pp 770–789. © Royal Aeronautical Society 2017
doi: 10.1017/aer.2017.36

Trajectory feasibility evaluation using path prescribed control of unmanned aerial vehicle in differential algebraic equations framework*

T. Uppal
tarunuppal2007@gmail.com
Aeronautical Development Establishment
Bangalore
India

S. Raha
Indian Institute of Science
Bangalore
India

S. Srivastava
O/o Director General Aeronautical
Bangalore
India

ABSTRACT

Mission simulation is a critical activity in the development and operation of Unmanned Aerial Vehicles (UAVs). It is important to ascertain the feasibility of a trajectory in a mission. In this work, an algorithm has been developed for feasibility study of a trajectory of a UAV using prescribed path optimal control through an inverse simulation method. This has been done under a Differential Algebraic Equations (DAE)/Inequalities (DAI) framework. The UAV model together with constraints is represented as a high index DAE system. The trajectory that UAV shall take is prescribed as one of the constraint equations. The solution for the DAE system is obtained using a variation of the alpha method that is capable of handling both equality and inequality constraints on system dynamics. The algorithm involves direct numerical integration of a DAI formulation in a time-stepping manner using a Sequential Quadratic Programming (SQP) solver that detects and satisfy active path constraints at each

* Corresponding author Tarun Uppal. Tel. +91-80-25605681. Fax +91-80-25283188.

Received 27 January 2016; revised 8 March 2016; accepted 14 March 2017.

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

time step (mesh point). In this unique approach, the model and the constraints are always solved together. The method ensures stable solution at each time step, local minimum at each iteration of simulation and provides a regularised basis to the solver. A typical UAV trajectory has been simulated and demonstrated in this paper. This new approach can be used for path planning of UAVs before the actual control law is designed for flight control computer. Compared to other existing computationally intensive techniques, this approach is computationally simple, ensures continuous constraint satisfaction and provides a viable option for model predictive control of UAVs.

Keywords: Path planning; regularisation; DAEs; inverse simulation

SYMBOLS & NOTATIONS

α	angle-of-attack
β	sideslip angle
u	component of perturbed speed in X-direction
υ	component of perturbed speed in Y-direction
ω	component of perturbed speed in Z-direction
p	roll rate
q	pitch rate
r	yaw rate
τ	quaternion
ϕ	roll angle
θ	pitch angle
ψ	heading of aircraft
x	aircraft position in X-direction
y	aircraft position in Y-direction
z	aircraft position in Z-direction
δe	elevator angle deflection
δa	aileron angle deflection
δr	rudder angle deflection
R_M	total rolling moment
P_M	total pitching moment
N_M	total yawing moment
F_X	total force in X-direction
F_Y	total force in Y-direction
F_Z	total force in Z-direction
I_{xx}	moment of inertia of the aircraft about x-axis
I_{yy}	moment of inertia of the aircraft about y-axis
I_{zz}	moment of inertia of the aircraft about z-axis
x_n	value of state variables at nth step
x_{n+1}	value of state variables at nth step
a_n	value of acceleration term at nth step
a_{n+1}	value of acceleration term at $n+1$st step
ρ	alpha method tuning parameter-1
σ	alpha method tuning parameter-2
m	Mass of the aircraft

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

Abbreviations

DAE	DifferentialAlgebraic Equation
DAI	Differential Algebraic Inequality
DDAE	Delayed Differential Algebraic Equation
DDAI	Delayed Differential Algebraic Inequality
DOF	Degrees of Freedom
DT	Direct Transcription
FCC	Flight Control Computer
FCS	Flight Control System
MALE	Medium Altitude Long Endurance
NGC	Navigation, Guidance & Control
ODE	Ordinary Differential Equation
PID	Proportional Integral Derivative
MS	Multiple Shooting
POC	Proof of Concept
SNOPT	Sparse NonLinear Optimiser
SQP	Sequential Quadratic Programming
UAV	Unmanned Aerial Vehicle

