Stabilization of constrained mechanical systems with DAEs and invariant manifolds

Uri M. Ascher*
Department of Computer Science
University of British Columbia
Vancouver, British Columbia
Canada V6T 1Z2
ascher@cs.ubc.ca

Hongsheng Chin †
Department of Mathematics
University of British Columbia
Vancouver, British Columbia
Canada V6T 1Z2
hqin@math.ubc.ca

Linda R. Petzold‡
Department of Computer Science
University of Minnesota
Minneapolis, Minn. 55455
petzold@cs.umn.edu

Sebastian Reich §
Institut für Angewandte Mathematik und Stochastik
Mohrenstraße 39
Berlin, D–10117, Germany
na.reich@na-net.ornl.gov

October 26, 1994

*The work of this author was partially supported under NSERC Canada Grant OGP0004306.

†The work of this author was partially supported under NSERC Canada Grants OGP0004306 and OGP0000236

‡The work of this author was partially supported by the U. S. Army Research Office contract number DAAL03-89-C-0038 with the University of Minnesota Army High Performance Computing Research Center, and by ARO contract number DAAL03-92-G-0247 and DOE contract number DE-FG02-92ER25130.

§The work of this author was partially supported under NSERC Canada Grant OGP0004306 and by the German Science Foundation.

Abstract

Many methods have been proposed for the simulation of constrained mechanical systems. The most obvious methods have mild instabilities and drift problems, and consequently stabilization techniques have been proposed. A popular stabilization method is Baumgarte's technique, but the choice of parameters to make it robust has been unclear in practice.

Here we first review some of the simulation methods which have been proposed and used in computations from a stability point of view. This involves concepts of differential-algebraic equations (DAE) and ODE invariants. We explain why Baumgarte's method may run into trouble, and why a further quest for finding better parameter values for this method will always remain frustrating. We then show how to improve it.

We propose an efficient stabilization technique which may employ explicit ODE solvers in case of nonstiff or highly oscillatory problems and relates to coordinate projection methods. Examples of a two-link planar robotic arm and a squeezing mechanism illustrate the effectiveness of this new stabilization method.

1 Introduction

Many methods have been proposed and implemented in commercial codes for the simulation of constrained mechanical systems; see, e.g. [1, 2] and references therein. However, the most obvious of these methods have mild instabilities and drift problems, and consequently stabilization techniques have been proposed. A popular stabilization method is Baumgarte's technique [3], but the choice of parameters to make it robust has remained unclear in practice. Many attempts have been made in the literature to find a robust choice for these parameters – see, for example, the papers in [2]. One purpose of this paper is to survey some of these techniques, their advantages and limitations, from a stability point of view. We explain what troubles the Baumgarte technique may run into and explain why a further heuristic search for its parameter values is bound to fail. We then develop some new and better stabilization techniques. The mathematical and additional numerical analysis background behind this exposition can be found in [4, 5, 6].

In order to better understand the issues involved, it is useful to write down the Lagrangian formulation of the equations of motion describing the dynamics of a constrained multibody system:

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{v} \\ M(\mathbf{q})\dot{\mathbf{v}} &= \mathbf{f}(\mathbf{q},\mathbf{v}) - G(\mathbf{q})^T\lambda \\ \mathbf{0} &= \mathbf{g}(\mathbf{q}) \end{aligned} \tag{1.1}$$

where

$\mathbf{q}$ is the vector of generalized coordinates
$\mathbf{v}$ is the vector of generalized velocities
$M(\mathbf{q})$ is the mass matrix
$\mathbf{f}(\mathbf{q},\mathbf{v})$ is the vector of external forces (other than constraint forces)
$\mathbf{g}(\mathbf{q})$ is the vector of (holonomic) constraints
$G(\mathbf{q}) = \frac{\partial \mathbf{g}}{\partial \mathbf{q}}$ is the constraint Jacobian matrix
λ is the vector of Lagrange multipliers

We assume for simplicity that the mass matrix is symmetric positive definite and that the constraint Jacobian has a full row rank for all $\mathbf{q}(t)$ encountered. For notational simplicity, we have suppressed any explicit dependence of M, $\mathbf{f}$ or $\mathbf{g}$ on the time t. Also, we consider only holonomic constraints because they are the ones producing more stability difficulties when integrated numerically.

The system (1.1) is a system of differential-algebraic equations (DAE) of index 3 (the *index* is *one plus* the number of differentiations of the constraints that are needed in order to be able to eliminate the Lagrange multipliers λ). It is well-known that a direct finite-difference discretization of an index-3 DAE may yield practical difficulties [7], and this relates to the classical ill-posedness of higher index DAEs [8, 4]. This type of system is obtained for the dynamics of rigid bodies (or when applying modal analysis to flexible bodies) using the augmentation method (e.g. [1]). For some simple multibody systems, notably open loop systems, it is possible to explicitly reduce the

DAE to an ODE (of a smaller size) by using relative generalized coordinates and eliminating the constraints. The resulting ODE can then be integrated using ODE methods without worrying about the stability issues with which we are concerned here. Such a reduction cannot be done in general, though, and even when it can, the obtained differential equations are typically more complicated. We assume in any case, for the purposes of this article, that this reduction is not performed.

A very popular approach in practice is to differentiate the constraints twice, obtaining at each time t an algebraic system for the accelerations and the Lagrange multipliers. Thus, differentiating the position constraints

$$\mathbf{0} = \mathbf{g}(\mathbf{q}) \tag{1.2}$$

once, we obtain the constraint equations on velocity level

$$\mathbf{0} = \dot{\mathbf{g}} = G(\mathbf{q})\mathbf{v} \tag{1.3}$$

and a further differentiation with respect to time results in the constraint equations on acceleration level[1]

$$\mathbf{0} = \ddot{\mathbf{g}} = G(\mathbf{q})\dot{\mathbf{v}} + \mathbf{v}^T \mathbf{g}_{\mathbf{qq}}(\mathbf{q})\mathbf{v} \tag{1.4}$$

The ODE in (1.1) is written together with (1.4) as an index-1 DAE

$$\begin{pmatrix} M & G^T \\ G & 0 \end{pmatrix} \begin{pmatrix} \dot{\mathbf{v}} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \varphi \end{pmatrix} \tag{1.5}$$

where $\varphi = -\mathbf{v}^T \mathbf{g}_{\mathbf{qq}}(\mathbf{q})\mathbf{v}$, and this allows elimination of λ in terms of the accelerations $\dot{\mathbf{v}}$, obtaining an ODE system for $\mathbf{v}$ and $\mathbf{q}$

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{v} \\ M(\mathbf{q})\dot{\mathbf{v}} &= \tilde{\mathbf{f}}(\mathbf{q}, \mathbf{v}) \end{aligned} \tag{1.6}$$

which may be integrated using standard codes. (Note that, in principle, the index-reduced system (1.5) or (1.6) needs more initial conditions than the original system (1.1) to specify a unique solution. We assume, however, that *consistent* initial conditions (see, e.g. [7]) for the generalized position and velocity coordinates are provided.)

