Proceedings of the 1992 IEEE
International Conference on Robotics and Automation
Nice, France - May 1992

A New Approach to the Numerical Solution of Constrained Mechanical System Dynamics

S. S. Keerthi, R. Sudarsan and N. K. Sancheti

Department of Computer Science & Automation
Indian Institute of Science, Bangalore-560 012, India

Abstract

A new approach to the numerical solution of the vector field associated with constrained mechanical systems, called as the perturbation approach is introduced. It is a specialization of a method of solving general vector fields, due to Shampine. The perturbation approach has several advantages over the existing approaches. The performance of the approaches on a constrained robot problem is used to show the efficacy of the new approach.

1 Introduction

The dynamic simulation of constrained mechanical systems formed by interconnected rigid bodies that undergo large displacements is a research area with applications in a broad range of engineering fields, e.g., robotics, automobile, aerospace and biomechanics. Constrained robotic systems have been addressed in a number of recent research papers [6, 7, 9, 10, 11, 12]. Constrained motion of a robotic manipulator leading to holonomic constraints results when the end-effector comes in contact with an environment that is completely rigid. The equations of motion of a constrained robot can be written as [7]

$$M(q)\ddot{q} + H(q,\dot{q}) = u + J'(q)\lambda, \tag{1.1}$$

$$\phi(q) = 0, \tag{1.2}$$

where $q \in R^N$ denotes the vector of generalized displacements, $M(q)$ is the manipulator inertia matrix which is symmetric and positive definite, $H(q,\dot{q})$ denotes the vector of coriolis, centripetal and gravity forces, u represents the vector of generalized forces applied at each joint, $\lambda \in R^L$ is a vector of multipliers corresponding to the constraint vector function $\phi : R^N \rightarrow R^L$, $J(q) = \phi_q(q)$ is the Jacobian matrix of $\phi(q)$ (which we will assume to have rank L always) and prime denotes transpose.

The control problem, i.e., choosing u (say, as a function of q and $\dot{q}$) so that (1.2) is satisfied and the solution of (1.1) behaves in a desired way, is an important problem which is addressed in [7, 10, 11, 12]. The choice for u is usually made by considering a linear model of (1.1)–(1.2). A validation of this choice requires the numerical solution of (1.1)–(1.2) for given initial conditions, which is the problem addressed in this paper.

Two repeated differentiations of (1.2) yield

$$J(q)\dot{q} = 0, \tag{1.3}$$

$$J(q)\ddot{q} + \rho(q,\dot{q}) = 0, \tag{1.4}$$

where ρ is appropriately defined. It is easy to verify that (1.1) and (1.4), which form a square linear system of equations in $(\ddot{q}, \lambda)$, can be solved to obtain $\ddot{q}$ and λ as functions of q and $\dot{q}$. Using this fact and letting $n = 2N$, $x = (q, \dot{q}) \in R^n$ we obtain the ordinary differential equation(ODE),

$$\dot{x} = f(x). \tag{1.5}$$

Also, combining (1.2) and (1.3) we get

$$g(x) = 0, \tag{1.6}$$

where $g(x) = (\phi(q), J(q)\dot{q}) \in R^m$, $m = 2L$. The system in (1.5)–(1.6) defines a vector field. In other words, if

$$\mathcal{M} = \{x : g(x) = 0\}$$

(a manifold of dimension $(n - m)$) and $T_x\mathcal{M}$ denotes the tangent space of $\mathcal{M}$ at $x \in \mathcal{M}$, then

$$f(x) \in T_x\mathcal{M} \ \ \forall x \in \mathcal{M}.$$

A direct numerical solution of (1.5) (starting from kinematically determined initial conditions) using ODE techniques is unsuitable because such an approach ignores (1.6). When local error control is employed by the ODE solver errors can build up and lead to a huge violation of (1.6). Such a violation is intolerable because (1.6) is an important physical constraint associated with the mechanical system. Thus, integration of (1.5) is as important as the stabilization of (1.6). Several approaches have been suggested in the literature for doing this. The popular ones are: parametrization[5, 8]; inexact constraint stabilization [1]; and exact constraint stabilization [2, 3]. In this paper we introduce a new approach which is conceptually very different from previous ones and which has good promise. We call this approach as the perturbation approach. It is a specialization of a method of solving a general vector field of the form (1.5)–(1.6), due to Shampine [13], to the constrained mechanical system equations defined by (1.1)–(1.4).

