INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 7, Number 2, Pages 240–260

# THE $\alpha$ METHOD FOR SOLVING DIFFERENTIAL ALGEBRAIC INEQUALITY (DAI) SYSTEMS

JOEY PETER, NIGAM CHANDRA PARIDA, AND SOUMYENDU RAHA

(Communicated by A K Pani)

**Abstract.** This paper describes an algorithm for ”direct numerical integration” of the initial value Differential-Algebraic Inequalities (DAI) in a time stepping fashion using a sequential quadratic programming (SQP) method solver for detecting and satisfying active path constraints at each time step. The activation of a path constraint generally increases the condition number of the active discretized differential algebraic equation's (DAE) Jacobian and this difficulty is addressed by a regularization property of the $\alpha$ method. The algorithm is locally stable when index 1 and index 2 active path constraints and bounds are active. Subject to available regularization it is seen to be stable for active index 3 active path constraints in the numerical examples. For the high index active path constraints, the algorithm uses a user-selectable parameter to perturb the smaller singular values of the Jacobian with a view to reducing the condition number so that the simulation can proceed. The algorithm can be used as a relatively cheaper estimation tool for trajectory and control planning and in the context of model predictive control solutions. It can also be used to generate initial guess values of optimization variables used as input to inequality path constrained dynamic optimization problems. The method is illustrated with examples from space vehicle trajectory and robot path planning.

**Key Words.** Differential-algebraic equations, Trajectory planning, Numerical optimization, Inequality path constraints

## 1. Background

Many engineering control problems, especially those with inequality path constraints yield Differential-Algebraic Inequalities (DAI) models. The need for solving a DAI system arises in robotic path planning [19], safety envelope [10] and trajectory [7] generation, in model predictive control approaches [8], and in voltage control of electrical equipments [11]. Traditionally, a DAI is handled, almost always, in the context of optimal control problems where the inequality path constraints in the discretized optimal control problem are handled by the optimizer as inequality constraints at each mesh point (e.g., in Direct Transcription and collocation schemes) or as a sub-interval wise cumulative error integral that is minimized (e.g., in multiple shooting schemes).

However, in certain situations the direct simulation of a DAI does arise. For example, when interior point methods are used for dynamic optimization problems,

Received by the editors November 15, 2008 and, in revised form, July 9, 2009.

2000 *Mathematics Subject Classification.* 65L80, 49J15, 65K99, 65F99.

This research was supported by the IISc-ISRO Space Technology Cell with grant ISTC 162 and ISTC 194.

the initial guess of a feasible solution involving the satisfaction of the DAI system is needed. Intermediate level trajectory planning through constraint programming also entails the necessity of a DAI solver [19].

Available Differential-Algebraic Equation (DAE) solvers are limited to addressing inequalities only in the form of positivity constraints on the states and controls (e.g., DASPK [3]). Some DAE solvers can detect whether a constraint has become active with root-finding techniques (e.g., DASRT [3], DASPKE [13]) and mainly concern with DAE systems with discontinuities.

Examples of DAI integrators are few. A solver with constraint smoothing and local planning has been described in [19]. This algorithm checks for weakly approximate safety condition of using the control values from the previous time step and proceeds with activating a buffer zone as a path inequality constraint becomes nearly active. A barrier function minimization is used if the safety was violated to obtain a new set of initial guesses for the controls at that time step. Then the dynamics is integrated provided the states obtained satisfy the bounds and inequality path constraints.

**1.1. Introduction to the present work.** At every time step a DAI integrator must

- detect active path constraints
- determine which algebraic variables control on to the active path constraints
- handle the possible numerical row rank deficiency in the active constraints Jacobian in the SQP method. The numerical rank deficiency may occur due to activation of high differential index (see definition in [3] and called the index hereafter) active path constraints and due to abrupt changes in the active path constraint set.

The present algorithm addresses the above requirements as follows.

- A standard sequential quadratic programming (SQP) method that is used as a solver at each time step detects the active path constraints.
- The SQP method also determines which algebraic variables control on to the active path constraints by constructing a square basis matrix (defined in section 3.1) which has the least condition number over available column permutations in the active constraints Jacobian in the SQP method.
- The numerical row rank deficiency is addressed by varying a parameter in the DAI discretization which leads to increase in the smallest singular values (i.e., regularization) of the basis matrix.

The DAI solver finds a feasible solution locally in contrast to the Multiple Shooting method or the Direct Transcription method where the dynamic optimization problem discretized over the entire simulation interval enters the QP iterations of an SQP solver. The trade-off for a DAI solver is cheaper computational cost in finding a feasible solution one time step at a time involving much smaller matrices than the dynamic optimization.

The method is intended to be either a cheap tool for generating feasible initial guess for the dynamic optimization problem solved by Multiple Shooting or Direct Transcription with an interior point method, or to be a solver for rapid trajectory planning via constraint programming at an intermediate specification level. It is not meant to replace methods that find solutions over the entire simulation interval, such as the Multiple Shooting or Direct Transcription methods.

The regularization property of the present algorithm is different from the existing approaches of path constraint perturbation and/or modification (such as the buffer zone in [19]) which add additional computational effort including automatic differentiation of the path constraints and/or affect inconsistent activation of path constraints (in case of constraint perturbation).

## 2. The initial value differential-algebraic inequality (DAI) problem

The initial value differential-algebraic inequality (DAI) problem that this paper addresses may be generally stated as

$$\dot{x} \;=\; f(x, u, w(t), t) \text{ (dynamics)} \tag{1a}$$

$$0 \;\geq\; \phi(x, u, t) \text{ (path constraints)} \tag{1b}$$

$$\text{subject to}$$

$$b_{upper} \geq \begin{pmatrix} x \\ u \end{pmatrix} \geq b_{lower} \text{ (bounds) and} \tag{1c}$$

$$x(t_0) = x_0 \text{ (consistent initial conditions)} \tag{1d}$$

where $x \in \mathbb{R}^m$ denotes the differential state variables; $f : \mathbb{R}^m \times \mathbb{R}^{q_c(t)} \times \mathbb{R} \to \mathbb{R}^m$ $u \in \mathbb{R}^{q_c(t)}, q_c(t) \leq m$ is the vector of controls and algebraic state variables (i.e., in general, all the algebraic variables and hereafter referred to as only algebraic variables); and $\phi : \mathbb{R}^m \times \mathbb{R}^{q_c(t)} \times \mathbb{R} \to \mathbb{R}^q$ is the vector of scleronomic and path inequality constraints, hereafter referred to as path constraints only and $t \in [t_0, t_f] \subset \mathbb{R}$. Also, we assume that $r_{active}(t) \leq q_c$ *independent* path constraints are active (hereafter, in the analysis, active path constraints would be used to mean only independent active path constraints) at a $t \in [t_0, t_f]$. Also assumed is $r_{active}(t) \leq m$. Obviously, $r_{active}(t) \leq q$ since the active constraints are a subset of the given inequality constraints. The vector $w(t)$ is a finite dimensional real vector of the algebraic variables which are specified by the user as input functions of time as part of the intermediate level specification of the constraint programming. Over subintervals of $[t_0, t_f]$ depending on the user input, individual algebraic variables may be either a component of $w(t)$ when specified as a function of $t$ or is a component of $u$ when available to the DAI solver as an unknown corresponding to an active constraint in $\phi$. Thus the dimension $q_c$ of $u$ may vary in time.

The Jacobians $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial u}$ are assumed to exist and to be finite in a suitable norm everywhere on the range of values taken by $x$ and $u$ over $[t_0, t_f]$. Let the time interval $[t_0, t_f]$ be partitioned with $N + 1 \in \mathbb{N}$ mesh points $t_0 < t_1 < t_2 \cdots < t_n < t_{n+1} \cdots < t_N = t_f$. We also assume that $x(t_0) = x_0$ is given, and that the algebraic variables in $u(t_0) = u_0$ are either given or are computed with sufficient accuracy satisfying the prescribed initial bounds if any. The initial condition $x_0, u_0$ are consistent with the path constraints active at $t_0$. Further, it is assumed that $\dot{f}(t_0)$, evaluated as $\frac{df}{dt}$ at $t = t_0$, exists and can be computed with the necessary accuracy.

We assume that the path constraints and the bounds do not combine to form a set of infeasible constraints and that a solution ($x(t)$ and $u(t)$) exists everywhere on $[t_0, t_f]$ satisfying the dynamics, path constraints and the bounds (if any) on states and controls. The DAI integrator, when possible, generates the solution sequentially at each mesh point starting from $t_1$ to $t_N$ in the fashion of a numerical integrator.

## 3. Constraint Jacobian in the SQP solver

A full rank square *basis* matrix $\mathcal{B}$ can be formed by permuting the columns of the full row rank matrix (called the *working set* Jacobian) $\mathcal{J}$ [15] of active SQP constraint gradients (rows).

**3.1. Construction of the basis matrix.** All the independent active path constraint gradients (the row vectors in the constraint Jacobian) that should be binding at a mesh point are qualified to the working set Jacobian when the linear independence constraint qualification (LICQ) is satisfied, i.e., the resulting working set Jacobian has full numerical row rank with respect to a computationally significant tolerance. The path constraints in the working set can be represented as $g_i := g(x_i, u_i, t_i) = 0$ (a subset of the constraints in $\phi_i$) at the $i$th mesh point. We can partition (in the sense of [17]) each $g_i$ into $g_{1_i}(x_i, u_i, t_i)$ so that $\frac{\partial g_{1_i}}{\partial \tilde{u}_i}$ is invertible (index 1) at the particular iterate values of $x_i, u_i$; $g_{2_i}(x_i, u_i, t_i)$ so that $\frac{\partial g_{1_i}}{\partial v_i}$ is not invertible but $\frac{\partial g_{2_i}}{\partial x_i}\frac{\partial f_i}{\partial v_i}$ is invertible (i.e., index 2); and $g_{3_i}(x_i, u_i, t_i)$ for index greater than 2. Corresponding to the active path constraint partitions, the algebraic variables are been partitioned into $\tilde{u}_i$ (index 1), $v_i$ (index 2), $\tilde{v}_i$ (index greater than 2) and $\tilde{w_i}$ (rest of the algebraic variables) based on techniques discussed in [17]. For simplicity of analysis, we shall assume that $g_{2_i}(x_i, u_i, t_i)$ and $g_{3_i}(x_i, u_i, t_i)$ are in the form of $g_{2_i}(x_i, t_i)$ and $g_{3_i}(x_i, t_i)$ as is common in most trajectory planning problems. The differential variables $x_i$ along with the algebraic variables $\tilde{u}_i$ (index 1), $v_i$ (index 2), and $\tilde{v}_i$ (index greater than 2) constitute the set of basic variables. The algebraic variables that become basic variables correspond to the permuted columns of the working set Jacobian $\mathcal{J}$ so that the basis matrix has a minimum condition number over all the available permutations of the columns [15]. The variables in $\tilde{w}_i$ make up the set of non-basic and super-basic variables (as in [15]).

