0.3). In particular we may require x and/or \dot{x} to have preassigned values at some final time $t = T$.

This is certainly applicable to the control of remote manipulator systems, where final velocities must be kept small. Here we choose target velocity \dot{x}_T , and target position x_T . The cost function employed will use penalty terms to attain target states. The resulting problem will look like

$$(2.1) \quad M\ddot{x} + Ax + C\lambda = f + Bv, \quad t \in (0, T],$$

$$(2.2) \quad x(0) = x_0, \quad \dot{x}(0) = x_1,$$

$$(2.3) \quad \lambda_i(t) = 0 \text{ if } c_{ii} = 0, \quad |\lambda_i(t)| \leq 1 \text{ if } c_{ii} > 0, \text{ and } C\lambda(t) \cdot \dot{x}(t) = \sum_{i=1}^d c_{ii} |\dot{x}_i(t)|$$

where we want to solve

$$(2.4) \quad \min_{v \in \mathfrak{R}^d} J(v) = \int_0^T Nv \cdot v dt$$

$$(2.5) \quad \text{subject to:} \\ x(T) = x_T$$

$$(2.6) \quad \dot{x}(T) = \dot{x}_T.$$

The vector, $v \in \mathbb{R}^p$, is the vector of control variables. The matrix $B \in \mathbb{R}^{d \times p}$ will dictate how the control is administered to the system. Likewise, the matrix $N \in \mathbb{R}^{p \times p}$, is a matrix that may be different from the identity if some parts of the system are more difficult to control than others. For simplicity we will assume that $p = d$ and that both B and N are multiples of the identity matrix. The algorithm used to solve this *nonsmooth* nonlinear programming problem (2.4) – (2.6) will be discussed in Section 4.

3. Optimization Technique. Problem (2.4) – (2.6) is an equality constrained nonlinear programming problem. The equality constraints are *not* differentiable with respect to the control variables, v , prohibiting the use of very fast algorithms that require smoothness of the constraints (e.g. see [GilMW81]). In fact, we expect that at the solution of (2.4) – (2.6), say v^* , the constraints will not be differentiable with respect to v . Whenever the velocity of any of the components of our dynamical system vanishes, the derivative with respect to v will not exist. For this reason we employed an algorithm that used no derivative information.

In [DenT91] an algorithm for unconstrained optimization that requires no derivative information was suggested. While algorithms for unconstrained minimization that required no derivative information (usually referred to as *direct search* methods) are not new (see, for example, [Cea71] and the references therein), a global first-order stationary point convergence theory for this particular method has only recently been developed in [Tor93]. An implementation of this Parallel Direct Search method (PDS) has been developed and tested (see [Tor92]). Recently, (PDS) has been modified to handle constraints [KeaTT93]. The method samples points on an evolving simplex, moving to the vertex on the simplex with value closest to optimality. The simplex then expands or contracts depending on where the point closest to optimality was located. The contractions reduce the size of the simplex; the procedure is halted when the lengths of the edges in the simplex fall below a tolerance set by the user. Direct search methods of this sort typically do not demonstrate rapid local convergence, but they are extremely robust and far less susceptible than faster higher-order methods to the difficulties introduced when functions are nonsmooth or the data is noisy.

Because we are dealing with only equality constraints, minimization problem (2.4) – (2.6) can be handled in a straightforward manner. After each simplex is constructed, a penalty function is evaluated. Let ρ_1 and ρ_2 be two large constants. Then we employ a penalty function, such as

$$(3.1) \quad P(v; \rho_1, \rho_2) = \int_0^T Nv \cdot v dt + \rho_1 \|x(T) - x_T\|_2 + \rho_2 \|\dot{x}(T) - \dot{x}_T\|_2.$$

We can now minimize this penalty function and the new problem is now a continuous, nonsmooth, nonconvex unconstrained optimization problem. Early numerical testing included trying various updating strategies for the penalty parameters ρ_1 and ρ_2 . Most of our attempts at dynamically adjusting these parameters for penalty function (3.1) failed to significantly improve the performance of our direct search method algorithm. The dynamic penalty parameter updating schemes,

$$(3.2) \quad \rho_1^+ \leftarrow \min\{10^6, (\rho_1^c + \|x(T) - x_T\|_\infty)\}$$

$$(3.3) \quad \rho_2^+ \leftarrow \min\{10^6, (\rho_2^c + \|\dot{x}(T) - \dot{x}_T\|_\infty)\}$$

for ρ_1 and ρ_2 were employed for comparison purposes only and remained fairly ineffective when compared to the performance with constant $\rho_1 = \rho_2 = 10^6$ (see Table 2 & 4). The superscript c denotes a quantity associated with the minimizer on the previous simplex.