However, there are two disadvantages to integrating (1.6) or (1.5) numerically. The easily visible one is that the position and velocity constraints (1.2) and (1.3) are no longer satisfied exactly – there is a *drift* off the constraints, which does not look good in a graphical depiction of motion simulation. Moreover, though, the drift magnitude as well as the error in generalized positions and velocities *grows* with time t – at worst quadratically [3, 9, 4]. This is not because of the numerical method used

[1] Throughout this paper we will refer to (1.2) as the *position constraints*, to (1.3) as the *velocity constraints* and to (1.4) as the *acceleration constraints*, although of course these are all just different forms of the original constraints which are given on the generalized position coordinates.

to integrate (1.6) but because the system (1.6) or (1.5) itself is mildly unstable. All of the stabilization methods reviewed in §§2 and 3 below reduce the index of the original system to at most 2 in a stable way [4, 5, 16] yielding systems which can be safely discretized under certain conditions.

In §2 we consider Baumgarte's technique [3], and show why it may run into trouble in difficult situations, and why a further heuristic search for good parameter values with this method is to be discouraged. In §3 we review a number of good stabilization techniques, briefly commenting on their merits and disadvantages. In §4 we view the position and velocity constraints (1.2) and (1.3) as defining an invariant manifold for the solution of the augmented ODE (1.6) and seek to stabilize the manifold. This leads in §5 to practical discretization schemes which in turn relate to, and shed a new light on, some of the methods of §3. These schemes are particularly useful for nonstiff problems (including highly oscillatory ones), where explicit ODE integration schemes may be employed. A particularly attractive method of this type is proposed and implemented [6]. Examples utilizing a double pendulum with a constrained path (or a two-link planar robotic arm) and a squeezing mechanism are given in §6.

2 Baumgarte's technique

Using Baumgarte's technique [3], we consider the index-1 DAE (1.5) or the corresponding ODE (1.6) obtained by eliminating the Lagrange multipliers, but now φ is defined by

$$\varphi = -\mathbf{v}^T \mathbf{g}_{\mathbf{qq}}(\mathbf{q})\mathbf{v} - \alpha_1 \dot{\mathbf{g}}(\mathbf{q}, \mathbf{v}) - \alpha_0 \mathbf{g}(\mathbf{q}) \tag{2.1}$$

where the parameters α_j are chosen so that the roots of the polynomial

$$\sigma(\tau) = \tau^2 + \alpha_1 \tau + \alpha_0$$

both have negative real parts. For instance, one may choose

$$\sigma(\tau) = (\tau + \gamma)^2 \tag{2.2}$$

for some $\gamma > 0$. The effect of this is to replace eq. (1.4) by

$$\mathbf{0} = \ddot{\mathbf{g}} + 2\gamma\dot{\mathbf{g}} + \gamma^2 \mathbf{g} \tag{2.3}$$

If we view $\mathbf{g}$ as a vector of dependent variables then we have replaced a mildly unstable ODE which allows perturbations to grow linearly in time by an asymptotically stable ODE, where perturbations decay with time. To make these observations more precise, we would linearize the position constraints and split a given unknown vector function into its component in the range space of G plus its component in the orthogonal subspace (i.e. a direct sum); see §3 of [4] for the details. Another way of viewing the Baumgarte technique is by regarding the invariant manifold that the unsatisfied constraints (1.2) and (1.3) define with respect to the ODE (1.6). In case of the unstabilized index reduction (1.4), this manifold is mildly unstable, while Baumgarte's technique makes it an attracting manifold.

In terms of a numerical discretization by finite differences with a fixed step size, truncation errors along the manifold in the unstabilized case may accumulate quadratically in time, because the error committed at each step grows linearly [9], whereas in the stabilized case these errors do not accumulate. Of course, the errors in the orthogonal direction to the manifold may well accumulate even in the stabilized case.

The apparent conceptual simplicity of the Baumgarte stabilization technique and the fact that it essentially replaces the index-3 DAE (1.1) by an ODE formulation must be considered a major reason for its popularity in engineering applications. But the practical choice of parameters (e.g. γ in (2.3)) to make it robust is widely regarded as unknown, despite many attempts (see, e.g., [2]). We now give three indications to explain why this parameter choice is indeed inherently difficult and in a sense impossible.

First, note that the form (1.5) or (1.6) with the stabilization (2.1) suggests that the parameter γ should be independent of the discretization method and of the discretization step size (say h). But such a conclusion would be wrong in practice. In fact, our results indicate that the optimal γ does indeed depend on both the discretization step size h and the discretization method. This can be easily seen using the following simple example:

Example 1

Let us simplify the multibody equations (1.1) by assuming a constant mass matrix M and a constant constraint Jacobian G, with $\mathbf{g} = G\mathbf{q}$. Then the position and velocity level constraints are $\hat{\mathbf{q}} = \hat{\mathbf{v}} = \mathbf{0}$, where $\hat{\mathbf{q}} = G\mathbf{q}$ and $\hat{\mathbf{v}} = G\mathbf{v}$. Further, apply a forward Euler discretization with a constant step h. Baumgarte's technique then gives

$$\begin{aligned} \mathbf{q}_{n+1} &= \mathbf{q}_n + h\mathbf{v}_n \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + hM^{-1}\mathbf{f}(\mathbf{q}_n, \mathbf{v}_n) - \\ & \quad hM^{-1}G^T(GM^{-1}G^T)^{-1}[GM^{-1}\mathbf{f}(\mathbf{q}_n, \mathbf{v}_n) + \alpha_0 G\mathbf{q}_n + \alpha_1 G\mathbf{v}_n] \end{aligned} \tag{2.4}$$

where $\mathbf{q}_n$ denotes the approximation of $\mathbf{q}(t_n)$, $t_{n+1} = t_n + h$, etc.