0-8186-2720-4/92 $3.00 ©1992 IEEE

The distinct feature of the perturbation approach is that, at each integration step it decouples the process of integrating (1.5) from the process of stabilizing the constraint in (1.6). It is this feature which helps the approach to overcome the defects usually associated with other approaches.

The paper is organized as follows. The perturbation approach for solving (1.5)–(1.6) is introduced and critically analyzed in section 2. Its specialization to constrained mechanical systems is the topic of section 3. In section 4 we compare the perturbation approach with other approaches and describe their performance on a constrained robot simulation problem. Conclusions are presented in section 5.

The following notations will be used in the paper. For $x \in R^n$, x^i denotes the i-th component of x. x_i will denote the i-th element of $\{x_i\}$, a sequence of vectors in R^n. Prime denotes transpose. For $x \in R^n$, $\|x\|$ =norm of x. For $1 \leq p \leq \infty$, $\|x\|_p = l_p$-norm of x. If $x \in R^n$ and $y \in R^m$, then (x, y) will denote the joint vector, $(x', y')'$. I_n is the $n \times n$ identity matrix. If $g : R^n \to R^m$ is differentiable then $g_x(x)$ is the $m \times n$ Jacobian of g evaluated at x.

2 The Perturbation Approach

In this approach, due to Shampine [13], a correction is applied to a numerical solution of (1.5) after each integration step so as to satisfy (1.6). Its theory is particularly strong when a single step integration method, such as Runge-Kutta, is used to solve (1.5).

To describe the approach, it is sufficient to say what is done in one integration step. Suppose k steps of the numerical solution of (1.5)–(1.6) have been done using the approach and $t = t_k$ has been reached. Let $x_k \in \mathcal{M}$ be the solution approximant at $t = t_k$. Denote the local solution by $x(\cdot)$, i.e., $x(\cdot)$ is the solution of (1.5) with $x(t_k) = x_k$. Let τ denote the integration tolerance. In the $(k+1)$st step, the aim is to determine a step size h_k and an $x_{k+1} \in \mathcal{M}$ that satisfy

$$\|x(t_{k+1}) - x_{k+1}\| \leq \tau, \tag{2.1}$$

where $t_{k+1} = t_k + h_k$. The determination of h_k and x_{k+1} is described by the following procedure.

Procedure P_1 Determination of h_k and x_{k+1} by the perturbation approach

1. Numerically integrate (1.5) from $x(t_k) = x_k$ using local error control (without concerning about (1.6)) to obtain a step size h_k and an approximant, $\bar{x}_{k+1}$ that satisfy

$$\|x(t_{k+1}) - \bar{x}_{k+1}\| \leq \tau/2. \tag{2.2}$$

2. Solve the problem

$$\min \ \|x - \bar{x}_{k+1}\| \quad \text{s.t.} \quad g(x) = 0; \tag{2.3}$$

and set x_{k+1} = the minimizer of (2.3). ■

The following theorem establishes the correctness of the procedure.

Theorem 3.1 The x_{k+1} determined by Procedure P_1 satisfies $x_{k+1} \in \mathcal{M}$ and (2.1).

Proof Since x_{k+1} is feasible for (2.3), it satisfies $x_{k+1} \in \mathcal{M}$. Consider $x(t_{k+1})$, the true local solution of (1.5) at t_{k+1}. Since $x(t) \in \mathcal{M}$ $\forall t$, $x(t_{k+1})$ is feasible for (2.3). By optimality, $\|x_{k+1} - \bar{x}_{k+1}\| \leq \|x(t_{k+1}) - \bar{x}_{k+1}\|$. Using this together with (2.2) and the triangle inequality we get

$$\begin{aligned}\|x_{k+1} - x(t_{k+1})\| &\leq \|x_{k+1} - \bar{x}_{k+1}\| + \|x(t_{k+1}) \\ &\quad -\bar{x}_{k+1}\| \leq 2\|x(t_{k+1}) - \bar{x}_{k+1}\| \leq \tau.\end{aligned} \tag{2.4}$$

■

Step 1 of Procedure P_1 can be carried out using any well-known numerical method for solving ODEs. Step 2 requires a subprocedure that solves (2.3). Before going in to the details of this subprocedure, let us make some useful remarks on the procedure and the approach.