More complicated penalty functions for (PDS) involving penalty parameter schemes, dynamical choices of norms for penalization, and exploiting feasibility are discussed in [KeaTT93]. One such suggestion is to use the penalty function,

$$(3.4) \quad \Pi_K(v; \rho_3) = \int_0^T Nv \cdot v dt + \rho_3(G_K(v)) + Z_K(v, v^c)$$

where $Z_K(v, v^c)$ is

$$(3.5) \quad Z_K(v, v^c) = \min \left\{ \frac{G_K(v^c)}{\max\{G_K(v), 10^{-8}\}}, 1 \right\} \max \left\{ \int_0^T N v^c \cdot v^c dt - \int_0^T N v \cdot v dt, 0 \right\}$$

and $G_K(v)$ is

$$(3.6) \quad G_K(v) = \|(x(T) - x_T, \dot{x}(T) - \dot{x}_T)^T\|_K.$$

Here v^c denotes the value of control variables yielding smallest penalty function value on the preceding simplex. The augmentation function $Z_K(v, v^c)$ incorporates a balance between decreasing infeasibility and moving towards optimality. The choice of norm, K ,

$$(3.7) \quad K = \begin{cases} \infty & \text{if } G_2(v^c) \geq 2 \\ 2 & \text{if } 1 < G_2(v^c) < 2 \\ 1 & \text{if } G_2(v^c) \leq 1 \end{cases}$$

depends on the distance from feasibility. The parameter ρ_3 can be chosen initially to be a large constant or updated dynamically. The updating strategy employed here combines (3.2) and (3.3)

$$(3.8) \quad \rho_3^+ \leftarrow \frac{1}{2}(\rho_1^+ + \rho_2^+).$$

It is worth commenting that consistent performance of (PDS) with respect to the penalty parameters ρ_1 , ρ_2 and ρ_3 when minimizing penalty functions (3.1) and (3.6) is characteristic of direct search methods. This is inherent in the way search directions and trial points are generated independently of any quantity dependent on the penalty parameters. Moreover, a new point is selected from the set of trial points by having the smallest associated objective function value. There are no minimum decrease or maximum allowable change conditions enforced. In this way, there is no ill-conditioning introduced into the problem by penalty parameters that are too large. The price paid for this robustness is, as mentioned above, no rapid local convergence.

4. Numerical Results. The test problems presented here are small ones, and are intended only to demonstrate the effectiveness of the simulation method and regularization. Actually, much larger problems have been solved using similar techniques, as shown for example in [DeaGKN90] where an algorithm like (1.12) – (1.15) has been used to simulate the motion of an elastic string in the presence of dry friction; employing the additional regularization procedure discussed here has been straight forward and the corresponding results will be given in a forthcoming article [GloK95].

In order to make our first test problem more significant, we have chosen for A the 3×3 Hilbert matrix because it is a small matrix with a large condition number (≈ 524) and more importantly it strongly couples the components of the state variable $x(t)$, implying a fairly complicated dynamics particularly for the friction multipliers. We have also chosen a final time, T , relatively small. For this problem we prescribed a desired final position, but no target velocity.

The second test problem can be physically motivated by considering a spring-mass system consisting of $(d+1)$ springs with known spring constants and d known masses. The springs are suspended between two fixed ends and connected by exactly one of the masses between every two springs. The spring constants determine the stiffness matrix A and the masses determine the matrix M .

We solve the initial value problem (2.1) – (2.3) to evaluate the objective function (2.4) and the constraints (2.5) and (2.6). The following numerical experiments demonstrate the behavior of the algorithm and specifically the multipliers, with and without the regularization. For simplicity we took $d = 3$, with the following values and functions:

Test problem (TP1)

- $x_0 = (1, 2, 3)^T$ (initial positions),
- $x_1 = (1, 1, 1)^T$ (initial velocities),

- $m_{11} = 4, m_{22} = 5, m_{33} = 6$ (diagonal matrix),
- $a_{ij} = \frac{1}{(i+j)-1}$ (Hilbert matrix),
- $c_{ii} = 10$ (diagonal matrix),
- $f_i(t) = -100e^{-10t}$ (external forces)
- $\Delta t = .0125$ (time step)
- $T = 5$ (final time)
- $\epsilon = (\Delta t)^{6/5}$,
- $r = \frac{2}{\delta \Delta t}$,
- $x_T = (2, 0, -2)^T$ (desired final position)
- $n_{ii} = \frac{1}{3}$ (diagonal matrix).
- $b_{ii} = 1$ (identity matrix)