To observe the drift we multiply both equations of (2.4) by G and write them in terms of $\hat{\mathbf{q}}$ and $\hat{\mathbf{v}}$:

$$\begin{pmatrix} \hat{\mathbf{q}}_{n+1} \\ \hat{\mathbf{v}}_{n+1} \end{pmatrix} = B \begin{pmatrix} \hat{\mathbf{q}}_n \\ \hat{\mathbf{v}}_n \end{pmatrix}, \qquad B = \begin{pmatrix} I & hI \\ -\alpha_0 hI & (1-\alpha_1 h)I \end{pmatrix}$$

The question then becomes how to choose the Baumgarte parameters α_0 and α_1 to minimize the spectral radius of the amplification matrix B (note that $\|B\|_\infty > 1$ regardless). Calculating the eigenvalues of B we find that they can be optimally set to 0 if we choose $\alpha_1 = \frac{2}{h}$, $\alpha_0 = \frac{\alpha_1^2}{4} = \frac{1}{h^2}$, i.e. $\gamma = \frac{1}{h}$ in (2.2). This choice,

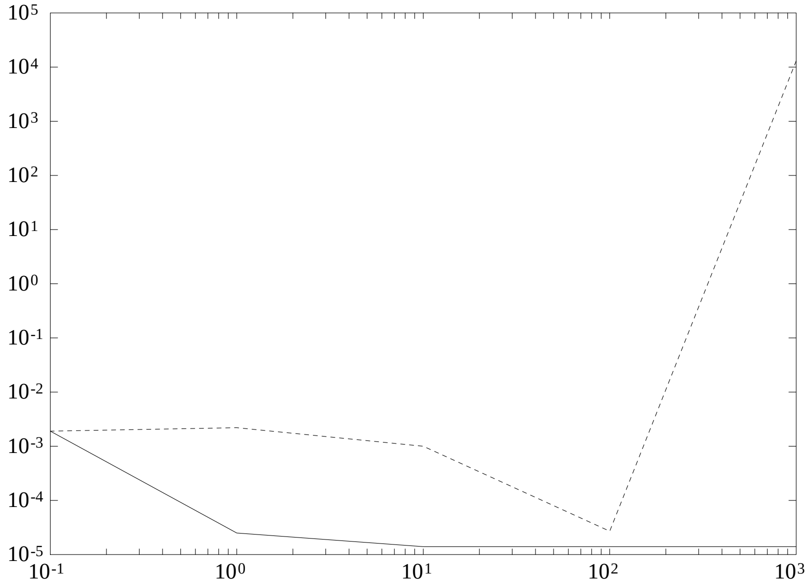

Figure 2.1: Error behaviour as a function of γ. The solid line is good, the dashed line is bad.

which certainly does depend on h, gives $B = \begin{pmatrix} I & hI \\ -h^{-1}I & -I \end{pmatrix}$, $B^2 = 0$. The fact that $\|B\|_\infty > h^{-1}$ now gives cause for concern, although of course there are other matrix norms in which $\|B\| < 1$ (since the spectral radius is 0). Thus we do not expect the choice $\gamma = \frac{1}{h}$ to be necessarily optimal in practical, nonlinear situations either. (For such calculations it is really important to make both $\|B\|$ not large and $\|B^2\|$ small, and these are seen to be conflicting desires for the Baumgarte technique.)

The method that we propose later on yields, by contrast, $B = 0$. $\Box$

Another difficulty with Baumgarte's technique is explained as follows: Since from (2.3) the larger the parameter γ the more attracting the invariant manifold becomes, one would have hoped that "it is safe to take γ as large as we wish", which would have made the choice of γ simpler. But when $\gamma \to \infty$ such that $\gamma h \gg 1$, the discretized problem is close to a direct discretization of the original index-3 DAE (1.1), and therefore numerical stability difficulties arise. Thus, referring to Fig. 2.1 which depicts solution errors as a function of γ, while one would hope for an error curve like the solid line, which never increases, one may get instead a curve like the dashed line.

Finally, an additional difficulty arises when $\|GM^{-1}G^T\| \ll \|G\|\|M^{-1}G^T\|$, which may occur in a heterogeneous mechanical system. In such a case an unreasonably small step size may be needed in order to recover the asymptotic stability of the stabilized manifold. For examples and discussion see §4 of [4]. Example 2 of [4] yields the dashed curve of Fig. 2.1, and a variant using GG^T instead of $GM^{-1}G^T$ in §4 below yields the solid curve of Fig. 2.1 and resolves the difficulty in that example.

3 Other good techniques

A variety of other solution techniques, consisting at least in part of reformulating the given constrained formulation (1.1), have been proposed in the literature. They can be divided into state-space formulations and projection methods. All of these methods require the solution of a set of nonlinear equations at each step. Below we give a short characterization of these two types of methods. We start with the state-space formulation.

3.1 State-space formulation

One class of methods reformulates the problem locally into state-space form [10, 11, 12]. The DAE (1.1) is considered as an ODE on the manifold defined by the position and velocity constraints (1.2), (1.3), and a local parameterization is carried out to explicitly yield this reduction. Suppose that there are n generalized coordinates in $\mathbf{q}$ and that there are m constraints in (1.2). At each point t consider an $(n-m)\times n$ matrix R such that $\begin{pmatrix} R \\ G \end{pmatrix}$ is nonsingular. A simple practical choice for R is to be piecewise constant (at least over one step of integration). Then an ODE can be locally derived for the state-space variable

$$\mathbf{u} = R\mathbf{q} \tag{3.1}$$

insisting that the constraints be satisfied. The matrix function R must be chosen, for stability reasons, such that

$$\left\| \begin{pmatrix} R \\ G \end{pmatrix}^{-1} \right\| \le K, \qquad \|R\| \le K \tag{3.2}$$

for a constant K of moderate size (cf. [4]). In a coordinate partitioning method [10] R is a matrix whose rows are unit vectors, thus choosing certain components of $\mathbf{q}$ to form $\mathbf{u}$. Another idea is to make $RG^T = 0$ [11] at the beginning of each integration step. In any case, when (3.2) is deemed violated, a new constant matrix R is chosen based on a new reference point, giving a different state-space ODE for a new $\mathbf{u}$ of (3.1). The segments are connected in such switching points through continuity of $\mathbf{q}$ and $\mathbf{v}$. The advantages of such schemes are their reduced size and their stability (provided (3.2) holds) and no-drift. A robust detection scheme for the necessity to change R is the more difficult aspect of these schemes, however.

3.2 Projection methods

For the rest of this section, consider the reduced DAE (1.5) or the ODE (1.6), and the invariant manifold defined by the position and velocity constraints (1.2), (1.3). One can view the system (1.5), (1.2), (1.3) as an *overdetermined DAE* [1, 13]. Given appropriate initial conditions, such a system has a unique solution; however, upon numerical discretization there is no exact solution to the overdetermined system. Various projection schemes were proposed in order to solve it, in that their discretization is no longer overdetermined.

There are two basic ways to project the solution onto the constraint manifold (or part of it). One is to redefine the ODE (1.6) by adding new Lagrange multipliers [14, 4, 15, 16]. This is the method of *projected invariants*. For example, if we only project onto the original position constraints (1.2) we get (with μ the new Lagrange multipliers)

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{v} - G(\mathbf{q})^T \mu \\ M(\mathbf{q})\dot{\mathbf{v}} &= \tilde{\mathbf{f}}(\mathbf{q}, \mathbf{v}) \\ \mathbf{0} &= \mathbf{g}(\mathbf{q}) \end{aligned} \tag{3.3}$$

This is a stable index-2 DAE which can be discretized either by a BDF method or by a stiffly stable implicit Runge-Kutta method. Thus, a stable index reduction has been achieved. A projection onto the velocity constraints (1.3) can be added as well, and this may in some cases allow a larger step size in the ensuing discretization at the expense of a larger system to solve at each step.