Remark 2.1 Step 2 of Procedure P_1 is stronger than what is really needed. It is sufficient if x_{k+1} satifies

$$g(x_{k+1}) = 0, \quad \|x_{k+1} - \bar{x}_{k+1}\| \leq \frac{\tau}{2}. \tag{2.5}$$

By the first inequality of (2.4), and (2.2) it follows that x_{k+1} also satisfies (2.1). Though the feasibilty problem, (2.5) is usually as difficult to solve as the optimization problem, (2.3), it is useful because its solution is simpler to verify.

Remark 2.2 With all the popular integration methods, something more than an $\bar{x}_{k+1}$ satisfying (2.2) is available: an interpolant, $\bar{x} : [t_k, t_{k+1}] \to R^n$ that satisfies $\bar{x}(t_k) = x_k$, $\bar{x}(t_{k+1}) = \bar{x}_{k+1}$, and, $\|\bar{x}(t) - x(t)\| \leq \tau/2 \ \forall t \in [t_k, t_{k+1}]$. For each t, define $\tilde{x}(t)$ to be the solution of

$$\min \ \|x - \bar{x}(t)\| \ \text{s.t.} \ g(x) = 0. \tag{2.6}$$

Then, using the same ideas of Theorem 3.1's proof, it is easy to establish

$$\tilde{x}(t) \in M, \ \|\tilde{x}(t) - x(t)\| \leq \tau \ \forall t \in [t_k, t_{k+1}]. \tag{2.7}$$

Thus $\tilde{x}$ is a natural interpolant associated with the perturbation approach. The evaluation of this interpolant is somewhat expensive because it requires the solution of (2.6) at each t.

Remark 2.3 In step 1 integration is done using half the tolerance specified for the solution accuracy. This means that the perturbation approach is inefficient when compared with the direct approach of simply integrating (1.5). This is a cost one has to pay for ensuring (1.6). It should be mentioned that halving the tolerance does not double the integration cost. This is because, all the popular integration methods use a formula of the form αh^{p+1} =tolerance, to determine the step size, where p is the order of integration formula used. Let NINT_τ represnt the number of integration steps for tolerance $= \tau$. If p_{ave} denotes the average order of formula used for integration, then we can roughly say that

$$\frac{\text{NINT}_{\tau/2}}{\text{NINT}_\tau} \approx r \triangleq (2)^{1/(p_{ave}+1)}.$$

Even if we take $p_{ave} = 4$, which is a reasonable value for good codes, $r = 1.149$. Thus, halving the tolerance increases the number of integration steps needed to do an integration task only by about 15%.

Now let us look at the possibility of replacing the integration tolerance, $\tau/2$ used in step 1 by something bigger while ensuring that $x_{k+1} \in \mathcal{M}$ and (2.1) hold. There is something special when: the norm used in integration is a scaled l_2-norm, i.e., $\|x\| = \sqrt{x'Wx}$, where W is a symmetric positive definite matrix; and g is affine in the τ neighborhood of $\bar{x}_{k+1}$, i.e.,

$$g(x) = Ax + b \;\; \forall x \ni \|x - \bar{x}_{k+1}\| \leq \tau \,. \tag{2.8}$$

Since x_{k+1} solves (2.3), $(y - x_{k+1})'W(\bar{x}_{k+1} - x_{k+1}) = 0$ $\forall y \in \mathcal{M} \ni \|y - \bar{x}_{k+1}\| < \tau$. Thus

$$\begin{aligned} &\|\bar{x}_{k+1} - x(t_{k+1})\|^2 \\ &\quad = \|(\bar{x}_{k+1} - x_{k+1}) - (x(t_{k+1}) - x_{k+1})\|^2 \\ &\quad = \|(\bar{x}_{k+1} - x_{k+1})\|^2 + \|(x(t_{k+1}) - x_{k+1})\|^2 \end{aligned} \tag{2.9}$$

and hence,

$$\|x(t_{k+1}) - x_{k+1}\| \leq \|\bar{x}_{k+1} - x(t_{k+1})\|, \tag{2.10}$$

which means that x_{k+1} is more accurate than $\bar{x}_{k+1}$. In such a situation, then, it is alright to replace the tolerance $\tau/2$ in (2.2) by τ.