We represent $\begin{pmatrix}\frac{\partial g_{1_i}}{\partial x_i}\\ \frac{\partial g_{2_i}}{\partial x_i}\\ \frac{\partial g_{3_i}}{\partial x_i}\end{pmatrix}$ simply as $\frac{\partial g_i}{\partial x_i}$. Similarly we represent $\begin{pmatrix}\frac{\partial f_i}{\partial \tilde{u}_i} & \frac{\partial f_i}{\partial v_i} & \frac{\partial f_i}{\partial \tilde{v}_i}\end{pmatrix}$ as $\frac{\partial f_i}{\partial u_{b_i}}$; and $\frac{\partial g_i}{\partial u_{b_i}}$ represents the Jacobian of $(g_{1_i}^T, g_{2_i}^T, g_{3_i}^T)^T$ with respect to the basic variables $u_{b_i} := (\tilde{u}_i^T, v_i^T, \tilde{v}_i^T)^T$. Also, $\mathsf{g}_i := (g_{2_i}^T, g_{3_i}^T)^T$ is the vector of active path constraints and active bounds with index 2 or higher that must be satisfied at mesh point $i$ and $\mathsf{v}_i := (v_i^T, \tilde{v}_i^T)^T$ is the vector of all index 2 and higher algebraic variables computed at mesh point $i$. For any function $\varphi$, $\varphi(t_i) := \varphi(x(t_i), u(t_i), w(t_i), t_i)$ denotes the analytical value at the mesh point $i$ whereas $\varphi_i$ is the numerically computed value at the same mesh point.

## 4. The $\alpha$ method

The so called generalized-$\alpha$ method which we refer to simply as the $\alpha$ method was originally formulated for integration of stiff second order systems in [5] and was applied for solving the stabilized index 2 DAEs of flexible multibody dynamics in [20]. In this paper, the method (similar to that in [12]) is adopted to the first order DAI systems with inequality algebraic constraints using the same algorithmic parameters as in [5].

**4.1. Discretization of the DAI.** The $\alpha$ method discretization of the DAI problem with first order differential equations is obtained by trivially setting $M = 0$ followed by elimination of the velocity level equations in the formulation found in [5]. Given a feasible solution $x_n, u_n$ and an algorithmic variable $a_n$ at the $n$th mesh point, the DAI solver finds at the $n$th time step a feasible solution $x_{n+1}, u_{n+1}$ and an $a_{n+1}$ using the SQP method such that

$$
\begin{aligned}
x_{n+1} \quad &= \quad x_n + \left(1-\frac{\beta}{\gamma}\right) h_n f(x_n,u_n,w(t_n),t_n) + \frac{\beta}{\gamma} h_n f(x_{n+1},u_{n+1},w(t_{n+1},t_{n+1}) \\
\text{(2a)} \quad & \qquad + \left(\frac{1}{2}-\frac{\beta}{\gamma}\right) h_n^2 a_n \\
a_{n+1} \quad &= \quad \frac{1}{h_n\gamma}\left(f(x_{n+1},u_{n+1},w(t_{n+1},t_{n+1}) - f(x_n,u_n,w(t_n),t_n)\right) \\
\text{(2b)} \quad & \qquad + \left(1-\frac{1}{\gamma}\right) a_n, \\
\text{(2c)} \quad 0 \geq \quad & \phi(x_{n+1},u_{n+1},w(t_{n+1}),t_{n+1}) \\
\text{(2d)} \quad & b_{{upper}\,n+1} \geq \begin{pmatrix} x_{n+1} \\ u_{n+1} \end{pmatrix} \geq b_{{lower}\,n+1}
\end{aligned}
$$

where

$$
\gamma = \frac{2}{\rho+1} - \frac{1}{2} \tag{3a}
$$

$$
\beta = \frac{1}{(\rho+1)^2} \tag{3b}
$$

and $\rho \in [0,1)$ is a user-selectable regularization parameter and $h_n := t_{n+1} - t_n$ is the time step size on the normalized time interval $[0,1]$ (i.e., $t_0 = 0$ and $t_f = 1$). A numerically computed solution $x_n$ and $u_n$ at $t_n$ approximate an analytical feasible solution $x(t_n)$ and $u(t_n)$ which is assumed to exist. It must be remarked that a DAI step (2a-2c) does not generally have a unique numerical solution unlike a DAE with index 2 or lower. The initial condition $a(t_0)$ is evaluated as $\dot{f} = \frac{df}{dt}$ at $t = 0$ with $x(t_0) = x_0$ and $u(t_0) = u_0$. Thus, we use $a_0 = a(t_0) = \dot{f}_0$.

**4.2. Continuity assumption.** In our analysis $u$ is assumed to be Lipschitz continuous everywhere on $[t_0, t_f]$. A smoother continuity assumption would not be realistic for many trajectory planning problems where sharp changes are likely in $u$ . Following [1] the Lipschitz constant for $u$ may be thought to be very large, i.e, as large as needed, but finite whenever there are sharp changes in the values of $u$. Consistently, we assume that $f$ too is Lipschitz continuous in $t$ everywhere on $[t_0, t_f]$. We also assume that $x$ is Lipschitz continuous on $[t_0, t_f]$.

**4.3. Assumption on high index active path constraints.** For simplicity of analysis but without loss of broader applicability and as is common in trajectory planning problems we assume that the high index active path constraints are only in the form of $\mathsf{g}_i(x_i, t_i) = 0$ at the $i$th mesh point.

**4.4. Local error estimate for differential variables.** Let $\mathcal{T}$ be the set of mesh points $\{t_1, \cdots, t_N\}$. For any $\varphi(x,u,t)$ we represent $\varphi(x_n,u_n,t_n)$ as $\varphi_n$ when $x_n,\ u_n$ are numerically computed at $t_n$ and represent $\varphi(x(t_n),u(t_n),t_n)$ as $\varphi(t_n)$ when $x(t_n),\ u(t_n)$ are analytically evaluated at $t_n$. For $0 < h_i < C_h$ where $C_h$ is some finite constant and any function $\varphi(t)$, we define

$$
\|\varphi\|_S \quad := \max_{t_i \in \{\mathcal{T}\backslash t_N\} \cup t_0} \frac{\|\varphi(t_{i+1}) - \varphi(t_i)\|_\infty}{h_i} \tag{4}
$$

and $\Delta f(t_n) := f(t_{n+1}) - f(t_n)$. Let $h := \max_{i\in\{0,\cdots,N-1\}} h_i = \max_{t_i \in t_0 \cup \{\mathcal{T}\backslash t_N\}} |t_{i+1} - t_i|$ be the maximum step size.

**Lemma 1.** *For $f$ being Lipschitz continuous, the local error in $x$, $\varepsilon_L x$, of the $\alpha$-method discretization (2a-2b) is $O(h^2)$ and that of the algorithmic variable $a$ is $O(1)$.*

**Proof.** After solving the recurrence

$$a(t_{n+1}) \quad = \quad \frac{1}{h_n\gamma}\left(f(t_{n+1}) - f(t_n)\right) + \left(1 - \frac{1}{\gamma}\right) a(t_n), \tag{5}$$

the analytical value (at $n$th mesh point) of the algorithmic variable $a$, denoted as $a(t_n)$, can be written as

$$a(t_n) \quad = \quad a_0\left(\frac{\gamma-1}{\gamma}\right)^n + \sum_{j=1}^{n}\frac{(\gamma-1)^{j-1}\gamma^{-j}(f(t_{n+1-j}) - f(t_{n-j}))}{h_{n-j}}, \tag{6}$$

from which it may be seen that

$$\left(a(t_n) - \frac{\Delta f(t_n)}{h_n}\right) = \left(1 - \frac{1}{\gamma}\right)^n a_0 + \frac{1}{\gamma}\sum_{j=1}^{n}\left(1 - \frac{1}{\gamma}\right)^{j-1}\frac{\Delta f(t_{n-j})}{h_{n-j}} - \frac{\Delta f(t_n)}{h_n} \tag{7}$$

Since $\left|1 - \frac{1}{\gamma}\right| < 1$ and $0 \le \left(\frac{1}{2} - \frac{\beta}{\gamma}\right)\frac{\left|\frac{1}{\gamma}\right|}{\left|1-\frac{1}{\gamma}\right|-1} \le \frac{1}{6}$ and $f$ is assumed to be Lipschitz continuous in $t$ on $[0,1]$, the expansion in equation (7) leads us to the estimate that

$$\begin{aligned}
\left|\frac{1}{2} - \frac{\beta}{\gamma}\right| \|\Delta f(t_n) - a(t_n)h\|_\infty \quad &\le \quad \left|\frac{1}{2} - \frac{\beta}{\gamma}\right| \frac{\left|\frac{1}{\gamma}\right|}{1 - \left|1 - \frac{1}{\gamma}\right|}\left(1 - \left|1 - \frac{1}{\gamma}\right|^n\right)\|f\|_S h \\
&\quad + \left|\frac{1}{2} - \frac{\beta}{\gamma}\right|\left(\left|1 - \frac{1}{\gamma}\right|^n (\|f\|_S + \|a_0\|_\infty)h + \|f\|_S h\right) \\
&\le \quad (\|f\|_S + \|a_0\|_\infty)\, h \qquad \text{(8a)} \\
&= O(h). \qquad \text{(8b)}
\end{aligned}$$

The local error in $x$, $\varepsilon_L x$, can be written from considering the error for trapezoidal rule for integration applied to Lipschitz continuous functions [6]:

$$\begin{aligned}
\varepsilon_L x_{n+1} \quad &= \quad \int_{t_n}^{t_{n+1}} f(s)ds - \left(\frac{f(t_n) + f(t_{n+1})}{2}\right) h_n \\
&\quad + \left(\frac{1}{2} - \frac{\beta}{\gamma}\right) h_n \left(\Delta f(t_n) - a(t_n)h_n\right) \qquad \text{(9)}
\end{aligned}$$

and then estimating

$$\begin{aligned}
\|\varepsilon_L x_{n+1}\|_\infty \quad &\le \quad \frac{h^2}{4}\|f\|_S + (\|f\|_S + \|a_0\|_\infty)\, h^2 \qquad \text{(10a)} \\
&= \quad O(h^2). \qquad \text{(10b)}
\end{aligned}$$

The local error in computing $a$ at a mesh point $t = t_{n+1}$ is obtained from getting $a_{n+1} - a(t_n)$ from (2b) and then substituting appropriate terms in (2a).

$$\|\varepsilon_L a_{n+1}\|_\infty \le \frac{\|\varepsilon_L x_{n+1}\|_\infty}{\beta h_n^2} + O(h) = O(1). \tag{11}$$

□

**4.4.1. Remark 1.** The larger error in $a$ is computationally acceptable as long as $x$ convergences to a feasible solution since $a$ is only an algorithmic variable and does not approximate any physical quantity.