The other approach is to proceed to discretize numerically the ODE (1.6), but at the end of each discretization step to project the obtained approximate solution onto the selected constraints manifold. This is referred to as the method of *coordinate projection* (see, e.g., [17, 18, 16]). In both approaches it is possible to choose to project onto the position constraints manifold (1.2), or onto the velocity constraints manifold (1.3), or onto both (which may be more expensive). Both approaches lead effectively to stable index reduction and thus to the possibility of a stable solution of the original problem. Finally, when the mass matrix M involves different scales, it can be important in both approaches to project using G^T, not $M^{-1}G^T$, in order to allow a reasonably large discretization time step h [4]. Note, though, that there is potentially an additional expense involved per step because a decomposition of $GM^{-1}G^T$ but not of GG^T is already used anyway to obtain (1.6). However, this expense does not have to be significant, see [19, 5].

4 Stabilization of invariants

In this and the following section we derive a stabilized ODE formulation that improves the stabilizing properties of Baumgarte's method and makes the choice of the parameters straightforward. Our aim is to retain the computational simplicity of

Baumgarte's approach. In particular, we later explore the use of explicit ODE integration schemes. This means that we do not explore stiff systems and do not insist that the solution precisely lie in the constraint manifold.

Let us write the ODE (1.6) as

$$\mathbf{z}' = \hat{\mathbf{f}}(\mathbf{z}) \tag{4.1}$$

for $\mathbf{z} = (\mathbf{q}, \mathbf{v})$. The position and velocity constraints together form an invariant set $\mathcal{M}$ of this ODE, given by

$$\mathbf{0} = \mathbf{h}(\mathbf{z}) = \begin{pmatrix} \mathbf{g}(\mathbf{q}) \\ G(\mathbf{q})\mathbf{v} \end{pmatrix} \tag{4.2}$$

assuming, to recall, that

$$H(\mathbf{z}) = \mathbf{h}_{\mathbf{z}}(\mathbf{z}) = \begin{pmatrix} G & 0 \\ \mathbf{v}^T \mathbf{g}_{\mathbf{qq}} & G \end{pmatrix}$$

has a full row rank. (If we choose to concentrate only on the velocity constraints (1.3) then (4.2) is identified with (1.3) for $\mathbf{v}$ alone, and $H = G$.)

Consider the family of stabilization methods

$$\mathbf{z}' = \hat{\mathbf{f}}(\mathbf{z}) - \gamma F(\mathbf{z})\mathbf{h}(\mathbf{z}) \tag{4.3}$$

where $\gamma > 0$ is a parameter and

$$F = D(HD)^{-1} \tag{4.4}$$

with $D(\mathbf{z})$ smooth such that HD is nonsingular (indeed, $\|HD\|\|(HD)^{-1}\|$ should be nicely bounded) for each $\mathbf{z}$. For instance, we can choose $D = H^T$, or the cheaper variant

$$D = \begin{pmatrix} G^T & 0 \\ 0 & G^T \end{pmatrix} \tag{4.5}$$

yielding

$$F = G^T(GG^T)^{-1} \begin{pmatrix} I & 0 \\ -\mathbf{v}^T \mathbf{g}_{\mathbf{qq}} G^T (GG^T)^{-1} & I \end{pmatrix} \tag{4.6}$$

which has the advantage that only GG^T needs to be decomposed (or "inverted"). See [5] for additional possibilities.

This ODE, its invariant and its stabilization given by (4.1)–(4.4) have a general form. It does not include the Baumgarte technique for an index-3 DAE (2.1)–(2.3), but it does include a Baumgarte technique for an index-2 DAE (e.g. for a multibody system with nonholonomic constraints $\mathbf{g}(\mathbf{v}) = \mathbf{0}$), where (2.3) is replaced by $\mathbf{0} =$

$\dot{\mathbf{g}} + \gamma \mathbf{g}$. In this case we can identify H with G and D with $M^{-1}G^T$, see [5] and recall §2.

It was shown in [5] that the manifold $\mathcal{M}$ of (4.2) is an asymptotically stable invariant manifold of the ODE (4.1) for all $\gamma > 0$, and that the flow of (4.3) on $\mathcal{M}$ reduces to the flow of (4.1) restricted to $\mathcal{M}$. Moreover, rewriting the stabilized system (4.3) as

$$\begin{aligned} \mathbf{z}' &= \hat{\mathbf{f}}(\mathbf{z}) - D(\mathbf{z})\mu \\ \mathbf{0} &= \mathbf{h}(\mathbf{z}) - \frac{1}{\gamma}(HD)(\mathbf{z})\mu \end{aligned} \tag{4.7}$$

we see that as we let $\gamma \to \infty$ the formulation reduces to the index-2 DAE

$$\begin{aligned} \mathbf{z}' &= \hat{\mathbf{f}}(\mathbf{z}) - D(\mathbf{z})\mu \\ \mathbf{0} &= \mathbf{h}(\mathbf{z}) \end{aligned} \tag{4.8}$$

obtaining the projected invariant method described in the previous section. Thus, unlike for Baumgarte's technique, the limit $\gamma \to \infty$ is a "safe" limit. It therefore makes sense to discretize the stable ODE (4.3), which we now proceed to consider.

5 Discretization of the stabilized ODE formulation

The stabilized ODE (4.3) can be safely integrated by a general purpose package for initial value ODEs. However, there remains the question of determining the parameter γ. If the package integrates stiff ODE problems effectively then γ can be taken very large. But if a nonstiff ODE solver is desired then γh should be in the absolute stability region of the method used when the stepsize is h. Moreover, the two terms on the right hand side of (4.3) differ substantially from each other, both in purpose ($-\gamma F \mathbf{h}$ is just a stabilization term) and in size. Hence it makes sense to apply different discretization schemes to them.