The scenario is different when g is nonlinear. It is easy to construct a near worst case situation to show that using $\tau/2$ in (2.2) is necessary. Although such a severe case is improbable when τ is small, the nonlinearity of g does cause distortions which disallow the obtainment of nice bounds such as (2.10). However, it is alright to say that, at stringent tolerances (2.8) is very nearly correct for $A = g_x(\bar{x}_{k+1})$, $b = g(\bar{x}_{k+1}) - g_x(\bar{x}_{k+1})\bar{x}_{k+1}$, and so the probability of (2.10) occuring is very high.

Remark 2.4 Shampine cautions the use of a variable order multistep integration method, such as Adams method, in step 1 of Procedure P_1. His main objection is that, the perturbations of step 2 may cause some "roughness" in the interpolant of the multistep method, which in turn may affect the order changing mechanism and cause inefficiencies. But, since the perturbations are structured (i.e, they are associated in a special way with the fundamental constraint, (1.6), of the true solution), it is our belief that Shampine's caution is not serious. However, except for the case when $g(x)$ is affine, we do not have a proper theory to support our argument. In numerical tests conducted on several problems we have found that the order changing process associated with the solution obtained by the perturbation approach is not much different from that associated with the direct solution of (1.5).

Let us now consider the solution of (2.3) required in step 2 of Procedure P_1. Scaled l_2 and l_∞ norms are the popular norms used for measuring integration errors. Shampine suggests a method for solving (2.3) when the scaled l_2 norm is used. He supports the use of l_2 norm by saying that, "if the ODE solver is based on the maximum norm (l_∞), one might well prefer to alter it to use the Euclidean norm (l_2) so as to arrive at a linear algebra problem which has a classical solution." For the scaled (l_2) norm we suggest a method which is slightly different but more efficient than Shampine's method.

Let $\|x\| = \sqrt{x'Wx}$, where W is a symmetric positive definite matrix. Typically, W is a diagonal matrix, the diagonal elements representing the variable weights applied to the different components of x. To simplify the notations, let us assume the objective function in (2.3) is replaced by half its square, and an appropriate coordinate transformation,

$$x = \bar{x}_{k+1} + TX, \tag{2.11}$$

(T = a nonsingular matrix such that $T'WT = I_n$) is done, so that (2.3) becomes

$$\min \;\; \|X\|_2^2/2 \quad \text{s.t.} \quad G(X) = 0, \tag{2.12}$$

where

$$G(X) = g(\bar{x}_{k+1} + TX). \tag{2.13}$$

The first order necessary conditions for (2.12) are

$$X - G'_X(X)\mu = 0, \quad G(X) = 0. \tag{2.14}$$

Remark 2.1 says that finding an optimal solution of (2.12) is not crucial. This allows us to incorporate the following efficiency-improving modification: replace the term $G'_X(X)$ in (2.14) by $G'_X(0)$. Thus we solve

$$X - G'_X(0)\mu = 0, \quad G(X) = 0, \tag{2.15}$$

instead of (2.14), to obtain a solution X^*.

Remark 2.5 Clearly, the X^* found above may not solve (2.12). If $\|X^*\| \leq \tau/2$, then we accept it and set $x_{k+1} = \bar{x}_{k+1} + TX^*$ for use in step 2 of Procedure P_1. This is because such an x_{k+1} satisfies (2.5), and, Remark 2.1 says this is sufficient for the success of Procedure P_1. If $\|X^*\| > \tau/2$ then there are two alternatives. Either a more detailed method is used to solve (2.12), or, step 1 of Procedure P_1 is repeated with a smaller integration step size and then step 2 is solved via (2.14). We feel the latter to be better because, the failure of the first order approach indicates a rapid change in g and so it is good to do a careful solution with a smaller integration step size. Our experience with numerical problems is that, cases where $\|X^*\| > \tau/2$ occurs are extremely rare.