1.0 INTRODUCTION

1.1 Unmanned aerial vehicle control system

Unmanned Aerial Vehicles (UAVs) are being used in both military and civil applications worldwide. Due to their obvious advantages for the military, they are main elements of modern day warfare being employed in surveillance, reconnaissance and even combat roles. UAVs are essentially a force multiplier in battlefield. The missions taken by the military UAVs are complex and involve various trajectories. The control system of the aircraft plays a vital role in meeting the vehicle's mission objectives. The complex mission necessitates use of more onboard intelligence, rigorous design, simulation and off-line analysis techniques. Modern-day UAV Flight Control System (FCS) consists of Flight Control Computer (FCC) with embedded control logics for Navigation, Guidance and Control (NGC) of the aircraft. The current-generation UAV control laws are, in general, based on sensor feedback–based classical control employing cascaded gain-scheduled Proportional Integral Derivative (PID) control loops. The longitudinal and lateral-directional decoupled controller design is done in a linear domain using small perturbation analysis and validated in non-linear simulation models. Detailed simulations are carried out for flight mechanics and control studies of the air vehicle during vehicles design and development phases. Various new generic control methods have appeared in the literature in the past, each citing their own advantages over the currently used techniques of control system design. There is a need for methods that can act as both design and analysis tools for control system development. Inverse simulation is one such approach that has been discussed in the literature and has received attention in aerospace applications[1-6]. The inverse simulation approach can be used to do fly-ability of trajectories that a UAV can take during a mission. Though an aerodynamics designer would calculate the manoeuvering envelopes through purely analytical equations with various assumptions, using the inverse simulation approach cited in this paper, the flight mechanics analyst/control

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

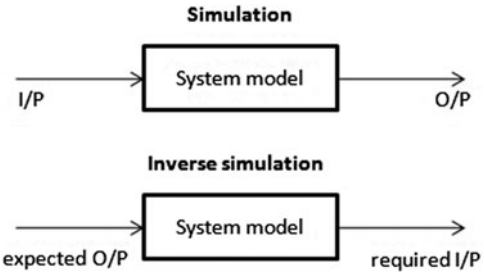


Figure 1. Inverse simulation vs simulation.

law designer can validate the performance in a comprehensive model much before the actual control laws are designed and flown in the simulation.

1.2 Inverse simulation and its requirement

In the conventional simulations, the input is provided to the model of the plant and the plant outputs are studied. On the contrary, in the inverse simulation approach the intended outputs are provided to the plant model, and inputs are obtained as outputs from this model (Fig. 1). This approach is a powerful technique for system analysis studies, as it can bring out system limitations and may also help in providing clues for suitable design modifications in the system to meet the prescribed system objectives. Path prescribed control of UAVs is one such research problem wherein theory of inverse simulation finds its application. The specific manoeuvre for an aircraft in terms of a series of positions in three dimensions (in an earth-based axis system) can be defined as the path the UAV must take[7]. This can act as the prescribed input for the aircraft inverse simulation model. The solution to this model can give information on control surface deflections and actuator movement to allow the aircraft to follow the prescribed manoeuvre. The UAV trajectory planning and optimisation problem has been discussed in the literature, and finding feasible trajectories is a critical activity for flight mechanics engineer. The current methods for trajectory planning are computational intensive involving a large set of simulation studies and are difficult to apply in practical applications. This paper uses the Differential Algebraic Equations (DAE)/Differential Inequalities (DAE/DAI)-based numerical solution approach to carry out inverse simulation for a UAV.

1.3 DAEs

The general DAE systems are of the form

$$\dot{x}(t) = f(x(t), u(t), t), \quad \ldots (1)$$

$$0 = g(x(t), u(t), t), \quad \ldots (2)$$

where f is a function of system dynamics, g is a function of constraints, u is the control vector, x is the vector of state variables and t is time. Here, if $\partial f/\partial \dot{x}$ is singular, then this system is not equivalent to Ordinary Differential Equation (ODE) system of the form

$$\dot{x} = f(x(t), u(t), t) \quad \ldots (3)$$

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

DAE formulation naturally appears in constraint variational problems[8,9] such as that of prescribed path control. If inequalities are involved in the problem, which will usually be the case in motion planning, the DAE is known as a Differential Algebraic Inequality (DAI) problem. The DAI-based numerical path planning of robots has been demonstrated in Ref. 10.