Test problem (TP2)

- $x_0 = (1, 3, 5)^T$ (initial positions)
- $x_1 = (2, 2, 2)^T$ (initial velocities)
- $m_{11} = 1, m_{22} = 2, m_{33} = 3$ (diagonal matrix)
- $\kappa_1 = 1, \kappa_2 = 2, \kappa_3 = 3, \kappa_4 = 4$ (spring constants)
- $a_{11} = 3, a_{22} = 4, a_{33} = 5, a_{23} = a_{32} = -3, a_{12} = a_{21} = -2$ (Stiffness matrix)
- $c_{11} = 6, c_{22} = 5, c_{33} = 4$ (diagonal matrix)
- $f_i(t) = -10 \sin(\frac{3\pi t}{5}) e^{-t^2}$ (external forces)
- $\Delta t = .01$ (time step)
- $T = 10$ (final time)
- $\epsilon = (\Delta t)^{6/5}$
- $r = \frac{2}{\delta \Delta t}$
- $x_T = (5, 10, 15)^T$ (desired final position)
- $\dot{x}_T = (0, 0, 0)^T$ (desired final velocity)
- $n_{ii} = \frac{1}{3}$ (diagonal matrix)
- $b_{ii} = 1$ (identity matrix).

The control problems (3.1) and (3.6) were solved on an Intel iPSC/860, using 8 nodes simultaneously to evaluate 216 function evaluations per processor for each simplex generated. The minimization procedure was halted in the event that the largest edge of a generated simplex was less than 10^{-10} .

The simulation was performed on a Sun sparystation (1+), in double precision arithmetic (IEEE 64-bit floating point arithmetic). The fixed point iterations were applied until (using stopping criteria (1.15)) convergence was detected (with a value of $\eta = 1.5 \times 10^{-8} \approx \sqrt{\epsilon_{machine}}$ and the norm used was $\|\cdot\| = \|\cdot\|_2$).

The tables illustrate how effective the regularization procedure was. In the simulation, the number of fixed point iterations decreases drastically when the regularization is employed (see Table 2 & 4). While the regularization could not decrease the number of simplexes generated by (PDS) and hence, could not decrease the number of initial value problems (2.1) – (2.3) that needed to be solved, it did result in a decrease in the elapsed time. The dynamical updating of the penalty parameter had little influence on the performance of (PDS). Employing the more sophisticated penalty function (3.6) did consistently result in fewer (PDS) simplexes. The relative error in the final position, $E_\infty^1(v)$,

$$(4.1) \quad E_\infty^1(v) = \frac{\|x(T) - x_T\|_\infty}{\max\{1, \|x_T\|_\infty\}}$$

was recorded in the numerical results. The relative error associated with the velocities, $E_\infty^2(v)$,

$$(4.2) \quad E_\infty^2(v) = \frac{\|\dot{x}(T) - \dot{x}_T\|_\infty}{\max\{1, \|\dot{x}_T\|_\infty\}}$$

was not reported in the tables. For the first test problem (TP1) no final velocities were prescribed, and for the second problem the desired velocities ($\dot{x}_T = (0, 0, 0)^T$) were satisfied exactly.

The effect of regularizing on the positions and velocities of our test problems is summarized in Table 5. The maximum relative difference between regularized and unregularized position and velocity is recorded. More precisely, the discrepancies

$$D(x_R, x_{NR}) = \frac{\|\max_n |x_{NR}(n\Delta t) - x_R(n\Delta t)|\|}{1 + \max_n |x_R(n\Delta t)|} \quad \text{and} \quad D(\dot{x}_R, \dot{x}_{NR}) = \frac{\|\max_n |\dot{x}_{NR}(n\Delta t) - \dot{x}_R(n\Delta t)|\|}{1 + \max_n |\dot{x}_R(n\Delta t)|}$$

are in the third and fourth columns of Table 5, respectively. The subscripts R and NR denote regularized and nonregularized values respectively.

Position, velocity, and the Coulomb multipliers are plotted versus time. The first set of figures (Figures 1 – 4 and Figures 9 – 12) represents the system coming to rest without any control present and hence the velocities of all three components vanish quite rapidly. In the second set of figures (Figures 5 – 8 and Figures 13 – 16) the control is activated, and the final positions of the components are the desired targets.