Let us consider the discretization of the ODE (4.1) (i.e. the ODE (1.6), obtained by directly differentiating the constraints twice from the original constrained multibody system (1.1)), by a textbook one–step scheme, e.g. Runge-Kutta of order $p \geq 1$. This results in the time–h–map

$$\mathbf{z}_{n+1} = \phi_h^f(\mathbf{z}_n), \tag{5.1}$$

which advances the solution from the approximate state $\mathbf{z}_n$ at $t = t_n$ to an approximate state $\mathbf{z}_{n+1}$ at $t_{n+1} = t_n + h$. For the stabilization term, it suffices to apply a first order method (this term vanishes at the exact solution). Using backward Euler, for example, we get

$$\mathbf{z}_{n+1} = \phi_h^f(\mathbf{z}_n) - \alpha F(\mathbf{z}_{n+1})\mathbf{h}(\mathbf{z}_{n+1}) \tag{5.2}$$

with $\alpha = h\gamma$. It was shown in [5] (Theorem 3.1) that the obtained scheme does retain the global error $O(h^p)$ in $\mathbf{z}_{n+1}$, and that the constraints (4.2) are satisfied to $O(h^{p+1}/\alpha)$. This scheme possesses an invariant manifold $\mathcal{M}_h$ which is asymptotically stable. This is a solid, stable scheme (for an appropriate choice of D), but it does not buy us much new: it is implicit, the best choice of α is $\alpha \to \infty$, and in this limit we obtain the coordinate projection method

$$\begin{aligned} \mathbf{z}_{n+1} &= \phi_h^f(\mathbf{z}_n) - D(\mathbf{z}_{n+1})\mu \\ \mathbf{0} &= \mathbf{h}(\mathbf{z}_{n+1}) \end{aligned} \tag{5.3}$$

(compare this to (4.8)).

An explicit alternative to the scheme (5.2) is therefore derived next by evaluating the stabilizer $F\mathbf{h}$ at the argument obtained by applying the higher order discretization scheme (5.1). This yields the method

$$\tilde{\mathbf{z}}_{n+1} = \phi_h^f(\mathbf{z}_n) \tag{5.4a}$$

$$\mathbf{z}_{n+1} = \tilde{\mathbf{z}}_{n+1} - \alpha F(\tilde{\mathbf{z}}_{n+1})\mathbf{h}(\tilde{\mathbf{z}}_{n+1}) \tag{5.4b}$$

The obtained scheme can be also viewed as a modification of a forward Euler discretization of the stabilization term. It was shown in [5] (Theorem 3.2) that the obtained scheme (5.4) retains the global error $O(h^p)$ in $\mathbf{z}_{n+1}$, and that the constraints (4.2) are satisfied to $O(h^{p+1})$. This scheme possesses an invariant manifold $\mathcal{M}_h$ which is asymptotically stable. Moreover, any choice of α in the range $0 < \alpha < 2$ is stable, and the choice $\alpha = 1$, i.e. $\gamma = 1/h$ (which certainly depends on the discretization step size h), is close to optimal.

The scheme (5.4) appears to give a good compromise between the requirements of efficiency and stability. It can be considered as applying one Newton iteration, constrained to be in the range of D, for the solution of the nonlinear system appearing in the coordinate projection method (5.4) at each time step. While such an approximation to (5.4) has been proposed before and observed to work well in practice (see, e.g. [17, 18]), here this approximation is actually justified, using a different point of view. The scheme (5.4) is identical to the coordinate projection method if the constraints $\mathbf{h}(\mathbf{z})$ are linear (even if they depend explicitly on time). Note that the velocity constraints (4.3) are indeed linear in the generalized velocities $\mathbf{v}$ (assuming holonomic constraints (1.2) to begin with).

Setting $\alpha = 1$ we write this method for the mechanical system model: At a time step n, apply the following two-stage discretization step:

1. Starting with $(\mathbf{q}_n, \mathbf{v}_n)$ at $t = t_n$, use a favourite ODE integration scheme ϕ_h^f (e.g. Runge-Kutta or multistep) to advance the system

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{v} \\ M(\mathbf{q})\dot{\mathbf{v}} &= \mathbf{f}(\mathbf{q}, \mathbf{v}) - G^T(\mathbf{p})\lambda \\ \mathbf{0} &= G(\mathbf{q})\dot{\mathbf{v}} + \mathbf{v}^T \mathbf{g}_{\mathbf{qq}}(\mathbf{q})\mathbf{v} \end{aligned}$$

by one step. Denote the resulting values at $t_{n+1} = t_n + h$ by $(\tilde{\mathbf{q}}_{n+1}, \tilde{\mathbf{v}}_{n+1})$.

2. Stabilize:

$$\begin{pmatrix} \mathbf{q}_{n+1} \\ \mathbf{v}_{n+1} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{q}}_{n+1} \\ \tilde{\mathbf{v}}_{n+1} \end{pmatrix} - F(\tilde{\mathbf{q}}_{n+1}, \tilde{\mathbf{v}}_{n+1})\mathbf{h}(\tilde{\mathbf{q}}_{n+1}, \tilde{\mathbf{v}}_{n+1})$$

where $\mathbf{h}$ is given by (1.2) and (1.3), and F is given by (4.4).

For instance, F may be given by $D = H$ or by (4.6).

Unfortunately, both of these choices of F contain a term involving $\mathbf{g}_{\mathbf{qq}}$ which we may wish to avoid in order to economize the computation. We therefore consider also the choices

$$F = M^{-1}G^T(GM^{-1}G^T)^{-1}\begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} \tag{5.5}$$

or

$$F = G^T(GG^T)^{-1}\begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} \tag{5.6}$$

which are sufficient in many applications (when $\mathbf{v}^T\mathbf{g}_{\mathbf{qq}}$ does not significantly dominate G).

A choice which we finally recommend [6] is to replace the stabilization step above by the double step

$$\begin{pmatrix} \hat{\mathbf{q}}_{n+1} \\ \hat{\mathbf{v}}_{n+1} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{q}}_{n+1} \\ \tilde{\mathbf{v}}_{n+1} \end{pmatrix} - F(\tilde{\mathbf{q}}_{n+1}, \tilde{\mathbf{v}}_{n+1})\mathbf{h}(\tilde{\mathbf{q}}_{n+1}, \tilde{\mathbf{v}}_{n+1})$$
$$\begin{pmatrix} \mathbf{q}_{n+1} \\ \mathbf{v}_{n+1} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{q}}_{n+1} \\ \hat{\mathbf{v}}_{n+1} \end{pmatrix} - F(\tilde{\mathbf{q}}_{n+1}, \tilde{\mathbf{v}}_{n+1})\mathbf{h}(\hat{\mathbf{q}}_{n+1}, \hat{\mathbf{v}}_{n+1})$$

with F given by (5.5) or (5.6). This gives a drift error of $O(h^{2p})$ rather than $O(h^{p+1})$ (see [6]) at a negligible additional cost, since F is evaluated at most once per time step. In particular, using (5.5) the cost of the entire stabilization step can be easily made to be well below the cost of one stage of a Runge-Kutta step.