Although the Newton-Raphson method can be used to solve (2.15), it will not be efficient because it requires the evaluation and factorization of the Jacobian of G at each iteration. The modified Newton-Raphson (MNR) method, which approximates the Jacobian of the future iterations by the Jacobian of the first iteration, is more appropriate for solving (2.15). The appropriate starting iterate for the MNR method applied to (2.15) is $X_0 = 0$, $\mu_0 = 0$. Let (x_l, μ_l) denote the l-th iterate. It is easy to verify, using the structure of the system in (2.15), that the $(l+1)$-th MNR iteration consists of: (i) solving

$$G_X(0)G'_X(0)\delta_\mu = G(X_l) \tag{2.16}$$

for δ_μ; and (ii) setting

$$\mu_{l+1} = \mu_l + \delta_\mu, \quad X_{l+1} = G'_X(0)\mu_{l+1}. \tag{2.17}$$

An efficient, as well as accurate, way of solving (2.16) is by computing the skinny QR decomposition of $G'_X(0)$, i.e., $G'_X(0) = Q_1 R_1$, where $Q_1 \in R^{n\times m}$ has orthonormal columns and $R_1 \in R^{m\times m}$ is upper triangular. Then (2.16) can be solved by replacing $G_X(0)G'_X(0)$ by $R'_1 R_1$ and solving the resulting twin triangular systems.

Shampine suggests a method that is equivalent to the Newton-Raphson method applied to (2.15). The first iterations of his method and our method are equivalent. Each additional iteration of his method requires the evaluation of a jacobian of G and the computation of a partial QR decomposition of its transpose; whereas, our method does not require these computations.

3 Specialization to Constrained Mechanical Systems

The vector field (1.5) and (1.6), which is derived from (1.1)–(1.4) has special features that can be nicely utilized to improve the efficiency of the perturbation approach. There are two key computations which benefit from the special structure: (i) the evaluation of $f(x)$; and (ii) the manifold-correction in step 2 of Procedure P_1. The evaluation of $f(x)$ requires the solution of the $(N + L)$ dimensional square linear system defined by (1.1) and (1.4). Special linear equation solvers, such as those described in [4] can be used to solve this system efficiently.

The manifold-correction step requires a more detailed discussion. Let $v = \dot{q}, x = (q, v)$ and $\|x\|$, the integration norm on x, be defined by $\|x\|^2 = q^T W_1 q + v^T W_2 v$ where W_1 and W_2 are positive definite matrices. Compute $N \times N$ nonsingular matrices T_1 and T_2 such that $T_1' W_1 T_1 = T_2' W_2 T_2 = I_N$. Usually W_1 and W_2 are diagonal weighting matrices and so the computations of T_1 and T_2 is easy.) Let $T =$ Block Diag$\{T_1, T_2\}$, $\bar{x}_{k+1} = (\bar{q}, \bar{v})$ and $X = (Q, V)$ so that (2.11) becomes

$$q = \bar{q} + T_1 Q, \quad v = \bar{v} + T_2 V. \tag{3.1}$$

Also, (2.12) becomes,

$$\min (\|Q\|_2^2 + \|V\|_2^2)/2 \text{ s.t. } G_1(Q) = 0,\ G_2(Q, V) = 0, \tag{3.2}$$

where

$$G_1(Q) = \phi(\bar{q}+T_1 Q),\ G_2(Q, V) = J(\bar{q}+T_1 Q)[\bar{v}+T_2 V]. \tag{3.3}$$

As mentioned in Remark 2.1 an exact solution of (3.2) is not necessary. This allows a simplifying assumption to be made on (3.2). Typically $J(q)$ is a slowly varying function of q. Also, since the value of Q in the solution of (3.2) is small ($\|Q\|_2 \le \tau/2$ so that q varies over a small range) we can replace $G_2(Q, V)$ in (3.2) by $\widetilde{G}_2(V)$ where

$$\widetilde{G}_2(V) = J(\bar{q})[\bar{v} + T_2 V]. \tag{3.4}$$

It is then easy to see that (3.2) gets decomposed in to the following smaller problems:

$$\min \ \|Q\|_2^2/2 \quad \text{s.t.} \quad G_1(Q) = 0, \tag{3.5}$$

$$\min \ \|V\|_2^2/2 \quad \text{s.t.} \quad \widetilde{G}_2(V) = 0. \tag{3.6}$$

Each of these problems can be solved using ideas similar to those in (2.14)–(2.17). The MNR iterations parallel to (2.16) and (2.17) require the skinny QR decomposition of

$$(G_1)_Q(0) = J(\bar{q})T_1 \quad \text{and} \quad (\widetilde{G}_2)_V(0) = J(\bar{q})T_2. \tag{3.7}$$

Since $\widetilde{G}_2$ is an affine function of V one MNR iteration will yield the exact solution of (3.6). The solution of (3.5) may require more than one iteration. Usually a tolerance on the satisfaction of (1.2) is used to terminate these iterations.