A significant decrease in computation was seen when the regularization (using (1.19) instead of (1.14)) was employed. The oscillating behavior of the multipliers was virtually extinguished by the regularization procedure both in the controlled system and the uncontrolled system. An interesting observation is that the values of the regularized multipliers were always close to the time averaged values of the unregularized multipliers, as observed in [DeaGKN92]. The external forces were chosen to change the sign of the velocity in the interval $t \in (0, T]$. The only nonsmooth behavior in the multipliers occurred at the jumps, where the velocity of the system changed sign. Since we are most interested in behavior of these systems when velocities become small or vanish, dealing with the nonsmooth behavior of the multipliers was the primary difficulty in these simulations.

The graphs associated with test problem one and test problem two are labeled (TP1) and (TP2) respectively. Similarly, the graphs of the Coulomb multipliers calculated with the regularization procedure are labeled with (R) and those graphs calculated without regularization are labeled by (NR). In the uncontrolled system, the friction damped the displacements to zero very rapidly. Here the weak convergence of the multipliers is evident, and the benefit of the regularization is clear. In the controlled systems the displacements did reach the correct targets, both with and without regularization as seen in Figures 1, 5, 9 and 13. Again, we see the weak convergence of the multiplier corresponding to the one component whose velocity vanishes for t large enough.

The smooth behavior of the multipliers when velocities are far from zero and the effectiveness of the regularization suggest that a combination of smooth and nonsmooth minimizations may prove useful in the study of these motions. In particular, using smooth or higher order minimization techniques (like those in [DenS83]) for (3.1) or (3.6) when velocities are far from zero seems appropriate. When velocities are small, more robust methods like [DenT91] may yield minima of a non-smooth function that describes the system more accurately than a smooth analog. Perhaps the development of an algorithm for the minimization of (3.1) or (3.6) that could harness the strengths of both types of minimization techniques is possible.

Acknowledgments

The authors would like to thank the Texas Board of Higher Education (Grant # 003652156-ARP) and the Patricia R. Harris Fellowship at Rice University for very generous support. We must also thank the Center for Research on Parallel computation for the use of the Intel iPSC/860. This work was facilitated by numerous helpful discussions with Prof. E. J. Dean, Prof. J. E. Dennis, Dr. Y. M. Kuo, Dr. G. Nasser, Prof. R. A. Tapia and Dr. V. J. Torczon. We wish to thank Andrea Reiff for helpful suggestions regarding efficient use of (*PDS*). Finally we must also thank Prof. C. T. Kelley and two very thorough anonymous referees whose reports contributed greatly the presentation and substance of this paper.

Multiplier Scheme	Fixed Point Iterations (FIP)	Average: (FIP) per time step	Control Present on RHS?
Not Regularized	35633	89.0825	no
Not Regularized	16676	41.6900	yes
Regularized	5135	12.8375	no
Regularized	4184	10.4600	yes

TABLE 1
Problem 1: Simulation

Penalty Scheme	Penalty constant	Multipliers Scheme	PDS Simplexes	# of IVPs solved	error $E_{\infty}^1(v^*)$	elapsed time (msec)
$P(v; \rho)$	Constant	Not Regularized	21	36624	7.82E-8	1878366
$P(v; \rho)$	Constant	Regularized	21	36624	7.82E-8	1141314
$P(v; \rho)$	Dynamic	Not Regularized	21	36624	7.79E-8	1878338
$P(v; \rho)$	Dynamic	Regularized	21	36624	7.80E-8	1141328
$\Pi(v; \rho)$	Constant	Not Regularized	19	33136	2.01E-8	1707533
$\Pi(v; \rho)$	Constant	Regularized	19	33136	1.99E-8	1040543
$\Pi(v; \rho)$	Dynamic	Not Regularized	19	33136	2.08E-8	1707433
$\Pi(v; \rho)$	Dynamic	Regularized	19	33136	1.94E-8	1040461

TABLE 2
Problem 1: Control

Multiplier Scheme	Fixed Point Iterations (FIP)	Average: (FIP) per time step	Control Present on RHS?
Not Regularized	20876	20.876	no
Not Regularized	11517	11.517	yes
Regularized	3238	3.238	no
Regularized	3142	3.142	yes