6 Examples and code

Example 2

We have performed a number of calculations for the problem of a two-link robotic arm. This is a double planar pendulum with a prescribed path at its "free" end (see, e.g., [20]). Thus, one end of a rigid rod is fixed at the origin, and the other is connected to another rigid rod with rotations allowed in the $x - y$ plane. Let θ_1 be

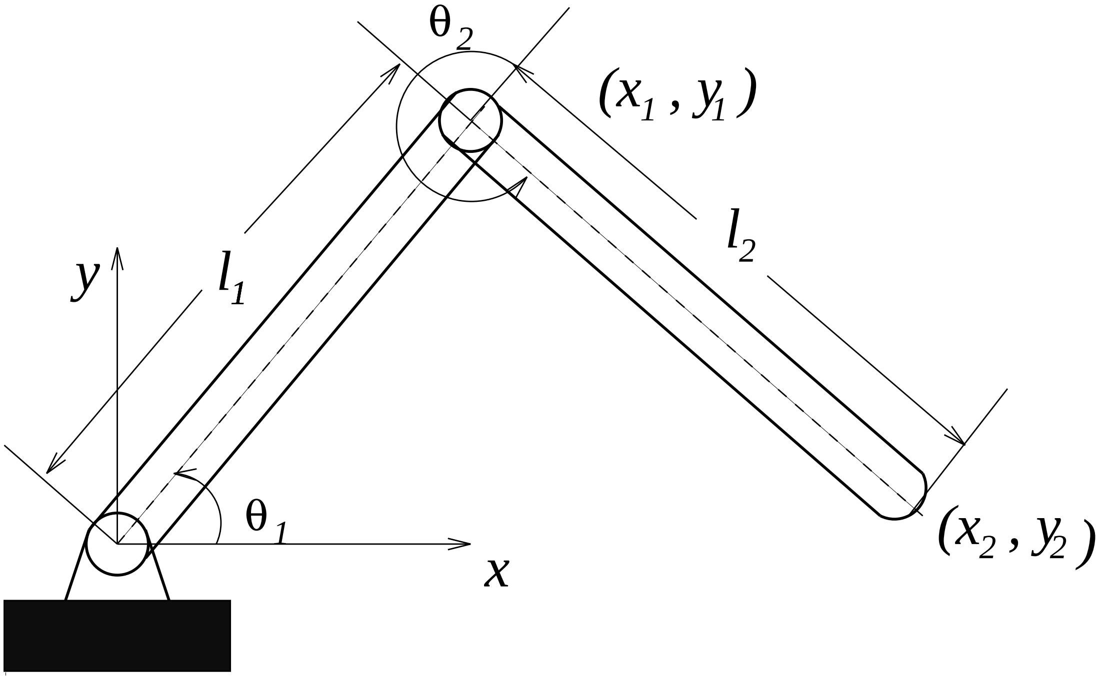

Figure 6.1: Two-link planar robotic system

the angle that the first rod makes with the horizontal axis, and let θ_2 be the angle that the second rod makes with respect to the first rod (see Fig. 6.1). The masses of the rods are denoted by m_i and their lengths are denoted by l_i. The coordinates of the link between the rods are given by

$$x_1 = l_1 c_1 \qquad y_1 = l_1 s_1$$

and those of the "free" end are

$$x_2 = x_1 + l_2 c_{12} \qquad y_2 = y_1 + l_2 s_{12}$$

where $c_i = \cos\theta_i$, $s_i = \sin\theta_i$, $c_{12} = \cos(\theta_1 + \theta_2)$, $s_{12} = \sin(\theta_1 + \theta_2)$.

Referring to the notation of (1.1), we let $\mathbf{q} = (\theta_1, \theta_2)^T$ and obtain

$$M = \begin{pmatrix} m_1 l_1^2/3 + m_2(l_1^2 + l_2^2/3 + l_1 l_2 c_2) & m_2(l_2^2/3 + l_1 l_2 c_2/2) \\ m_2(l_2^2/3 + l_1 l_2 c_2/2) & m_2 l_2^2/3 \end{pmatrix}$$

$$\mathbf{f} = \begin{pmatrix} -m_1 g l_1 c_1/2 - m_2 g(l_1 c_1 + l_2 c_{12}/2) \\ -m_2 g l_2 c_{12}/2 \end{pmatrix} + \begin{pmatrix} m_2 l_1 l_2 s_2/2(2\dot\theta_1\dot\theta_2 + \dot\theta_2^2) \\ -m_2 l_1 l_2 s_2 \dot\theta_1^2/2 \end{pmatrix}$$

In the following simulation we use the data

$$m_1 = m_2 = 36kg,\ l_1 = l_2 = 1m,\ g = 9.81m/s^2$$
$$\theta_1(0) = 70^o,\ \theta_2(0) = -140^o,\ \dot\theta_1(0) = \dot\theta_2(0) = 0$$

Note that in this example (and the next) $M^{-1}G^T$ does not form a "bad angle" with G. Examples where this is an issue have been examined in [4, 15]. Here we consider other issues.

We examine two choices for a constraint $\mathbf{g}(\mathbf{q}, t)$ on the position of (x_2, y_2).

Case I Consider the case where the coordinates (x_2, y_2) are constrained to lie on a parabola

$$y_2 = x_2^2 - \beta$$

where β is chosen such that the above initial conditions are consistent. Simulations for time up to $t_f = 40s$ have been carried out as follows:

1. using no stabilization (denoted Baum$(0, 0)$),
2. using Baumgarte's method with parameters α_0 and α_1 (denoted Baum(α_1, α_0)),
3. using the stabilization (5.4) with projection on both velocity and position constraint levels utilizing $D = H$ in (4.4) (denoted S-full),
4. using (5.4) with projection on both velocity and position constraint levels utilizing (5.6) (denoted S-both),
5. using the double stabilization step with projection on both velocity and position constraint levels utilizing (5.6) (denoted S-both^2),
6. using (5.4) with projection only on velocity level constraints utilizing $F = G^T(GG^T)^{-1}$ (denoted S-vel), and
7. using (5.4) with projection only on position level constraints utilizing $F = G^T(GG^T)^{-1}$ (denoted S-pos).

Note that the last among the projections above does not give an invariant manifold in the sense of (4.1), (4.2), although it still yields a method which may be interpreted as a good discretization of a stable index-reduced problem.

The path in cartesian coordinates traversed by (x_2, y_2), is depicted in Fig. 6.2.

In Table 6.1 we record the measured drifts based on runs using an explicit Runge-Kutta scheme of order 2 with a constant step size h. In case of a solution blowup we write $*$.