**4.4.2. Remark 2 (*A-stability*).** The one dimensional test equation $\dot{y} = \lambda y$ (with $\lambda \in \mathbb{C}$) when discretized with the $\alpha$-method equations (2a-2b) yields the amplification matrix $\hat{A}$ so that $\left(y_{n+1}\ h^2 a_{n+1}\right)^T = \hat{A}\left(y_n\ h^2 a_n\right)^T$ holds at the $n$th time step. It may be verified that

$$\hat{A} = \begin{pmatrix} -\frac{2\Omega^2}{2\Omega+(\rho-3)(\rho+1)} + \Omega + 1 & \frac{(\rho-1)^2}{2(2\Omega+(\rho-3)(\rho+1))} \\ -\frac{2\Omega^2(\rho+1)^2}{2\Omega+(\rho-3)(\rho+1)} & \frac{-(\Omega-3)\rho^2-2(\Omega-1)\rho+\Omega-1}{2\Omega+(\rho-3)(\rho+1)} \end{pmatrix} \tag{12}$$

where $\Omega := \lambda h$. The magnitude of the eigenvalues of $\hat{A}$ is given by

$$\left| -\frac{2(\Omega-\rho)\rho \pm \sqrt{-(2\Omega(\rho-1)-\rho-1)(\rho+1)^3}+2}{2\Omega+(\rho-3)(\rho+1)} \right| \leq 1$$

for all $\Omega \in \mathbb{C}^-$. This can be seen by taking limits as $\Omega \to 0$ and as $\Re\Omega \to -\infty, |\Im\Omega| \to \infty$ and observing that the magnitude of the eigenvalues grows smoothly in $[0,1]$ on $\mathbb{C}^-$ for $\rho \in [0,1)$. The amplification matrix is then strictly contractive for $\rho \in [0,1)$ and $\Omega \in \mathbb{C}^-$ (by Theorem IV.11.2 in [9]). Hence the discretization (2a-2b) is $A$-stable. When a feasible solution can be found such that no path and bounds constraints become active, the $\alpha$-method acts as an $A$-stable ODE integrator over successive time steps.

**4.5. Active DAE.** Let there be $r_{active}(t_{n+1}) = r_a$ independent path and bounds constraints $g_{n+1} = 0$ active at the $n$th time step. The equations $g_{n+1}(x_{n+1}, u_{n+1}, t_{n+1}) = 0$ must be satisfied at the $n+1$th mesh point along with the corresponding algebraic basic variables $u_{bn+1}$. The SQP method must satisfy the following discretized DAE.

$$\begin{aligned} x_{n+1} &= x_n + \left(1-\frac{\beta}{\gamma}\right) h_n f(x_n, u_n, t_n) + \frac{\beta}{\gamma} h_n f(x_{n+1}, u_{n+1}, t_{n+1}) \\ &\quad + \left(\frac{1}{2}-\frac{\beta}{\gamma}\right) h_n^2 a_n \end{aligned} \tag{13a}$$

$$\begin{aligned} a_{n+1} &= \frac{1}{h_n\gamma}\left(f(x_{n+1}, u_{n+1}, w(t_{n+1}), t_{n+1}) - f(x_n, u_n, w(t_n), t_n)\right) \\ &\quad + \left(1-\frac{1}{\gamma}\right) a_n \end{aligned} \tag{13b}$$

$$0 = g(x_{n+1}, u_{n+1}, t_{n+1}) \tag{13c}$$

at the $n$th time step. Any active bounds in (2d) are included in (13c).

As described in section 3.1, the basis matrix $\mathcal{B}$ for the discretized DAI problem as in (13) can be constructed as follows.

$$\begin{aligned}
\mathcal{B} &:= \begin{pmatrix} -I + \frac{\beta}{\gamma}\frac{\partial f_i}{\partial x_i}h_{i-1} & 0 & \frac{\beta}{\gamma}\frac{\partial f_i}{\partial u_{b\,i}}h_{i-1} \\ \frac{1}{\gamma h_{i-1}}\frac{\partial f_i}{\partial x_i} & -I & \frac{1}{\gamma h_{i-1}}\frac{\partial f_i}{\partial u_{b\,i}} \\ \frac{\partial g_i}{\partial x_i} & 0 & \frac{\partial g_i}{\partial u_{b\,i}} \end{pmatrix} \\
&:= \begin{pmatrix} -I + \frac{\beta}{\gamma}\frac{\partial f_i}{\partial x_i}h_{i-1} & 0 & \frac{\beta}{\gamma}\frac{\partial f_i}{\partial u_{b\,i}}h_{i-1} \\ \frac{1}{\gamma h_{i-1}}\frac{\partial f_i}{\partial x_i} & -I & \frac{1}{\gamma h_{i-1}}\frac{\partial f_i}{\partial u_{b\,i}} \\ \frac{\partial g_{1\,i}}{\partial x_i} & 0 & \begin{pmatrix} \frac{\partial g_{1\,i}}{\partial \tilde{u}_i} & \frac{\partial g_{1\,i}}{\partial v_i} & \frac{\partial g_{1\,i}}{\partial \tilde{v}_i} \end{pmatrix} \\ \frac{\partial g_{2\,i}}{\partial x_i} & 0 & \begin{pmatrix} \frac{\partial g_{2\,i}}{\partial \tilde{u}_i} & \frac{\partial g_{2\,i}}{\partial v_i} & \frac{\partial g_{2\,i}}{\partial \tilde{v}_i} \end{pmatrix} \\ \frac{\partial g_{3\,i}}{\partial x_i} & 0 & \begin{pmatrix} \frac{\partial g_{3\,i}}{\partial \tilde{u}_i} & \frac{\partial g_{3\,i}}{\partial v_i} & \frac{\partial g_{3\,i}}{\partial \tilde{v}_i} \end{pmatrix} \end{pmatrix} \\
&= \begin{pmatrix} -I + \frac{\beta}{\gamma}\frac{\partial f_i}{\partial x_i}h_{i-1} & 0 & \frac{\beta}{\gamma}\frac{\partial f_i}{\partial u_{b\,i}}h_{i-1} \\ \frac{1}{\gamma h_{i-1}}\frac{\partial f_i}{\partial x_i} & -I & \frac{1}{\gamma h_{i-1}}\frac{\partial f_i}{\partial u_{b\,i}} \\ \frac{\partial g_{1\,i}}{\partial x_i} & 0 & \begin{pmatrix} \frac{\partial g_{1\,i}}{\partial \tilde{u}_i} & \frac{\partial g_{1\,i}}{\partial v_i} & \frac{\partial g_{1\,i}}{\partial \tilde{v}_i} \end{pmatrix} \\ \frac{\partial g_{2\,i}}{\partial x_i} & 0 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ \frac{\partial g_{3\,i}}{\partial x_i} & 0 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \end{pmatrix} \qquad (14)
\end{aligned}$$

corresponding to an ordering $(x_i^T, a_i^T, {u_b}_i^T)^T$ of the differential, algorithmic and algebraic variables at the $i$th mesh point. The basis matrix $\mathcal{B}$ in (14) is essentially the Jacobian of the active DAE system discretized with the $\alpha$ method. The working set Jacobian is then $\mathcal{J} := \begin{pmatrix} \mathcal{B} & \mathcal{N}_S \end{pmatrix}$ in the column-block partitioned representation and $\mathcal{N}_S$ represents the columns corresponding to all the super-basic and non-basic variables.

### 4.6. Local error of index 1 algebraic basic variables.

**Lemma 2.** *If $f$ is Lipschitz continuous, then the local error in index 1 algebraic basic variables for the discretization (13) is $K_v\|\varepsilon_L v_{n+1}\|_\infty + K_{\tilde{v}}\|\varepsilon_L \tilde{v}_{n+1}\|_\infty + O(h^2)$.*

**Proof.** Let the active index 1 path and bounds constraints ${g_1}_{n+1}(x_{n+1}, u_{n+1}, t_{n+1}) = 0$ be solved for $\tilde{u}_{n+1}$ with at most $O(h^3)$ error at the $n$th time step, where $O(h^2)$ is the local error in updating $x$. Since $\frac{\partial {g_1}_{n+1}}{\partial \tilde{u}_{n+1}}$ is invertible, then by the implicit function theorem, one can introduce a unique linear function $G_1(x, v, \tilde{v}, t)$ for index 1 $\tilde{u}$ variables such that

$$\tilde{u} = G_1(x(t), v(t), \tilde{v}(t), t) + O(\|\delta x\|^2) + O(\|\delta v\|^2) + O(\|\delta \tilde{v}\|^2) + O(h^{p+1}) \qquad (15)$$

over $[t_n, t_{n+1}]$. The errors in linearization $\|\delta x\|, \|\delta v\|$ and $\|\delta \tilde{v}\|$ are $O(h)$ because of Lipschitz continuity assumption. The local error in index 1 algebraic variables is estimated as $\|\varepsilon_L \tilde{u}_{n+1}\|_\infty \leq K_x\|\varepsilon_L x_{n+1}\|_\infty + K_v\|\varepsilon_L v_{n+1}\|_\infty + K_{\tilde{v}}\|\varepsilon_L \tilde{v}_{n+1}\|_\infty + K_1 h^2 + K_2 h^3$by Lemma 1, where the $K$'s are suitable constants independent of $h$. In case of the index 1 constraints of the form $g_1(x_{n+1}, \tilde{u}_{n+1}, t_{n+1}) = 0$, $\|\varepsilon_L \tilde{u}_{i+1}\|_\infty \leq K_{\tilde{u}} h^2$ since by Lemma 1 the local error in $x$ is $O(h^2)$. □

**4.7. Numerical rank of $\mathcal{B}$ and the need for regularization.** In the analysis that follows we assume $\mathcal{B}$ as in (14) to be invertible at every time step. If the local index of the active DAE is less than or equal to 2, $\mathcal{B}$ would be well-conditioned enough to be inverted. For index greater than 2, $\mathcal{B}$'s condition number increases as $h^{-\nu}$, $\nu$ being the local index of the active DAE (see section 5.4.2 in [3]). This leads to numerical rank deficiency of $\mathcal{B}$ and to failure of the SQP convergence. Jumps and sharp changes in the active path and bounds constraints set also causes ill-conditioning. When an ill-conditioned basis matrix is detected, the present DAI solver tries to regularize $\mathcal{B}$ by varying $\rho$ in (2). Under certain conditions, a regularized and invertible $\mathcal{B}$ can be obtained. We discuss the regularization property in section 6.