TABLE 3
Problem 2: Simulation

Penalty Scheme	Penalty constant	Multipliers Scheme	PDS Simplexes	# of IVPs solved	error $E_{\infty}^1(v^*)$	elapsed time (msec)
$P(v, \rho)$	Constant	Not Regularized	23	40112	2.54E-9	2309235
$P(v, \rho)$	Constant	Regularized	23	40112	2.54E-9	1424193
$P(v, \rho)$	Dynamic	Not Regularized	23	40112	2.56E-9	2309188
$P(v, \rho)$	Dynamic	Regularized	23	40112	2.56E-9	1424085
$\Pi(v, \zeta)$	Constant	Not Regularized	20	34880	6.03E-8	2010565
$\Pi(v, \zeta)$	Constant	Regularized	20	34880	6.02E-8	1241696
$\Pi(v, \zeta)$	Dynamic	Not Regularized	20	34880	6.11E-8	2010213
$\Pi(v, \zeta)$	Dynamic	Regularized	20	34880	6.11E-8	1241428

TABLE 4
Problem 2: Control

Problem	Control present on RHS ?	position discrepancy $D(x_R, x_{NR})$	velocity discrepancy $D(\dot{x}_R, \dot{x}_{NR})$
Test Problem 1	yes	4.4E-8	2.8E-5
Test Problem 1	no	1.4E-7	3.8E-4
Test Problem 2	yes	6.1E-7	5.7E-2
Test Problem 2	no	8.9E-8	3.2E-3

TABLE 5
Difference between regularization and no regularization

Subject: dryfric.bbl

REFERENCES

- [Cab81] H. CABANNES, *Study of the motions of a vibrating string subject to solid friction*, Math. Meth. Appl. Sci., 3 (1981), pp. 287–300.
- [CamOK82] L. T. CAMPOS, J. T. ODEN, N. KIKUCHI, *A numerical analysis of a class of contact problems with friction in elastodynamics*, Comput. Meth. Appl. Mech. Engrg., 34 (1982), pp. 821–845.
- [Cea71] J. CÉA, *Optimisation : théorie et algorithmes*, Dunod, Paris, 1971.
- [DeaGKN90] E. J. DEAN, R. GLOWINSKI, Y. M. KUO, G. NASSER, *On the discretization of some second order in time differential equations. Applications to nonlinear wave problems.*, in Computational Techniques in Identification and Control of Flexible Flight Structures, A.V. Balakrishnan ed., *Optimization Software Inc.*, pp. 199-246, 1990
- [DeaGKN92] E. J. DEAN, R. GLOWINSKI, Y. M. KUO, G. NASSER, *Multiplier techniques for some dynamical systems with dry friction*, C. R. Acad. Sci. (Paris), 314 (1992), pp. 153–159.
- [DenS83] J. E. DENNIS, R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ., 1983.
- [DenT91] J. E. DENNIS, V. TORCZON, *Direct search methods on parallel machines*, SIAM Journal on Optimization, 1 (1991), pp. 448–474.
- [DuvL76] G. DUVAUT, J. L. LIONS, *Inequalities in Mechanics and Physics*, Springer-Verlag, Berlin, 1976.
- [Glo84] R. GLOWINSKI, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, 1984.
- [GloK95] R. GLOWINSKI, A. J. KEARSLEY, *Numerical control of a vibrating string subject to dry friction*, In preparation.
- [KeaTT93] A. J. KEARSLEY, R. A. TAPIA, V. TORCZON, *On the use of parallel direct search methods for nonlinear programming problems*, Technical Report #93-33, Department of Computational & Applied Mathematics, Rice University, Houston TX, 1993.
- [KikO88] N. KIKUCHI, J. T. ODEN, *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*, SIAM, Philadelphia, 1988.
- [Lio71] J. L. LIONS, *Optimal Control of Systems Governed by Partial Differential Equations*, Springer-Verlag, Berlin, 1971.
- [Lio88] J. L. LIONS, *Exact controllability, stabilization and perturbations for distributed systems*, SIAM Rev., 30 (1988), pp. 1–68.
- [GilMW81] P. E. GILL, W. MURRAY, AND M. H. WRIGHT, *Practical Optimization*. Academic Press, New York, 1981.
- [Ren92] M. RENARDY, *Ill-posedness at the boundary for elastic solids sliding under Coulomb friction*, Journal of Elasticity, 27 (1992), pp. 281–287.
- [Tor92] V. TORCZON, *PDS: Direct search methods for unconstrained optimization on either sequential or parallel machines*, Technical Report #92-9, Department of Computational & Applied Mathematics, Rice University, Houston TX, 1993. (submitted to ACM Transactions on Mathematical Software)
- [Tor93] V. TORCZON, *On the convergence of pattern search algorithms*, Technical Report #93-10, Department of Computational & Applied Mathematics, Rice University, Houston TX, 1993. (submitted to SIAM Journal on Optimization)