We observe that the stabilizations on both velocity and position levels and the stabilization on velocity level alone yield good results, while stabilization on the position level alone in the sense of (5.4) (i.e. one iteration step) does not yield good results for this example, in agreement with [9]. If we add a few more restricted Newton iterations per step, i.e. we perform a "true" coordinate projection on the position level

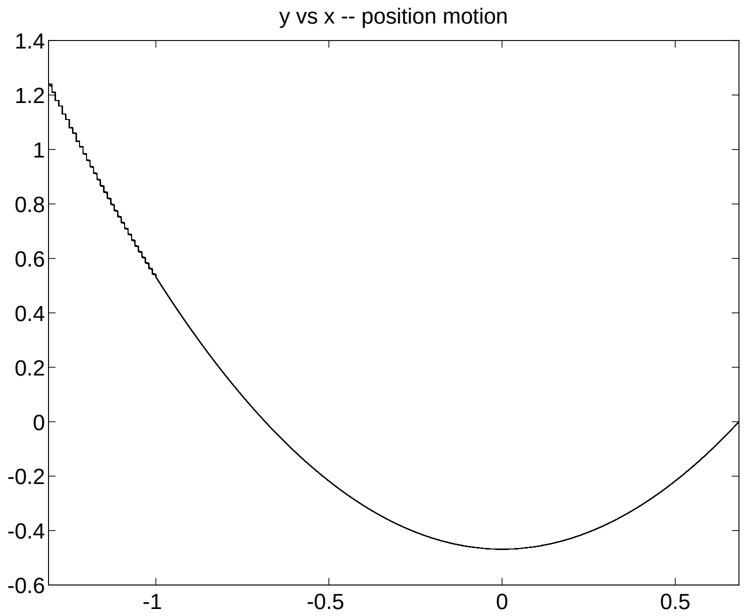

Figure 6.2: Constraint path for (x_2, y_2), Case I

alone, then the drift at the position level essentially vanishes, but the other errors in solution and velocity drift are not improved by much. We emphasize, though, that in various other examples which were tried the projection S-pos proved useful.

Of all these variants the stabilization S-both^2 may be the preferred one, because it cheaply yields a smaller residual on the position level. It costs almost the same as the only-velocity stabilization S-vel, and the residuals are much smaller.

The Baumgarte stabilization performs reasonably well here for $h = .001$. Even without stabilization we obtain decent drift values in this case.

Case II A more difficult case is obtained when the coordinates (x_2, y_2) are constrained to obey

$$y_2 = \sin^2(\omega t)$$

The problem gets tougher the larger the parameter ω becomes. We choose $\omega = \frac{1}{2}$ below. The obtained constrained path for (x_2, y_2) is depicted in Fig. 6.3. In this case the constraint forces become large at a few distinct times.

In Table 6.2 we record the measured drifts based on runs up to $t_f = 10s$ using an explicit Runge-Kutta scheme of order 2 with a constant step size h.

Note that the Baumgarte stabilization is not as effective as the S- stabilizations, especially for the case $h = .01$. Other parameters (α_0, α_1) tried (including $\gamma = 1/h$ in (2.2)) do not yield significantly better results. $\square$

h	stabilization	drift-velocity	drift-position
.01	Baum$(0,0)$	.33e-2	.17e-2
.01	Baum$(12,70)$	.72e-2	.14e-2
.01	S-full	.12e-7	.61e-9
.01	S-both	.16e-3	.39e-9
.01	S-both^2	.67e-8	.15e-13
.01	S-vel	.36e-14	.84e-3
.01	S-pos	$*$	$*$
.001	Baum$(0,0)$	.32e-4	.17e-4
.001	Baum$(12,70)$	.70e-4	.14e-4
.001	S-full	.15e-13	.40e-14
.001	S-both	.16e-6	.39e-14
.001	S-both^2	.18e-13	.31e-14
.001	S-vel	.36e-14	.81e-5
.001	S-pos	.48e-2	.76e-11

Table 6.1: maximum drifts for Case I

Based on these experiments and others we have determined that the stabilization technique S-both^2, i.e., using the double stabilization step with F given by (5.5) or (5.6), is a good compromise between stability and computational expense per step. An experimental code with error control based on the code DOPRI5 of [21], which in turn is based on the Dormand-Prince Runge-Kutta formulae [22], has been implemented – see [6].

Example 2 (cont.)

Using this code we can easily compare the performances of our stabilization technique and Baumgarte's, because the cost of a discretization step is similar and the same ODE integrator is used (the stabilization cost being negligible). In Table 6.3 we use relative local error tolerance $TOL = 10^{-5}$ and take the absolute tolerance to be 0.1$*$TOL. Also, NSTEP denotes the number of time steps (including rejected ones) that the code takes. We integrate case II above for different values of ω up to $t_f = 100s$.

The advantage of our stabilization method is clear. $\Box$

Example 3

A seven-body squeezing mechanism is described in [23] and tested in [24] as well. We have solved this popular example using the same tolerances as in the previous example. The interval of integration is from $t = 0$ to $t_f = .3s$, which makes the problem more challenging than with the value of $t_f = .015s$ taken in the above references. A plot of the solution components (mod 2π) is given in Fig. 6.4.