### 4.8. Local error of the higher index algebraic basic variables.

**Lemma 3.** *If $\mathcal{B}$ in (14) is full rank and $f, x$ are Lipschitz continuous in $t$, then the local error of the index $> 1$ the suitably scaled algebraic basic variables for the discretization (13) is $O(h)$.*

**Proof.** Let $\mathsf{v}_n := \begin{pmatrix} v_n^T & \tilde{v}_n^T \end{pmatrix}^T$ and $\mathsf{g}_n := ({g_2}_n^T, {g_3}_n^T)^T$. We introduce the following matrices:

$$\mathcal{A}_n := \left(\frac{\partial f}{\partial x}\right)_n \tag{16a}$$

$$\mathcal{M}_n := \left(I - \frac{\beta}{\gamma} h_n \mathcal{A}_n\right)^{-1} \tag{16b}$$

$$\mathcal{L}_n := I + \left(1 - \frac{\beta}{\gamma}\right) h_n \mathcal{A}_n \tag{16c}$$

$$\mathcal{D}_n := \left(\frac{\partial f}{\partial \mathsf{v}}\right)_n \tag{16d}$$

$$\mathcal{C}_n := \left(\frac{\partial \mathsf{g}}{\partial x}\right)_n \tag{16e}$$

$$\mathcal{Q}_n := \left(\mathcal{C}_n \mathcal{M}_n \mathcal{D}_n\right)^{-1} \mathcal{C}_n \mathcal{M}_n \tag{16f}$$

$$\mathcal{P}_n := I - \mathcal{D}_n \mathcal{Q}_n \qquad \text{(projection)} \tag{16g}$$

where $f$ is redefined as $f := f(x, G_1(x, \mathsf{v}, t), \mathsf{v}, w(t), t)$ after eliminating the index 1 algebraic variables as in the proof of Lemma 2. Let total $r \leq r_{active}$ path and bounds constraints be active with index greater than 1 at the $n$th time step. Then $\mathcal{D}_n \in \mathbb{R}^{m \times r}$ is formed by permuting $r$ appropriate columns of $\frac{\partial f}{\partial u}$ (cf. the construction in section 3.1). The matrix $\mathcal{C}_n$ is formed by selecting the $r$ rows of $\frac{\partial \phi}{\partial u}$ corresponding to the active high index path and bounds constraints $\mathsf{g}_{n+1} = 0$. Accordingly, $\mathsf{v}_n$ corresponds to the columns of $\mathcal{D}_n$ and are those algebraic basic variables that control onto the active path and bounds constraints (At every time step algebraic variables constituting $\mathsf{v}$ may change according to active path constraints.). Since $\mathcal{B}$ is assumed to be full rank $\mathcal{C}_n \mathcal{M}_n \mathcal{D}_n$ is invertible too. This allows us to construct the projection $\mathcal{P}_n$. It may be remarked that if the active path and bounds constraints are only index 2, then $\mathcal{P}_n = I - \mathcal{D}_n(\mathcal{C}_n \mathcal{D}_n)^{-1} \mathcal{C}_n + O(h)$ since $\mathcal{C}_n \mathcal{D}_n$ is invertible for index 2 active DAE. For index 1 and 0 active DAE, obviously $\mathcal{P}_n = I$; $\mathcal{Q}_n = 0$.

From (13), a perturbation $\delta x_{n+1}$ to the solution evaluated at the $n+1$th mesh point yields $\delta x_{n+1} = \frac{\beta}{\gamma} h_n \delta f_{n+1}$ subject to $\mathsf{g}_{n+1}(x_{n+1} + \delta x_{n+1}, t_{n+1}) = 0$, from which with proper projections we obtain

$$\delta f_{n+1} = \left(\mathcal{P}_{n+1} \mathcal{A}_{n+1} + \frac{\gamma}{\beta h_{n-1}} (I - \mathcal{P}_{n+1})\right) \delta x_{n+1} + O(\|\delta x_{n+1}\|^2) \tag{17a}$$

leading to

$$\mathcal{D}_{n+1} \delta \mathsf{v}_{n+1} = \left(\frac{\gamma}{\beta h_n} \mathcal{D}_{n+1} \mathcal{Q}_{n+1} - \mathcal{D}_{n+1} \mathcal{Q}_{n+1} \mathcal{A}_{n+1}\right) \delta x_{n+1} + O(\|\delta x_{n+1}\|^2) \tag{17b}$$

where $\|\delta x_{n+1}\|^2 = O(h^2)$ because of Lipschitz continuity of $x$. Letting $\delta x_{n+1} = \varepsilon_L x_{n+1}$, and taking norms, we get the estimate that

$$\|\mathcal{D}_{n+1}\varepsilon_L \mathsf{v}_{n+1}\|_\infty \leq K\|\mathcal{D}_{n+1}\varepsilon_L x_{n+1}\|_\infty / h_n + O(h^2) \leq K_1 h \tag{18}$$

using Lemma 1. □

## 5. Stability of the DAI solver

**Definition 1** (Stability of the (linearized) DAI solver)**.** *We define a (linearized) DAI solver to be stable if the error (with respect to the closest feasible analytical solution) in differential and algebraic variables, $\varepsilon y_n$, at the nth (even as $n \to \infty$, $n < \infty$) mesh point does not grow over the nth time step, i.e., the map $\varepsilon y_n \to \varepsilon y_{n+1}$ is contractive when the locally active DAE is stable.*

In the following theorem we show the stability of the present discretization for a linearized DAI problem.

**Theorem 1.** *For a sufficiently small $h$, let the DAI discretization (2) at the nth time step ($n = 0, \cdots, N-1$) induce an active DAE discretized as in (13) and is locally linearized at $t = t_n$ as in (16). Assume that $\mathcal{B}$ as in (14) has full numerical rank with respect to a computationally suitable cut-off tolerance at every time step and that the SQP method satisfying the locally linearized discretization (2) converges to a feasible solution $(x_n, u_n)$ with an equality constraint satisfaction tolerance of $O(h^3)$ or less at every mesh point $n = 1, \cdots, N$.*

- *(a) Then the $\alpha$-method DAI solver is stable for $\rho \in [0, 1)$ even as $N \to \infty, N < \infty$ given that at every time step $n$ where $n = 0, \cdots, N$, $\Lambda(\mathcal{P}_n\mathcal{A}_n) \in \mathbb{C}^-$, $\Lambda$ being the spectrum of $\mathcal{P}_n\mathcal{A}_n$.*

- *(b) Assume that at least one analytical feasible solution to the mesh point wise linearized DAI problem (1) exist over the normalized simulation interval $[0, 1]$ such that it satisfies exactly the same active path and bounds constraints as satisfied by the numerical solutions $x_n, u_n$ at every mesh point $n = 1, \cdots, N$. Let $x(t), u(t)$ be such an analytical solution. Then for $f, x$ being Lipschitz continuous in $t$, $\|x_n - x(t_n)\| = O(h^2)$ and $\|u_n - u(t_n)\| = O(h)$ at every mesh point $n = 1, \cdots, N$.*

**Proof.** For $h \to 0, h > 0$, the SQP solver updates $x, u$ at the $n$th time step as

$$\left(\begin{pmatrix} x_{n+1} \\ u_{bn+1} \end{pmatrix} \atop \tilde{w}_{n+1}\right) = \begin{pmatrix} \mathcal{B}^{-1} & -\mathcal{B}^{-1}\mathcal{N}_S \\ 0 & I \end{pmatrix} \left(\begin{pmatrix} x_n \\ u_{bn} \end{pmatrix} \atop \tilde{w}_n\right) \tag{19}$$

leading to the following after some tedious manipulation:

$$x_{n+1} = \mathcal{M}_n\mathcal{P}_n\mathcal{L}_n x_n + \left(\frac{1}{2} - \frac{\beta}{\gamma}\right) h_n^2 \mathcal{M}_n\mathcal{P}_n a_n + \mathcal{F}_x(\tilde{w}_n) \tag{20a}$$

$$\mathcal{D}_n\mathsf{v}_{n+1} = \mathcal{D}_n(-\frac{\gamma}{\beta h_n}\mathcal{Q}_n\mathcal{L}_n x_n + \left(1 - \frac{\gamma}{\beta}\right)\mathsf{v}_n + h_n\left(1 - \frac{\gamma}{2\beta}\right)\mathcal{Q}_n a_n) + \mathcal{F}_u(\tilde{w}_n) \tag{20b}$$

$$\begin{aligned} a_{n+1} = {} & \frac{1}{h_n^2}\left(\frac{h_n}{\gamma}\mathcal{A}_n\mathcal{M}_n\mathcal{P}_n\mathcal{L}_n - \frac{1}{\beta}(I - \mathcal{P}_n)\mathcal{L}_n - \frac{h_n}{\gamma}\mathcal{A}_n\right) x_n - \frac{1}{\beta h_n}\mathcal{D}_n\mathsf{v}_n \\ & +(\frac{h_n}{\gamma}\left(\frac{1}{2} - \frac{\beta}{\gamma}\right)\mathcal{A}_n\mathcal{M}_n\mathcal{P}_n - \left(\frac{1}{\gamma} - \frac{1}{2\beta}\right)\mathcal{P}_n + \left(1 - \frac{1}{2\beta}\right) I)a_n \\ & +\mathcal{F}_a(\tilde{w}_n), \end{aligned} \tag{20c}$$

where $\mathcal{F}_x, \mathcal{F}_u, \mathcal{F}_a$ are linear vector functions of super-basic and nonbasic algebraic variables which do not get updated.

From the updates (20) and using (6) and (17) and after some more tedious manipulations, the global error in $x$ at $n+1$th mesh point, denoted by $\varepsilon x_{n+1} := x(t_{n+1}) - x_{n+1}$, can be written as

$$\begin{aligned} \varepsilon x_{n+1} &= \mathcal{M}_n\mathcal{P}_n\mathcal{M}_n^{-1}\varepsilon x_n + \left(1-\frac{\beta}{\gamma}-\frac{\beta h_n}{\gamma^2 h_{n-1}}+\frac{h_n}{2\gamma h_{n-1}}\right) h_n \mathcal{M}_n\mathcal{P}_n\mathcal{A}_n\varepsilon x_n \\ &\quad -\left(\frac{1}{2}-\frac{\beta}{\gamma}\right)\frac{h_n^2}{\gamma^2}\sum_{j=1}^{n-1}\left(1-\frac{1}{\gamma}\right)^{j-1}\frac{(\gamma-1)h_{n-j}-\gamma h_{n-j-1}}{h_{n-j}h_{n-j-1}} \\ &\quad \times \mathcal{M}_n\mathcal{P}_n\left(\mathcal{P}_{n-j}\mathcal{A}_{n-j}+\frac{\gamma}{\beta h_{n-j-1}}(I-\mathcal{P}_{n-j})\right)\varepsilon x_{n-j} \\ &\quad + Ch_n^2 + O(h_n^3) \end{aligned} \tag{21}$$

for an equality constraint satisfaction tolerance of $O(h^3)$ in the SQP method and for a local error in $x$ of $O(h^2)$ (as obtained in Lemma 1 and $C$ is a suitable constant independent of $h_n$). The super-basic and nonbasic variable errors do not enter the projected equation since they do not control on to any active constraints in the current time step and the projection of their errors on to $x$ is zero.