A particular assumption on the integration norm helps to improve efficiency significantly. Suppose we assume that W_1 and W_2, the weighting matrices for the position and velocity vectors, satisfy

$$W_2 = \alpha^2 W_1,$$

where α is a non-zero scalar having the units of time. (This is a reasonable assumption because it simply requires that the relative weighting among the position variables is the same as the relative weighting among the corresponding velocity variables.) Then $T_2 = (1/\alpha)T_1$. Therefore, if $J(\bar{q})T_1 = Q_1 R_1$ is the skinny QR decomposition of $J(\bar{q})T_1$ then $J(\bar{q})T_2 = Q_2 R_2$ with $Q_2 = Q_1$ and $R_2 = (1/\alpha)R_1$ is the skinny QR decomposition of $J(\bar{q})T_2$, and so a separate computation of the decomposition of $J(\bar{q})T_2$ is unnecessary.

4 Comparison with other approaches

Our comparison of the perturbation approach with others will be brief. A detailed comparison requires a detailed discussion of each of the other approaches, and it will be given in a future paper.

In the parametrization approach [5, 8] $\mathcal{M}$ is locally parametrized around a given initial point, $x_0 \in \mathcal{M}$:

$$x = \psi(y), \tag{4.1}$$

where $\psi : \Theta_Y \to \Theta_X \cap \mathcal{M}$, $x_0 = \psi(0)$, Θ_Y is an open set in R^{n-m} containing the origin, Θ_X is an open set in R^n containing x_0, and ψ is a diffeomorphism. This parametrization is used to replace (1.5) by a differential equtaion in y:

$$\dot{y} = \widetilde{f}(y). \tag{4.2}$$

The determinations of $\psi(y)$ and $\widetilde{f}(y)$ given y are simultaneous and require the numerical solution of a square system of m nonlinear equations. The parametrization approach consists of numerically solving (4.2).

The chief defects of the parametrization approach are that: (i) each $\widetilde{f}$ evaluation requires the solution of

an m-dimensional nonlinear system of equations apart from one f evaluation; and (ii) since the parametrization is local, a change in parametrization may be required during the solution, leading to an integration restart with associated inefficiencies. The perturbation approach does not suffer from these defects. It requires only one solution of an m-dimensional nonlinear system of equations in each integration step. (step 2 of Procedure P_1). Also, it does not require any integration restarts because it deals with the full ODE system in (1.5). The parametrization approach has the advantage that it integrates only the $(n-m)$-dimensional system of ODEs, (4.2), whereas the perturbation approach requires the integration of the n-dimesional system of ODEs, (1.5). This advantage, however, is only slight because the difference in the integration overhead costs of the two approaches is only $O(m)$ whereas the cost of every extra m-dimensional nonlinear system solution required by the parametrization approach is $O(m^3)$. A special parametrization approach called coordinate-partitioning [5] overcomes defect (ii) mentioned above. However its theory is not strong and it still suffers from defect (i).

In the inexact constraint stabilization approach [1], as applied to constrained mechanical systems, (1.4) is replaced by

$$[J(q)\ddot{q} + \rho(q,\dot{q})] + \alpha[J(q)\dot{q}] + \beta[\phi(q)] = 0, \qquad (4.3)$$

i.e., $\ddot{\phi} = 0$ is replaced by the equation $\ddot{\phi} + \alpha\dot{\phi} + \beta\phi = 0$, which makes $\phi = 0$ stable if α and β are chosen to be positive. Then the ODE system, derived by solving (1.1) and (4.3) for $\ddot{q}$ (and λ), is simply solved to obtain a solution of the dynamical system. The advantages of the approach are that it is simple and efficient. However it suffers from important defects such as: (i) a lack of a systematic way to choose α and β; (ii) the effect of α and β on the accuracy of the numerical solution; and (iii) the effect of α and β on the stiffness.