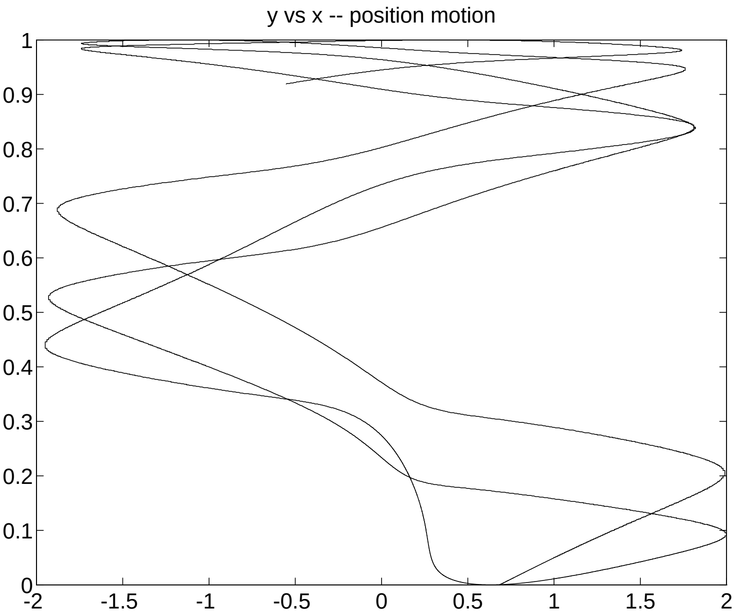

Figure 6.3: Constraint path for (x_2, y_2), Case II

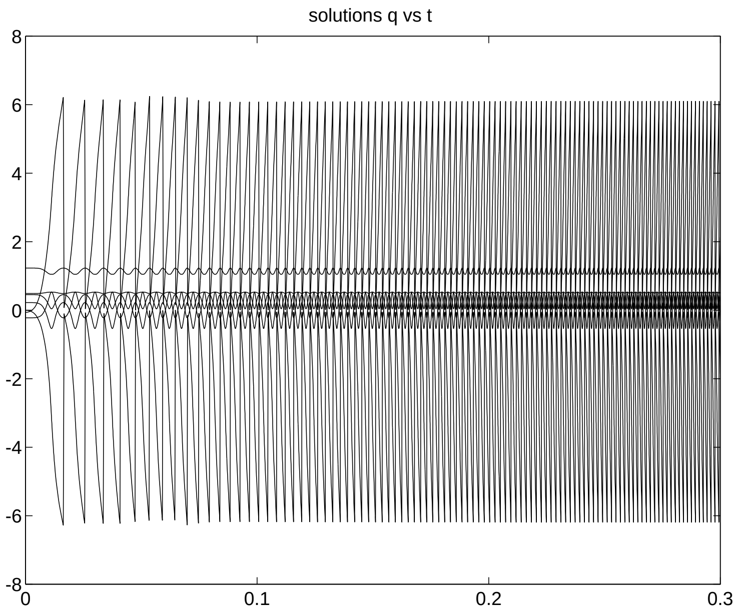

Figure 6.4: Solution components for example 3

h	stabilization	drift-velocity	drift-position
.01	Baum$(0,0)$	$*$	$*$
.01	Baum$(12,70)$	.51	.25e-1
.01	S-full	.72e-5	.63e-6
.01	S-both	.31e-1	.68e-5
.01	S-both^2	.20e-3	.68e-6
.01	S-vel	.60e-14	.78e-2
.01	S-pos	.73	.28e-2
.001	Baum$(0,0)$	.66e-4	.56e-4
.001	Baum$(12,70)$	.60e-3	.38e-4
.001	S-full	.39e-10	.53e-11
.001	S-both	.41e-4	.46e-11
.001	S-both^2	.20e-9	.78e-15
.001	S-vel	.43e-14	.58e-4
.001	S-pos	.44e-2	.88e-10

Table 6.2: maximum drifts for Case II

ω	stabilization	NSTEP	drift-position	drift-velocity
0.5	Baum$(0,0)$	10864	.96	.22e-1
	Baum$(12,70)$	4197	.81e-5	.11e-3
	S-both^2	3767	.66e-10	.17e-6
1.0	Baum$(0,0)$	15408	1.38	.36e-1
	Baum$(12,70)$	12826	.42e-4	.52e-3
	S-both^2	5381	.36e-9	.54e-6

Table 6.3: maximum drifts for Case II using automatic code

In Table 6.4 we list the number of steps taken by the various methods tested as well as the maximum drifts in position and velocity level constraints. While the various variants cost about the same to execute, the maximum drifts are much smaller using our method. Here the Baumgarte parameters have to be taken larger than in the previous example in order to observe a significant effect. Moreover, comparing solution values at $t = t_f$ to those obtained with a smaller tolerance it turns out that the first two entries in Table 6.4 correspond to solutions with an error in their leading digit, despite the much smaller drifts recorded. $\square$

While a full-blown comparison to other general purpose codes like MEXX [19] (see also §VI.9 in [24]) is well beyond the scope of this paper, we have made some preliminary such comparisons for both Examples 2 and 3, in which the code described here fares well. More details are given in [6].

stabilization	NSTEP	drift-position	drift-velocity
Baum$(0,0)$	2872	.93e-3	.11e-1
Baum$(12,70)$	2842	.45e-3	.44e-2
Baum$(200,10000)$	2853	.91e-5	.95e-4
S-both^2	2838	.29e-13	.17e-7

Table 6.4: maximum drifts for seven-body example using automatic code

References

[1] C. Führer and R. Schwertassek, *Generation and solution of multibody system equations*, J. Non-Linear Mechanics 25 (1990), 127-141.

[2] E. Haug and R. Deyo, *Real-Time Integration Methods for Mechanical Sustem Simulation*, NATO ASI Series, Springer 1991.

[3] J. Baumgarte, *Stabilization of constraints and integrals of motion in dynamical systems*, Comp. Math. Appl. Mech. Eng. 1 (1972), 1-16.

[4] U. Ascher and L. Petzold, *Stability of Computational Methods for Constrained Dynamics Systems*, SIAM J. SISC 14 (1993), 95-120.

[5] U. Ascher, H. Chin and S. Reich, *Stabilization of DAEs and invariant manifolds*, Numer. Math., to appear, 1994.

[6] H. Chin, PhD thesis, in preparation.

[7] K. Brenan, S. Campbell and L. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, North-Holland, 1989.

[8] E. Griepentrog and R. März, *Differential-Algebraic Equations and their Numerical Treatment*, Teubner-Texte Math. 88, Teubner, Leipzig, 1986.

[9] T. Alishenas, *Zur numerischen Behandlung, Stabilisierung durch Projektion und Modellierung mechanischer Systeme mit Nebenbedingungen und Invarianten*, PhD thesis, Königliche Technische Hochschule Stockholm, 1992.

[10] R. A. Wehage and E. J. Haug, *Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems*, J. of Mechanical Design 104 (1982), 247-255.

[11] F. A. Potra and W. C. Rheinboldt, *On the numerical solution of the Euler-Lagrange equations*, J. Mechanics of Structures and Machines, to appear.

[12] J. Yen, *Constrained equations of motion in multibody dynamics as ODEs on manifolds*, SIAM J. Numer. Anal. 30 (1993), 553-568.

[13] K. Führer and B. Leimkuhler, *A new class of generalized inverses for the solution of discretized Euler-Lagrange equations*. In: Haug, E., Deyo, R.(ed.): NATO Advanced Research Workshop on Real-Time Integration Methods for Mechanical System Simulation, Snowbird, Utah 1989. Springer, Berlin Heidelberg New York Tokyo(1990)

[14] C. W. Gear, *Maintaining solution invariants in the numerical solution of ODEs*, SIAM J. Sci. Stat. Comp., 7 (1986), 734-743.

[15] U. Ascher and L. Petzold, *Projected collocation for higher-order higher-index differential-algebraic equations*, J. Comp. Appl. Math. 43 (1992), 243-259.

[16] E. Eich, C. Führer, B. Leimkuhler, and S. Reich, *Stabilization and projection methods for multibody dynamics*, Research report, Inst. Math, Helsinki Univ. of Technology, 1990.

[17] L.F. Shampine, *Conservation laws and the numerical solution of ODEs*, Comp. Maths. Appls. 12B (1986), 1287-1296.

[18] E. Eich, *Convergence results for a coordinate projection method applied to mechanical systems with algebraic constraints*, SIAM J. Numer. Anal., to appear.

[19] Ch. Lubich, U. Nowak, U. Pöhle and Ch. Engstler, *MEXX – numerical software for the integration of constrained mechanical multibody systems*, Tech. Rep. SC 92-12, Zuse-Zentrum, Berlin, 1992.

[20] F. Amirouche, *Computational Methods in Multobody Dynamics*, Prentice-Hall, 1992.

[21] E. Hairer S.P. Norsett and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer-Verlag, 1987.

[22] J.R. Dormand and P.J. Prince, *A family of embedded Runge-Kutta formulae*, J. Comp. Appl. Math. vol. 6(1980), p.19-26.

[23] W. Schielen (Ed.), *Multibody Systems Handbook*, Springer, 1990.

[24] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer-Verlag, 1991.