At $n$th time step let exactly $r$ (as in proof of Lemma 3) path and bounds constraints with index greater than 1 be active. Therefore $\mathcal{P}_n \in \mathbb{R}^{m\times m}$ can be decomposed as $U_n\tilde{I}_nU_n^T$ such that $U_n^T\mathcal{D}_n = 0$ in which $U_n \in \mathbb{R}^{m\times m}$ is a unitary matrix. The matrix $\tilde{I}_n$ is constructed from an $m \times m$ identity matrix by replacing $r$ diagonal entries in the columns of basic algebraic variables (corresponding to the high index active path and bounds constraints) with zeros.

In the recurrence (21) the coefficient amplification matrices of the form

$$R_1 := \mathcal{M}_n\mathcal{P}_n\left(\mathcal{M}_n^{-1}+\left(1-\frac{\beta}{\gamma}-\frac{\beta h_n}{\gamma^2 h_{n-1}}+\frac{h_n}{2\gamma h_{n-1}}\right)\mathcal{A}_n h_n\right), \tag{22a}$$

$$R_2 := \left(\frac{1}{2}-\frac{\beta}{\gamma}\right)\frac{h}{\gamma^2}\left(1-\frac{1}{\gamma}\right)^{j-1}\mathcal{M}_n\mathcal{P}_n\mathcal{P}_{n-j}\mathcal{A}_{n-j} \quad \text{and} \tag{22b}$$

$$R_3 := \left(\frac{1}{2}-\frac{\beta}{\gamma}\right)\frac{1}{\gamma^2}\left(1-\frac{1}{\gamma}\right)^{j-1}\frac{\gamma}{\beta}\mathcal{M}_n\mathcal{P}_n(I-\mathcal{P}_{n-j})) \tag{22c}$$

amplify the error. To show that the DAI solver is stable, $R_1$, $R_2$ and $R_3$ need to be contractive so that the local errors do not grow with successive time steps.

Let $QT_AQ^H$ be the Schur Decomposition of $\mathcal{A}_n$. Then $\mathcal{M}_n$, by its construction in (16), has a Schur decomposition of the form $QT_1Q^H$. Similarly

$$\left(\mathcal{M}_n^{-1}+\left(1-\frac{\beta}{\gamma}-\frac{\beta h_n}{\gamma^2 h_{n-1}}+\frac{h_n}{2\gamma h_{n-1}}\right)\mathcal{A}_n h_n\right)$$

can be written in the form of $QT_2Q^H$. In the above decompositions $T_A, T_1, T_2$ are upper triangular matrices. Then, due to the projection $\mathcal{P}_n$, $R_1$ will have the same spectral radius as that of a matrix that has at most $m-r$ non-zero eigenvalues whose magnitudes are upper bounded by the maximum magnitude of the diagonal entries of $T_2T_1$. This means, for an eigenvalue $\lambda$ in the spectrum of $\mathcal{P}_n\mathcal{A}_n$ which is in $\mathbb{C}^-$, the spectral radius of $R_1$ expressed as

$$\left|\frac{1+\left(1-\frac{\beta}{\gamma}-\frac{\beta}{\gamma^2}+\frac{1}{2\gamma}\right)\Omega}{1-\frac{\beta}{\gamma}\Omega}\right| \leq 1$$

for $\rho \in [0,1)$ $\forall$ $\lambda h_n =: \Omega$. The upper bound on the spectral radius can be verified by observing that there are no poles for the eigenvalues on $\mathbb{C}^-$ and by taking the limit $\Re\Omega \to -\infty, |\Im\Omega| \to \infty$. Directly making the surface plots over $\Omega \in \mathbb{C}^-$ for various $\rho$'s also verifies the bound. Then $R_1$ does not amplify the error it operates on and is strictly contractive (by Theorem IV.11.2 in [9]).

Since $\|\mathcal{P}_n\|_2 = 1$, the spectral radius of $R_2$ is same as that of a matrix whose eigenvalues are upper bounded by the maximum magnitude of the diagonal entries of $T_A T_1$, i.e.,

$$\left|\frac{1}{2} - \frac{\beta}{\gamma}\right| \left|\frac{1}{\gamma^2}\right| \left|1 - \frac{1}{\gamma}\right|^{j-1} \left|\frac{\Omega}{1 - \frac{\beta}{\gamma}\Omega}\right| \leq \frac{2}{25}$$

(where $j \in \{1, \cdots, n-1\}$) for all $\lambda_j h =: \Omega \in \mathbb{C}^-$, $\lambda_j \in \Lambda(\mathcal{P}_{n-j}\mathcal{A}_{n-j}) \in \mathbb{C}^-$. (The upper bound on the spectral radius can be easily checked simply by making surface plots). Then, $R_2$ too is strictly contractive by Theorem IV.11.2 in [9].

Replacing $A_{n-j}$ with $I$ and $\mathcal{P}_{n-j}$ with $I - \mathcal{P}_{n-j}$ in $R_2$ and arguing in a similar fashion it can be seen that the spectral radius of $R_3$ is upper bounded by 0.06.

In the error recurrence (21) consider the coefficients under the summation sign. As $n+1 = N \to \infty, N < \infty$, the error from each of the mesh points $N-2, \cdots, 1$ is multiplied by a factor of at most

$$\max\left\{ \frac{\left|\frac{1}{2} - \frac{\beta}{\gamma}\right| \left(\lim_{N\to\infty} \frac{\left|1-\frac{1}{\gamma}\right|^N - \left|1-\frac{1}{\gamma}\right|}{\left|1-\frac{1}{\gamma}\right| - 1}\right)}{|\gamma|^2}, \frac{\left|\frac{1}{2} - \frac{\beta}{\gamma}\right| \left(\lim_{N\to\infty} \frac{\left|1-\frac{1}{\gamma}\right|^N - \left|1-\frac{1}{\gamma}\right|}{\left|1-\frac{1}{\gamma}\right| - 1}\right)}{|\gamma||\beta|} \right\} \|R_3\|.$$

But,

$$\frac{\left|\frac{1}{2} - \frac{\beta}{\gamma}\right| \left(\lim_{N\to\infty} \frac{\left|1-\frac{1}{\gamma}\right|^N - \left|1-\frac{1}{\gamma}\right|}{\left|1-\frac{1}{\gamma}\right| - 1}\right)}{|\gamma|^2} \leq \frac{1}{25}$$

and

$$\frac{\left|\frac{1}{2} - \frac{\beta}{\gamma}\right| \left(\lim_{N\to\infty} \frac{\left|1-\frac{1}{\gamma}\right|^N - \left|1-\frac{1}{\gamma}\right|}{\left|1-\frac{1}{\gamma}\right| - 1}\right)}{|\gamma||\beta|} \leq \frac{1}{16}$$

for $\rho \in [0,1)$. Then, as $N \to \infty, N < \infty$, the errors from the mesh points $N - 2, \cdots, 1$ does not grow, i.e., are not amplified. But we have already established that $R_1$ which amplifies error from $N-1$th mesh point is contractive. Since $R_2, R_3$ are strictly contractive too, we have overall strict contractivity for the map $\varepsilon x_n \to \varepsilon x_{n+1}$ for $\rho \in [0,1)$ and for all $\Lambda(\mathcal{P}_i\mathcal{A}_i) \in \mathbb{C}^-$, $i = 0, \cdots, n$ at the $n$th time step even when $n$ is very large but finite.

Scaling (20b) by $\mathcal{D}_n$ we estimate the error in $\mathsf{v}$, $\varepsilon\mathsf{v}_{n+1}$. We can express $\varepsilon a_n$ in terms of $\varepsilon x_{n-j}, j = 0, \cdots, n$ using (6) and (17). We have already established that at any mesh point $i$, $\varepsilon x_i$ does not grow over the time steps. The amplification matrix of $\varepsilon a_n$, $\mathcal{D}_n\mathcal{Q}_n$ in the scaled (20b), is a projection matrix while $|1 - \gamma/(2\beta)| \leq 1/4$ for $\rho \in [0,1)$. The amplification matrix of $\varepsilon\mathsf{v}_n$ is strictly contractive since $|1-\gamma/\beta| < 1$ for $\rho \in [0,1)$. Hence the map $\varepsilon\mathsf{v}_n \to \varepsilon\mathsf{v}_{n+1}$ is strictly contractive for the same conditions as for the map $\varepsilon x_n \to \varepsilon x_{n+1}$.

In (20c), following a similar approach as above the spectral radius of the matrix

$$(\frac{h_n}{\gamma}\left(\frac{1}{2} - \frac{\beta}{\gamma}\right)\mathcal{A}_n\mathcal{M}_n\mathcal{P}_n - \left(\frac{1}{\gamma} - \frac{1}{2\beta}\right)\mathcal{P}_n + \left(1 - \frac{1}{2\beta}\right)I)$$

can be shown to be upper bounded by

$$\left|\frac{\frac{1}{2}-\frac{\beta}{\gamma}}{\gamma}\right|\left|\frac{\Omega}{1-\frac{\beta}{\gamma}\Omega}\right|+\left|\frac{1}{\gamma}-\frac{1}{2\beta}\right|+\left|1-\frac{1}{2\beta}\right|<1\;\forall\;\lambda h_n=:\Omega\in\mathbb{C}^-;\;\lambda\in\Lambda(\mathcal{P}_n\mathcal{A}_n)\in\mathbb{C}^-$$

for $\rho\in[0,1)$. The matrix above is the amplification matrix for $\varepsilon a_n$. We have already established that $\varepsilon x_n\to\varepsilon x_{n+1}$ and $\varepsilon \mathsf{v}_n\to\varepsilon \mathsf{v}_{n+1}$ are strictly contractive. Then the map $\varepsilon a_n\to\varepsilon a_{n+1}$ is strictly contractive for the same conditions as for the map $\varepsilon x_n\to\varepsilon x_{n+1}$.

The DAI solver is then stable as defined in Definition 1. This completes the proof of item (a).