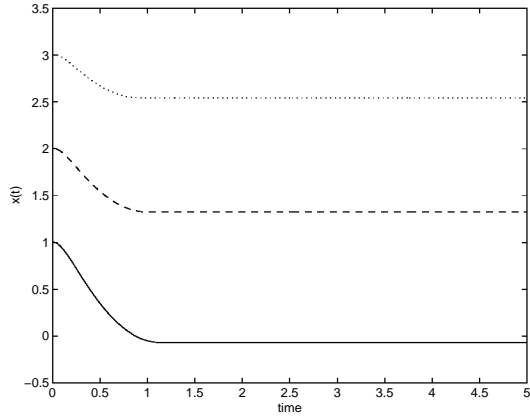


FIG. 1. - (TP1) - Displacements of uncontrolled system

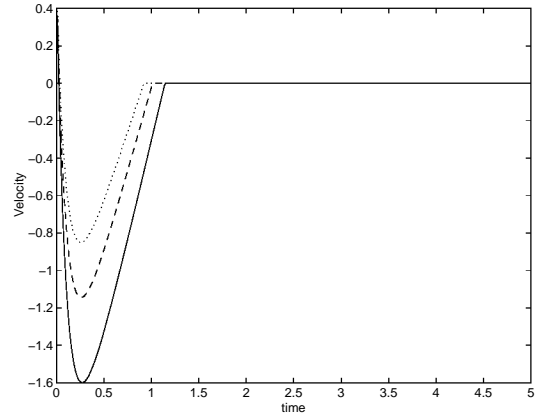


FIG. 2. - (TP1) - Velocities of uncontrolled system

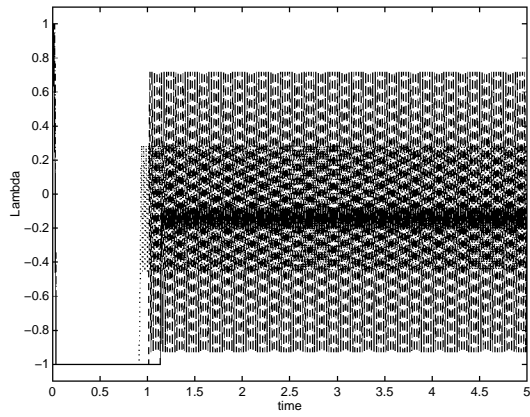


FIG. 3. - (TP1) - Multipliers of uncontrolled system (NR)

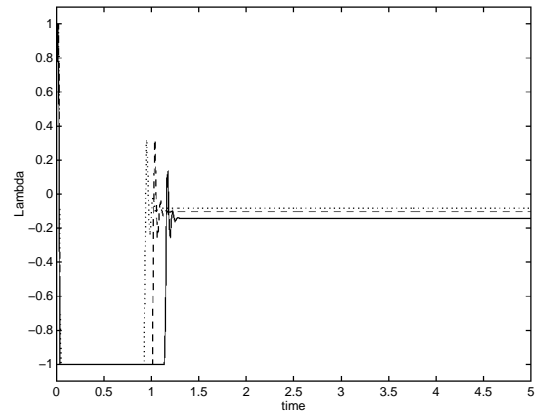


FIG. 4. - (TP1) - Multipliers of uncontrolled system (R)