In the exact constraint stabilization approach [2, 3], as applied to (1.5) and (1.6), (1.5) is replaced by

$$\dot{x} = f(x) - g'_x(x)\mu, \qquad (4.4)$$

so that (4.4) and (1.6) form a system of differential-algebraic equations (DAEs) in (x,μ). It can be easily shown that $\mu = 0$ along every solution of (4.4), (1.6). Thus this DAE system is the same as the vector field, (1.5)–(1.6) and, its numerical solution using an appropriate DAE method (e.g., using backward difference formulas) yields a solution of (1.5) and (1.6). The approach can be nicely specialized to constrained mechanical systems [2, 3]. The cost per integration step of this approach is only slightly more than that of the perturbation approach. However we expect that the exact constraint stabilization approach will require many more integration steps to complete a solution since it can employ only the restricted set of DAE methods.

Let us now compare the performance of the various approaches on a constrained robotic system. The equations of motion of a 3-link cylindrical coordinate manipulator [7] are given by (1.1) and (1.2) where

$$M(q) = \text{diag}\,\{J_1 + J_2 + J_3 + m_3(ql_3)^2, m_2 + m_3, m_3\},$$

$$H(q,\dot{q}) = [2m_3(ql_3)\dot{q}_1\dot{q}_3, (m_2+m_3)g, -m_3(ql_3)\dot{q}_1^2].$$

The position of the robot end-effector, given by

$$x = (ql_3)\cos q_1, y = (ql_3)\sin q_1, z = q_2,$$

is constrained to move on a circle defined by $y - 0.169 = 0$ and $x^2 + z^2 - 0.2214 = 0$, leading to the constraint (1.2) where

$$\phi(q) = [(ql_3)\sin q_1 - 0.169, (ql_3)^2\cos^2 q_1 + q_2^2 - 0.2214],$$

and $ql_3 = q_3 + l_3$.

The nominal parameter values are $l_3 = 0.2\ m$; $m_2 = 1\ Kg$; $m_3 = 2\ Kg$; $J_1 = 0.1\ Kg.m^2$; and $J_3 = 0.1\ Kg.m^2$. The control law for stabilizing the system to the equilibrium point, $q_e = (0.436, 0.3, 0.2)$, designed using a linearization approach [7], is given by.

$$\begin{aligned} u_1 &= .363 - 3.655(q_1 - .436) - 4.181(q_2 - .3) \\ &\quad + 3.136(q_3 - .2) - .455\dot{q}_1 - 2.167\dot{q}_2 + 1.083\dot{q}_3, \\ u_2 &= 29.4 - 15.934(q_1 - .436) - 18.229(q_2 - .3) \\ &\quad + 13.672(q_3 - .2) - 2.167\dot{q}_1 - 10.325\dot{q}_2 + 5.163\dot{q}_3, \\ u_3 &= .423 + 8.267(q_1 - .436) + 9.455(q_2 - .3) \\ &\quad - 7.091(q_3 - .2) + 1.083\dot{q}_1 + 5.163\dot{q}_2 - 2.581\dot{q}_3. \end{aligned} \qquad (4.5)$$

The Adams-Bashforth-Moulton predictor-corrector method was used to do the integration. The integration norm used was the l_2-norm, with integration tolerance, $\tau = 10^{-4}$. The ideas in [8] were used for implementing the parametrization approach. The parameters for the inexact constraint stabilization approach were chosen to be $\alpha = 2\delta$ and $\beta = \delta^2$, as suggested by Baumgarte [1] to give a critical damping. Various values for δ were tried and $\delta = 2$ gave an optimal performance. The numerical solution corresponding to $\delta = 0$ (i.e., a direct solution of the ODE system corresponding to (1.1) and (1.4)) itself satisfied the constraints (1.2) and (1.3) very well. Table 5.1 and 5.2 give the comparative performance of the four approaches. These tables correspond to the initial conditions, $q = (0.486, 0.345, 0.162), \dot{q} = 0$, and $q = (2.8, 0, 0.3), \dot{q} = 0$, respectively. In these tables, the perturbation, the parametrization, the inexact constraint stabilization and the exact constraint stabilization approaches are respectively abbreviated as PERT, PARAM, ICS and ECS. Also, number of integration steps and the number of f-evaluations i.e. solutions of (1.1) and (1.4) are abbreviated as N_i and N_f respectively.