To establish item (b) we use induction. Let $\|x(t_n)-x_n\|=O(h^2)$ and $\|\mathsf{v}(t_n)-\mathsf{v}_n\|=O(h)$. From Lemmas 1, 2 and 3, the error at the first mesh point is $\|x(t_1)-x_1\|=O(h^2)$ and $\|u(t_1)-u_1\|=O(h)$. Because of strict contractivity of the maps $\varepsilon x_n\to\varepsilon x_{n+1}$ and $\varepsilon\mathsf{v}_n\to\varepsilon\mathsf{v}_{n+1}$, we have $\|x(t_{n+1})-x_{n+1}\|=O(\|x(t_n)-x_n\|)+O(h^2)=O(h^2)$ and $\|\mathsf{v}(t_{n+1})-\mathsf{v}_{n+1}\|=O(\|\mathsf{v}(t_n)-\mathsf{v}_n\|)+O(h)=O(h)$ which leads to the overall estimate $\|u(t_{n+1})-u_{n+1}\|=O(h)$. This establishes item (b). $\square$

**5.1. Remarks.** In Theorem 1 the results are mesh point wise and unlike an estimate over the entire simulation interval as is commonly obtained for a DAE or an optimal control problem, since the DAI solver is local in strategy and there is generally no unique solution. The theorem establishes that the DAI integration procedure does not blow up provided the basis matrix is invertible and provided that the dynamics projected on to the active path and bound constraints is stable.

## 6. Regularization of the basis matrix

In this section we discuss the regularization property of the $\alpha$ discretization for improving the condition number of the basis matrix the need for which was stated in section 4.7. This is an important property since a large class of path constraints in trajectory planning when active is of index greater than 2 and the basis matrix tends to have high condition number leading to the failure of the SQP solver.

**6.1. Regularization via singular value perturbation.** We use a well known singular value perturbation expansion (Result 4.2 in [18]). Let

$$\mathcal{B}=U_B\Sigma_BV_B^T=U_B\begin{pmatrix}\mathrm{diag}(\tilde{\sigma}^{\downarrow}_{\{1,\cdots,2m+r_a-r_s\}}) & 0\\ 0 & \mathrm{diag}(\bar{\sigma}^{\downarrow}_{\{2m+r_a-r_s+1,\cdots,2m+r_a\}})\end{pmatrix}V_B^T$$

be the singular value decomposition of the basis matrix in (14) at any mesh point. The matrices $U_s, V_s$ are square unitary matrices and $\mathrm{diag}(\tilde{\sigma}^{\downarrow}_{\{1,\cdots,2m+r_a-r_s\}})$ is the diagonal matrix of singular values greater than a small positive real number $TOL$. The superscript $\downarrow$ denotes sorting in decreasing order of magnitude. Similarly $\mathrm{diag}(\bar{\sigma}^{\downarrow}_{\{2m+r_a-r_s+1,\cdots,2m+r_a\}})$ is the diagonal of singular values less than or equal to $TOL$ where $r_a$ is the number of active path constraints including some with index greater than 2 at the $i$th time step.

**Theorem 2** (Regularization)**.** *Suppose the matrix function $\mathcal{B}(\rho):(0,1)\subset\mathbb{R}\to\mathbb{R}^{(2m+r_a)\times(2m+r_a)}$ is constructed over the ith time step as in section 3.1 and has a singular value decomposition $U_B\Sigma_BV_B^T$; $U_B$ and $V_B$ being the unitary matrices of singular vectors. $\Sigma_B$ is the diagonal matrix of singular values (appearing in decreasing order of magnitude down the main diagonal). At the given $\rho$ and given iterate values $(x_i^T,a_i^T,u_i^T)^T$ at the ith mesh point, let $\mathcal{B}(\rho)$ have $r_s$ smallest singular values that are represented as zero in a given finite precision arithmetic. Further,* *let $U_B$ be column-partitioned as $(U_l\ U_s)$ where $U_s$ has $r_s$ columns. Similarly, $V_B$ is column-partitioned as $(V_l\ V_s)$ with $V_s$ having $r_s$ columns. Assume that $U_s^T \dot{\mathcal{B}} V_s$ is full rank where $\dot{\mathcal{B}} := \left(\frac{d\mathcal{B}(\rho-\varrho)}{d\varrho}\right)_{\varrho=0}$. Then $\dot{\mathcal{B}}$ has numerical rank at least equal to $r_s$ with a cut-off tolerance equal to or greater than that used for determining the numerical rank of $\mathcal{B}(\rho)$ and the perturbed basis matrix $\mathcal{B}(\rho-\varrho)$ has full numerical rank with the same cut-off tolerance as that used for $\mathcal{B}(\rho)$.*

**Proof.** Apply Result 4.2 in [18] to $\mathcal{B}(\rho-\varrho)$. Then, a decrease $\varrho > 0, \varrho \to 0$ in the value of $\rho$ perturbs the $r_s$ smallest singular values of $\mathcal{B}$ as

$$\sigma_{2m+r_a-r_s+k}\left(\mathcal{B}(\rho-\varrho)\right) \quad = \quad \sigma_k\left(U_s^T \dot{\mathcal{B}} V_s\right)\varrho + O(\varrho^2) \text{ where } k = 1, \cdots, r_s. \tag{23}$$

Hence the result. □

**Remark.** The condition $U_s^T \dot{\mathcal{B}} V_s$ being full rank may seem theoretically restrictive. However from a computational point of view it is realistic when structure of $\dot{\mathcal{B}}$ and the engineering context is considered. Since $U_s$ and $V_s$ are more likely to overlap the column space of $\mathcal{D}$, it is thus practical to make the assumption that for a significantly useful range of problems we shall obtain a full rank $U_s^T \dot{\mathcal{B}} V_s$. From an engineering point of view this is the coupling between dynamics and control and is physically realistic to be assumed being available.

**Theorem 3** (Magnitude of Regularization)**.** *Let the matrix $\mathcal{A}_i\mathcal{A}_i^T + \mathcal{D}_i\mathcal{D}_i^T$ (as defined in equations (16)) have numerical rank $p \le m$ ($m$ being the dimension of differential variables $x$) with a suitable cut-off tolerance at the $i$th mesh point with $h_{i-1}$ set to $h$.*
*Then, $\sigma_p(\dot{\mathcal{B}}) \ge h\frac{4(1-\rho)}{(\rho+1)^2(3-\rho)^2}\sigma_p^{\frac{1}{2}}(\mathcal{A}_i\mathcal{A}_i^T + \mathcal{D}_i\mathcal{D}_i^T)$.*

**Proof.** Differentiating $\mathcal{B}$ with respect to $\rho$ and taking $\varrho = 0$ we find $\dot{\mathcal{B}}$. Then we construct $\dot{\mathcal{B}}\dot{\mathcal{B}}^T$ and exploiting the structure we write

$$\dot{\mathcal{B}}\dot{\mathcal{B}}^T \quad = \quad X^T\left(\mathcal{A}_i\mathcal{A}_i^T + \mathcal{D}_i\mathcal{D}_i^T\right)X =: X^T\mathcal{Y}X \tag{24}$$

where $X := \left(\frac{d(\beta/\gamma)}{d\rho}hI_{m\times m}, \quad \frac{d(1/\gamma)}{d\rho}\frac{1}{h}I_{m\times m}, \quad 0\right)$. $X$ is clearly rank $m$. The matrix $\mathcal{Y}$ is real symmetric positive semi-definite. Obviously, $XX^T$ is invertible with one eigenvalue $(h\frac{d}{d\rho}\frac{\beta}{\gamma})^2 + (\frac{1}{h}\frac{d}{d\rho}\frac{1}{\gamma})^2$ of algebraic multiplicity $m$. Thus $\sigma^2_{\{1,\cdots,m\}}(X) = (h\frac{d}{d\rho}\frac{\beta}{\gamma})^2 + (\frac{1}{h}\frac{d}{d\rho}\frac{1}{\gamma})^2$. Applying Cauchy's interlacing theorem, we estimate that $\sigma_p^2(\dot{\mathcal{B}}) = \sigma_p(\dot{\mathcal{B}}\dot{\mathcal{B}}^T) \ge h^2\sigma_p(\mathcal{A}_i\mathcal{A}_i^T + \mathcal{D}_i\mathcal{D}_i^T)\left(\frac{d}{d\rho}\left(\frac{\beta}{\gamma}\right)\right)^2 = h^2\left(\frac{4(1-\rho)}{(\rho+1)^2(3-\rho)^2}\right)^2\sigma_p(\mathcal{A}_i\mathcal{A}_i^T + \mathcal{D}_i\mathcal{D}_i^T)$. □

From its structure $\sigma_1(\mathcal{B}) \ge 1 \gg TOL$. Then increase of the smallest singular values is sufficient for regularization.

**Corollary 1.** *If $r_s \le \mathbf{rank}(\mathcal{A}_i\mathcal{A}_i^T + \mathcal{D}_i\mathcal{D}_i^T, TOL) = p$ ($r_s$ as in Theorem 3), all the smallest singular values of $\mathcal{B}$ are regularized as*

$$\sigma_{2m+r_a-r_s+k}(\mathcal{B}(\rho-\varrho)) \quad \ge \quad \eta_r h\frac{4(1-\rho)}{(\rho+1)^2(3-\rho)^2}\sigma_p^{\frac{1}{2}}(\mathcal{A}_i\mathcal{A}_i^T + \mathcal{D}_i\mathcal{D}_i^T)\varrho + O(\varrho^2) \tag{25}$$
$$\textit{where } \eta_r \in (0,1],\ 1 \gg \varrho > 0,\ \rho \in [0,1),\ k = 1, \cdots, r_s.$$

*provided the assumptions made in Theorem 2 holds. Here $h$ is the time step size for the particular time step at which the solution is being sought and $\eta_r$ is a function of $\|U_s^T W\|$ and $\|Z^T V_s\|$ ($U_s, V_s$ as in Theorem 2, $W, Z$ being matrices of left and right singular vectors of $\dot{\mathcal{B}}$).*

**Proof.** The result follows directly from Theorems 2 and 3 and by rewriting (23). □

**6.1.1. Scaling.** A scaling (as in [2]) inside QP iterations can be applied to $\mathcal{B}$ by scaling $x$ as $x/h$ and $a$ as $ah$. Then obviously $X$ in (24) becomes

$$\begin{pmatrix} \frac{d(\beta/\gamma)}{d\rho} I_{m\times m}, & \frac{d(1/\gamma)}{d\rho}\frac{1}{h} I_{m\times m}, & 0 \end{pmatrix}$$

so that the magnitude of regularization is then independent of the time step and the first term on the right hand side of (25) becomes $\eta_r \frac{4(1-\rho)}{(\rho+1)^2(3-\rho)^2}\sigma_p^{\frac{1}{2}}(\mathcal{A}_i\mathcal{A}_i^T + \mathcal{D}_i\mathcal{D}_i^T)\varrho$.

**6.1.2. Remark.** The regularization is consistent since the DAI discretization (2) is stable and consistent for $\rho$ in $[0, 1)$. The regularization results in this section can be viewed as the time step by time step case of the more general result for a direct transcription process found in [16].