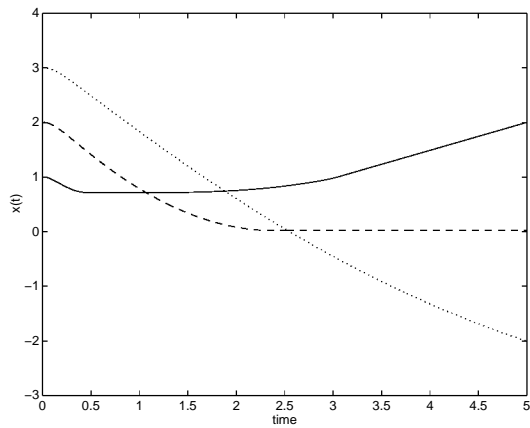


FIG. 5. - (TP1) - Displacements of controlled system

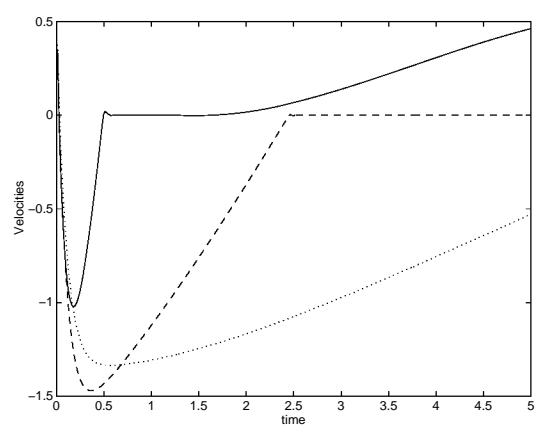


FIG. 6. - (TP1) - Velocities of controlled system

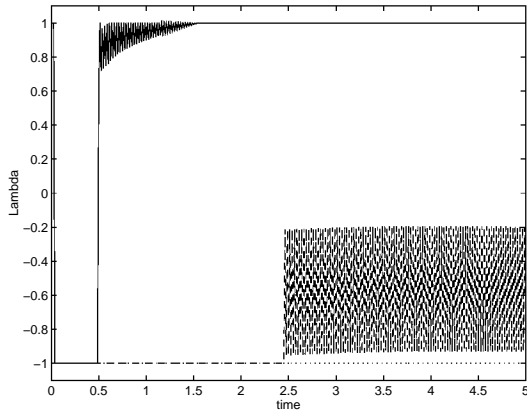


FIG. 7. - (TP1) - Multipliers of controlled system (NR)

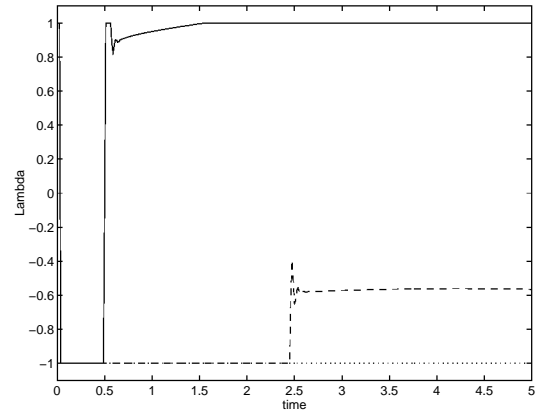


FIG. 8. - (TP1) - Multipliers of controlled system (R)

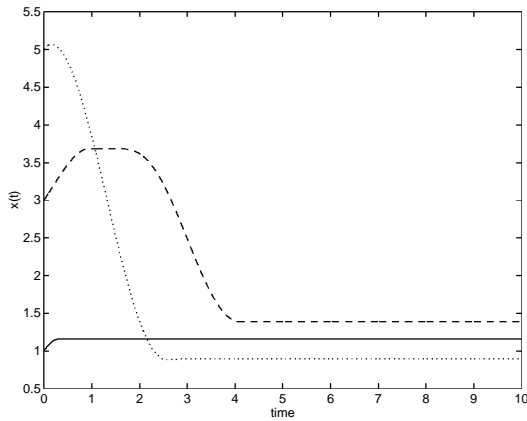


FIG. 9. - (TP2) - Displacements of uncontrolled system

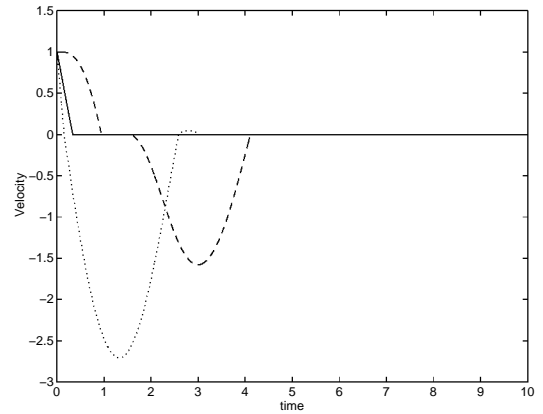


FIG. 10. - (TP2) - Velocities of uncontrolled system

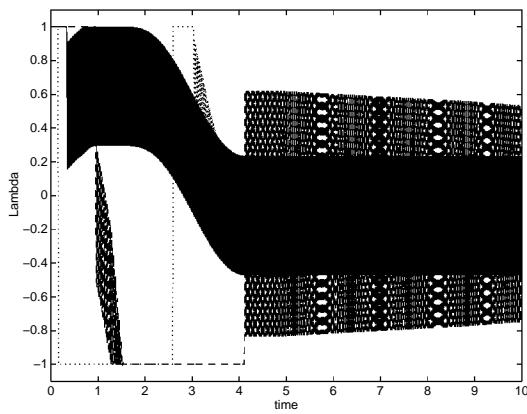


FIG. 11. - (TP2) - Multipliers of uncontrolled system (NR)