Let us now make some comments about the performance of the approaches. Owing to the strongly stabilizing nature of the feedback law, (4.5), Baumgarte's approach worked very well. In general this is not true. The initial condition corresponding to Table 5.1 was very close to the equlibrium point and hence all approaches except ECS performed nearly equally. The

Approach	N_i	N_f
PERT	33	66
PARAM	34	68
ICS, $\delta = 0$	32	64
ICS, $\delta = 2$	34	73
ECS	85	170

Table 4.1: Comparative Performance for the initial condition, $q = (0.486, 0.345, 0.162), \dot{q} = 0$.

Approach	N_i	N_f
PERT	121	243
PARAM	183	378
ICS, $\delta = 0$	98	196
ICS, $\delta = 2$	104	209
ECS	251	601

Table 4.2: Comparative Performance for the initial condition, $q = (2.8, 0, 0.3), \dot{q} = 0$.

second initial condition was far from the equilibrium point. Thus the parametrization approach required several changes in parametrization, leading to an overall inefficiency. As expected, the exact constraint stabilization approach required many more steps than others. The perturbation approach performed very well on the whole.

5 · Conclusions

In this paper we have introduced a new approach for the numerical solution of constrained mechanical system dynamics. The approach has a number of advantages over existing approaches. The comparison of the approaches on a constrained robotic system demonstrates this fact.

References

[1] J. Baumgarte, *"Stabilization of constraints and integrals of motion in dynamical systems"*, Comp. Meth. in Appl. Mech. Engg., vol. 1, 1972, pp. 1–16.

[2] C. Führer and B. Lumkuhrer, *"Numerical solution of the differential-algebraic equations for constrained mechnical motion"*, Numer. Math., Vol. 59, 1991, pp. 55–69.

[3] C. W. Gear, B. Leimkuhrer and G. K. Gupta, *"Automatic integration of Euler-Lagrange equations with constraints"*, J. Comp. Appl. Math., vol. 12, 1985, pp. 77–90.

[4] A. George, *"On block elimination for sparse linear systems"*, SIAM J. Numer. Anal., vol. 11, no. 3, 1974, pp. 585-603.

[5] E. J. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems*, vol. I: Basic Methods, Allyn and Bacon, Boston, 1989.

[6] R. Kankaanranta and H. N. Koivo, *"A model for constrained motion of a serial link manipulator"*, Proc. IEEE Intl. Conf. on Robotics and Automation, San Francisco, 1986.

[7] H. Krishnan and N. H. McClamroch, *"A new approach to position and contact free regulation in constrained robot systems"*, Proc. IEEE Intl. Conf. on Robotics and Automation, Cincinnati, 1990.

[8] N. K. Mani, E. J. Haug and K. E. Atkinson, *"Application of singular value decomposition for analysis of mechanical sytem dynamics"*, Trans. of ASME, J. Mech., Trans., Aut., in Design, vol. 107, 1985, pp. 82–87.

[9] N. H. McClamroch, *"Singular system of differential equations as dynamic models for constrained robot systems"*, Proc. IEEE Intl. Conf. on Robotics and Automation, San Francisco, 1986

[10] N. H. McClamroch and D. Wang, *"Feedback stabilization and tracking of constrained robot systems"*, IEEE Trans. Aut. Contr., Vol. AC-33, pp. 419–426,1988.

[11] J. K. Mills and A. A. Goldenberg, *"Force and position control of manipulators during constrained motion tasks"*, IEEE J. Rob. & Aut., Vol.RA–5, pp. 30–46, 1989.

[12] J. K. Mills, *"Control of robotic manipulators with flexible joints during constrained task execution"*, Proc. 28th CDC, Tampa, 1989.

[13] L. F. Shampine, *"Conservation laws and the numerical solution of ODEs"*, Comp. & Maths. with Applns., vol. 12B, 1986, pp. 1287–1296.