## 7. Outline of the DAI integrator algorithm

**Step 0.** Specify $t_0, t_f, h_{\max}, ERRTOL, TOL, MAXIT, MAXTRY$.
**Step 1.** Check for and compute, if necessary, consistent initialization, i.e., $x_0, u_0$ that satisfy the DAI at $t_0$.
**Set** $i = 0$.
**While** $t_i < t_f$
**If** $i = 0$ **then set** $h_{i-1} = h_{\max}$. **End If**
**Set** $NTRY = 0$.
**Step 2.** Select $\rho \in [0, 1)$ and a non-zero time step size

$$h_i = \sqrt{\min\{h_{\max}^2, h_{i-1}^2, \frac{ERRTOL}{\max_{j\in[0,i]} 2.25\|a_i\|_\infty)}\}}.$$

Solve using the SQP method (with constraint satisfaction tolerance $TOL$ and maximum number of total minor iterations $MAXIT$) for $x_{i+1}, u_{i+1}$ the following *constraint satisfaction* problem with $x_g := x_i,\ u_g := u_i$ as initial guess values:

$$\begin{aligned}
&\min_{u_{i+1},x_{i+1}} J(u_{i+1}, x_{i+1}) = 0 \text{ subject to}\\
x_{i+1} &= x_i + \left(1 - \frac{\beta}{\gamma}\right) h_i f(x_i, u_i, t_i) + \frac{\beta}{\gamma} h f(x_{i+1}, u_{i+1}, t_{i+1}) + \left(\frac{1}{2} - \frac{\beta}{\gamma}\right) h_i^2 a_i\\
a_{i+1} &= \frac{1}{h_i\gamma}\left(f(x_{i+1}, u_{i+1}, t_{i+1}) - f(x_i, u_i, t_i)\right) + \left(1 - \frac{1}{\gamma}\right) a_i,\\
0 &\geq \phi(x_{i+1}, u_{i+1}, t_{i+1})\\
&b_{upper} \geq \begin{pmatrix} x_{i+1} \\ u_{i+1} \end{pmatrix} \geq b_{lower}
\end{aligned}$$

**If** SQP has converged **then set** $t_{i+1} = t_i + h_i$; $i = i + 1$ **else**
**Step 2a. If** the SQP solver has returned infeasible constraints
**then go to** Step 2c. **End If**
**Step 2b. If** the SQP solver has returned rank deficiency **then**
**if** $NTRY = MAXTRY$ **or** ($\rho = 0$ **and** $h_i = h_{max}$) **then** go to Step 2c
**else if** $\rho \neq 0$ select a new $\rho \in [0, 1)$ reduced by 0.1. **if** $\rho = 0$, **set** $h_i = h_{max}$ (Try regularization)
**Set** $NTRY = NTRY + 1$.
**Go to** Step 2. **End If**

```
End If
Step 2c. If $NTRY < MAXTRY$ then
Choose new $x_g, u_g$. Set $NTRY = NTRY + 1$; go to Step 2
else exit with failure. End If
End If
End While
```

**7.1. Remark.** The SQP method is used not as an optimizer in the above algorithm but merely for detecting and satisfying active path constraints. A local objective function, i.e., non-zero $J$ can be used and this would be similar to trust region methods [14] in its approach.

## 8. Numerical Examples

**8.1. Re-entry of an RLV.** The simplified re-entry phase equations of a Re-usable Launch Vehicle(RLV) with stationary earth and zero bank angle is considered from [3] with data similar to [4]. The RLV re-enters the earth's atmosphere from orbit and descends to the terminal area energy management phase. The state variables are altitude ($h_a$), flight path angle ($\gamma$) and the velocity ($V$). The control variable is the angle of attack($\alpha$). The dynamics in quasi-equilibrium glide with no banking and spherical earth is given by

$$\begin{aligned}\dot{h}_a &= V\sin(\gamma)\\ \dot{V} &= -\frac{D}{m} - g\sin(\gamma)\\ \dot{\gamma} &= \frac{L}{mV} + \frac{\cos(\gamma)}{V}\left(\frac{V^2}{r} - g\right)\end{aligned}$$

where $D = \frac{1}{2}C_D(\alpha, M)SV^2$ is drag force, $L = \frac{1}{2}C_L(\alpha, M)SV^2$ is the lift force, $S$ is the reference area, $\rho_a(h_a)$ is the density taken from standard atmospheric data. $C_L(\alpha, M)$ is the coefficient of lift, $C_D(\alpha, M)$is the coefficient of drag, $M$ is the Mach number, $m$ is the mass of the vehicle, $g(r) = g(r_e)(r_e^2/r^2)$ is the acceleration due to gravity, $r = h_a + r_e$ is the distance from the Earth's center, $r_e$ is the radius of the Earth. The inequality path constraints are given by
Heat Flux : $C_H\rho_a^{0.5}V^{3.05} < 18.5W/m^2$
($C_H$ is determined by Chapman heat rate formula and vehicle's nose radius [4]. )
Dynamic Pressure : $\frac{1}{2}\rho_a V^2 < 25KPa$ and
Load Factor : $\frac{Lcos(\alpha)+Dsin(\alpha)}{mg} < 4.0$.
The first two of the above path constraints can be of local index greater than 2 when active.

*Results.* The initial condition vector was taken as $[h_a(0), \gamma(0), V(0)]^T = [150000, 0, 1500]^T$ in SI units. The SQP solver used was an implementation of an SQP algorithm similar to the SNOPT [15]. The control variable $\alpha$ has lower and upper bounds as shown in the figures. Figures 1 to 3 show the trajectories for different $\rho$ and integration step sizes. From figure 1 it is clear that $\rho \to 1.0$ provides the least regularization when the path constraints are active at around 25 km altitude. Note that the path constraints are activated only for 1 or 2 time steps. The problem does not have a unique solution and the solutions vary with $\rho$ and the size of the uniform time step.

**8.2. Navigation of a Robot Vehicle Around Obstacles.** Several simulations were performed on a simplified model of a wheeled robot. The problem is taken from [19]. The state variables are planar coordinates ($x$, $y$), velocity $V$, and orientation

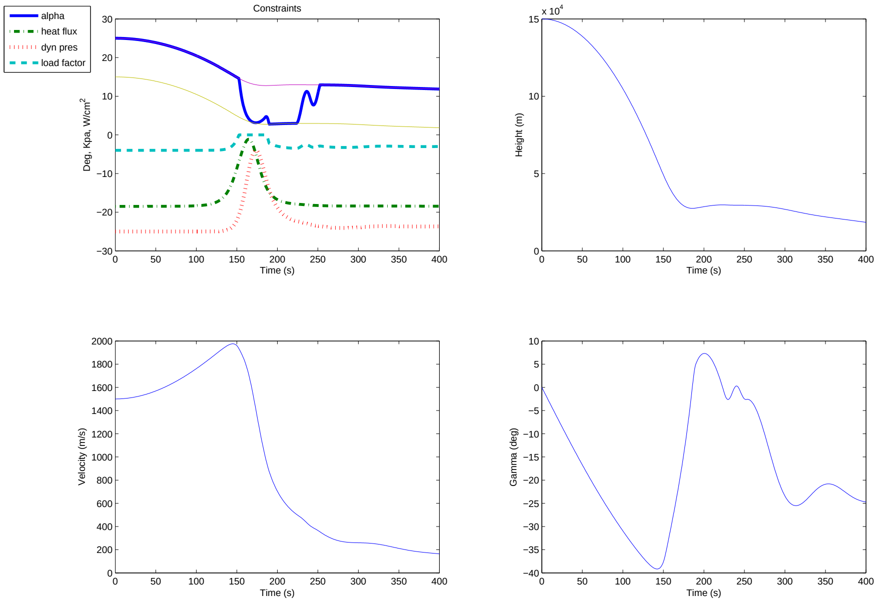

FIGURE 1. RLV trajectories for $stepsize = 0.25s$, $\rho = 0.99$

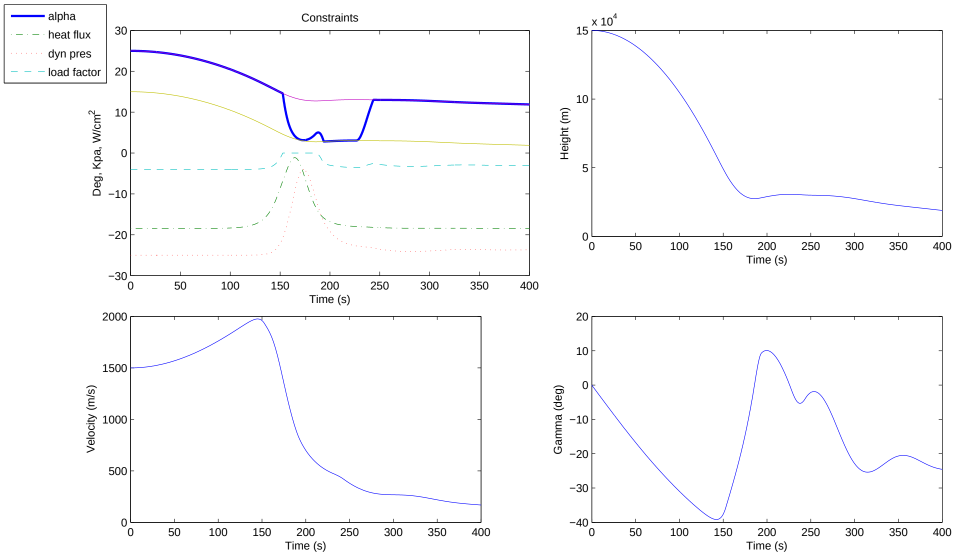

FIGURE 2. RLV trajectories for $stepsize = 0.25s$, $\rho = 0.5$

$\theta$. The control variables are steering $\alpha$ and acceleration $u$. The model takes into

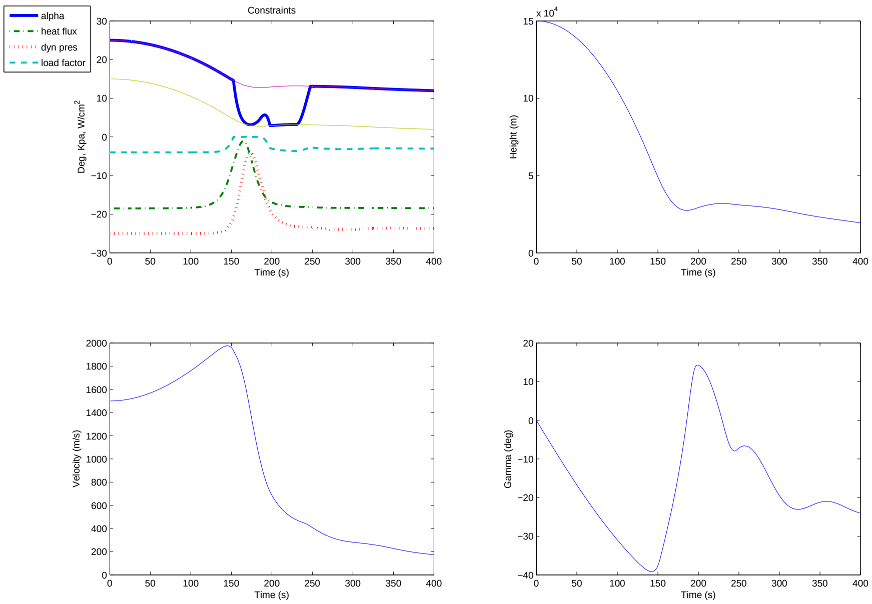

FIGURE 3. RLV trajectories for $stepsize = 0.25s$, $\rho = 0$

account dynamic friction $\mu_d$, a constant. The dynamics is given by

$$\begin{aligned}\dot{x} &= V\cos\theta\\ \dot{y} &= V\sin\theta\\ \dot{\theta} &= V\alpha\\ \dot{V} &= u - \mu_d V\end{aligned}$$