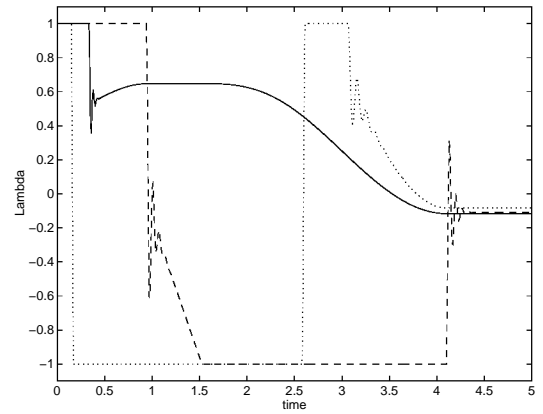


FIG. 12. - (TP2) - Multipliers of uncontrolled system (R)

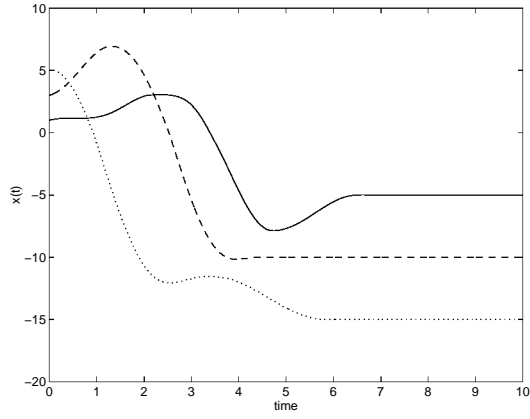


FIG. 13. - (TP2) - Displacements of controlled system

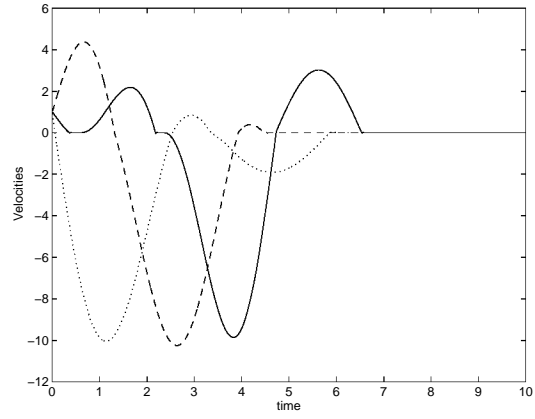


FIG. 14. - (TP2) - Velocities of controlled system

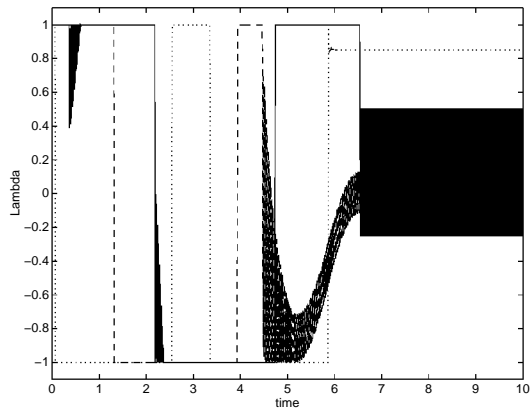


FIG. 15. - (TP2) - Multipliers of controlled system (NR)

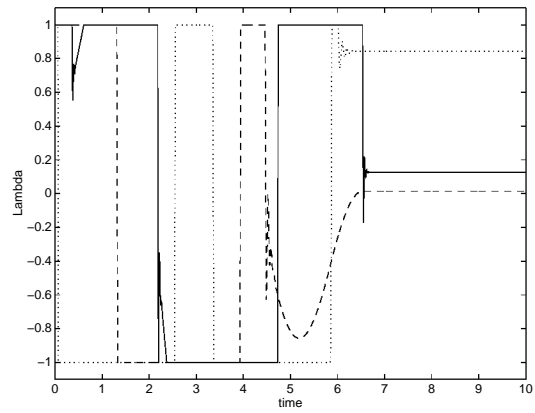


FIG. 16. - (TP2) - Multipliers of controlled system (R)