Collisions with obstacles are avoided by enforcing inequality constraints (which are index 3 when active) for each obstacle's convex hull modeled with its circumcircle $(r_v + r_{obs}^{(i)}) - \sqrt{(x - x_{obs}^{(i)})^2 + (y - y_{obs}^{(i)})^2} < 0$ where $(x, y)$ are the vehicle center coordinates, $(x_{obs}^{(i)}, y_{obs}^{(i)})$ are the coordinates of the centroid of the $i$th obstacle, $r_v$ is the vehicle radius, and $r_{obs}$ circumradius of the convex hull of the $i$th obstacle. Forward motion of the vehicle is achieved by means of a traveling wave of speed $W$ imposing an inequality constraint of the form: $Wt - x < 0$. The same bounds on states and controls as in [19] are used in the simulations. It may be seen that steering $\alpha$ corresponds to the obstacle avoidance constraints and that the forward motion constraint corresponds to the acceleration $u$. The problem obviously does not have a unique solution and trajectory generated varies with $\rho$ and the size of the uniform time step, $h$.

*Results.* The results of the simulation are shown as the trajectories traced out by the robot on the plane ($y$ as vertical axis and $x$ as horizontal) in the figures 4 to 7. The regularization affected by $\rho$ and step size $h$ influences the trajectory of the robot. As seen from Corollary 1, as $\rho \to 1$ the least regularization is obtained. In figure 7, a larger time step size improves the regularization over that in figure 6. However a larger time step may not be desirable since it may miss any path

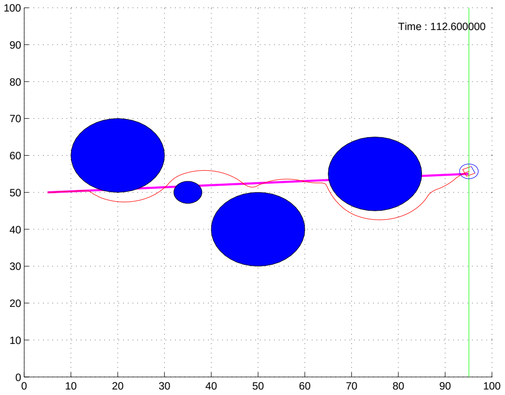

FIGURE 4. Navigation of a robot around obstacles, step size = 0.1s, $\rho = 0$

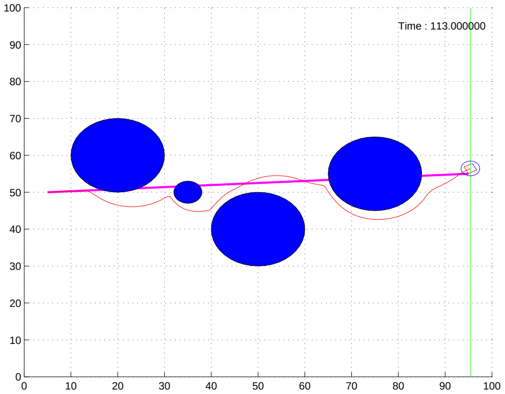

FIGURE 5. Navigation of a robot around obstacles, step size = 0.5s, $\rho = 0$

constraints active for short duration. It may be mentioned that the trajectory shown in figure 4 is comparable to that obtained using the method in [19]. At $\rho = 0$ with $0.1s$ step size a useful regularization is obtained. The steering $\alpha$ and acceleration $u$ both show sharp changes and low continuity across time steps but are admissible because of the Lipschitz assumption (with large but finite Lipschitz constant) made in our analysis.

## 9. Conclusions

An algorithm for generating a feasible solution to an initial value DAI problem at a computational cost cheaper than the full scale dynamic optimization algorithms (e.g., Multiple Shooting, Direct Transcription) has been presented and illustrated with examples. Usage as initial guess generator and as trajectory planner is intended. If adjoint sensitivity equations to the active DAE at each time step can be incorporated then time step wise adjoint sensitivities of the DAI system may

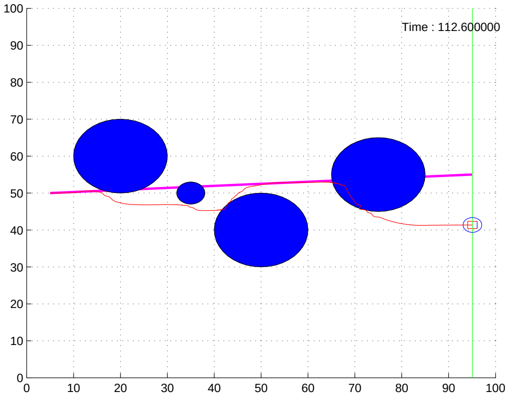

FIGURE 6. Navigation of a robot around obstacles: Failed Regularization, step size = 0.1s, $\rho$ =0.99

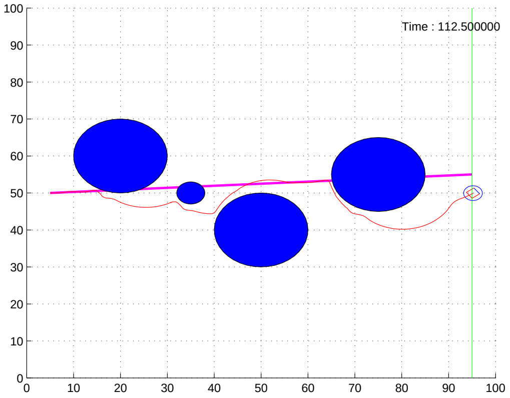

FIGURE 7. Navigation of a robot around obstacles: step size = 0.5s, $\rho$ = 0.99

be found. The algorithm can then be used an integrator in a Multiple Shooting algorithm for better handling of path inequality constraints over integration subintervals.

## References

[1] J.T. Betts, N. Biehn, and S.L. Campbell. Convergence of Nonconvergent IRK Discretizations of Optimal Control Problems with State Inequality Constraints. *SIAM Journal on Scientific Computing*, 23(6):1981–2007, 2002.

[2] C.L. Bottasso, O.A. Bauchau, and A. Cardona. A Time-step-size-independent conditioning and sensitivity to perturbations in the numerical solution of index three differential algebraic equations. *SIAM Journal on Scientific Computing*, 29:397–414, 2007.

[3] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* Society for Industrial and Applied Mathematics (SIAM), second revised edition, 1996.

[4] V. Brinda, R.K. Arora, and E. Janardhana. Mission analysis of a reusable launch vehicle technology demonstrator (RLV-TD). *Proc. AIAA/CIRA 13th International Space Planes and Hypersonics Systems and Technologies, Capua, Italy*, pages 1–10, 2005.

[5] J. Chung and G. M. Hulbert. Time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- $\alpha$ method. *Journal of Applied Mechanics, Trans. ASME.*, 60(2):371–375, 1993.

[6] D. Cruz-Uribe and C.J. Neugebauer. Sharp error bounds for the trapezoidal rule and Simpson's rule. *Journal of Inequalities in Pure and Applied Mathematics*, 3(4:49):1–49, 2002.

[7] C. Dever, B. Mettler, E. Feron, J. Popovic, and M. McConley. Nonlinear trajectory generation for autonomous vehicles via parameterized maneuver classes. *Journal of Guidance, Control, and Dynamics.*, 29(2):289–302, 2006.

[8] R. Findeisen, L. Imsland, F. Allgöwer, and B. Foss. State and output feedback nonlinear model predictive control: An overview. *European Journal of Control*, 9(2-3):190–206, 2003.

[9] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations 2. Stiff and Differential-Algebraic Problems*. Springer-Verlag, third printing of second revised edition, 2004.

[10] M. Kalisiak and M. van de Panne. Approximate safety enforcement using computed viability envelopes. *Proceedings. ICRA '04. 2004 IEEE International Conference on Robotics and Automation*, 5:4289–4294, 2004.

[11] Y-j. Li, D. J. Hill, and T-j. Wu. Optimal coordinated voltage control of power systems. *Journal of Zhejiang University - Science A*, 7(2):257–262, 2006.

[12] C. Lunk and B. Simeon. Solving constrained mechanical systems by the family of Newmark and alpha-methods. *Journal of Applied Mathematics and Mechanics (ZAMM)*, 86:772–784, 2006.

[13] G. Mao and L. R. Petzold. Efficient integration over discontinuities for differential-algebraic systems. *Computers and Mathematics with Applications*, 43(1–2):65–79, 2002.

[14] P. K. Moore, L. R. Petzold, and Y. Ren. Regularization of index-1 differential-algebraic equations with rank-deficient constraints. *Computers and Mathematics with Applications*, 35(5):43–61, 1998.

[15] W. Murray, P.E. Gill, and M.A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12:979–1006, 2002.

[16] N.C.Parida and S. Raha. The $\alpha$-method direct transcription in path constrained dynamic optimization. *SIAM Journal on Scientific Computing*, 31(3):2386–2417, 2009.

[17] S. Raha and L. R. Petzold. Constraint partitioning for structure in path-constrained dynamic optimization problems. *Applied Numerical Math.*, 39:105–126, 2001.

[18] T. Söderström. Perturbation results for singular values. Technical Report 1999-001, Information Technology Department, Uppsala University, Uppsala, Sweden, 1999.

[19] R.J. Spiteri, D.K. Pai, and U.M Ascher. Programming and control of robots by means of differential algebraic inequalities. *IEEE Transactions on Robotics and Automation*, 16(2):135–145, 2000.

[20] J. Yen, L. Petzold, and S. Raha. A Time integration algorithm for flexible mechanism dynamics : the DAE-$\alpha$ method. *J. Comput. Meth. Appl. Mech. Engr. (CMAME)*, 158:341–355, 1998.

GE India Infrastructures (Energy), GE John F. Welch Technology Centre, Bangalore 560066, India

*E-mail*: joey.peter@ge.com

Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore 560012, India

*E-mail*: parida@alumni.iitg.ernet.in and raha@serc.iisc.ernet.in