DAE and DAI are usually solved numerically and are much more challenging than initial value ODEs. The challenge in solution to a DAE system of equations is measured by a metric called the index of the DAE/DAI system.

The flight dynamics model of the UAV is generally represented as a set of initial value ODEs. However, for the path prescription problem, the model appears in the form of Equations (1) and (2), where in Equation (1) are the flight dynamic differential equations and (2) is the constraint equations in terms of path to be taken by the UAV, state/control and saturation constraints.

1.4 Index of DAE

Index of DAE is a metric that plays a key role in classification and behaviour of DAEs.

Definition 1. *It is defined as minimum number of differentiations needed to solve for $\dot{x}$ in terms of t and x in Equations (1) and (2). For instance, if*

$$F(\dot{x}, x, t) = 0 \qquad \ldots (4)$$

is the function, then if

$$\frac{dF}{dt}(t, x, \dot{x}, \ddot{x}) = 0, \qquad \ldots (5)$$

$$\vdots \qquad \ldots (6)$$

$$\frac{d^p F}{dt^p}(t, x, \ldots, x^{p+1}) = 0, \qquad \ldots (7)$$

p will be called as index for the given system.

In other words, if $F(t, x, \dot{x}) = 0$ is a DAE system, then p is the index of the system if p is the minimal number of required differentiations such that the system allows to extract an explicit ODE $\dot{x} = \phi(x(t), t)$ using only algebraic manipulators. As a simple example, $y = g(t)$ is index-1, whereas $\dot{y}1 = y2, y1 = g(t)$ is index-2.

For the general DAE system defined in Equations (1) and (2), if we differentiate Equation (2) with respect to t, we get

$$g_x(x, u, t)\dot{x} + g_u(x, u, t)\dot{u} = g_t(x, u, t)$$

If g_u is non-singular, then the system is a simple implicit ODE and the index is one. If g_u is singular, then we differentiate constraint equation again.

As a generality, an ODE can be classified as a DAE of index-0. The higher the index, the more difficult it will be to deal with the set of equations numerically[11] due to rank deficient Jacobian matrix and hidden algebraic constraints[9]. More details of DAEs and their properties can be found in Ref. 8.

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

1.5 High-index DAEs

Higher-index DAEs appear in complex engineering problems such as one discussed in this paper. The high-index problems have hidden constraints that are the derivatives of the constraints appearing in the actual problem. Numerical solution of very high-index problems (more than index-5) is still a widely discussed open research problem[12,13]. High-index systems have solvability- and regularity-related issues that must be handled well by numerical method used for direct solution of these systems. Traditionally, these problems are either dealt with by index reduction techniques or are attempted through some better direct method such as the alpha method[12] used in this work.

1.6 Numerical solvers for DAEs

The numerical solution of DAEs has been shown in Ref. 8. Other alternate methods such as solving DAEs by Taylor series: Differential Algebaric Equations by Taylor Series (DAETS) code have also been published in the past[14]. The numerical solution of DAE problems using the Differential Algebraic System Solver (DASSL) software package is shown in Ref. 8. This package and its variants, such as Large Scale Differential Algebraic Equation Solver (DSAPK), Differential Algebraic Equation Solver and Optimizer (DASOPT) etc., are most popular for solving low-index DAEs (up to index-2). The alpha method for solving a differential algebraic equations-based system has been discussed in Refs 15–18. The authors have brought out various features and properties of this method along with proofs. The compilation of various high-index DAEs in engineering applications such as the one discussed in this paper is discussed Refs 11 and 12. Combined with a suitable non-linear optimiser, the DAE solver can be extended to optimal control. Direct Transcription (DT) has been another popular optimal control method for solution of higher-index optimal control problems. The alpha method direct transcription in path-constrained dynamic optimisation has been brought out by Refs 18 and 19. The constraint partitioning for stability and structure are discussed in Refs 20 and 21. An innovative geometric approach to solving problems of control constraints in a DAE framework is discussed in Refs 19 and 22.

The approach used for the reported UAV path prescription and feasibility problem in this paper is a variation of the alpha method and is an alternate to computationally intensive methods such as the Sparse Non-Linear Optimiser (SNOPT) used in the industry[23] and can also be used in the model predictive control[24].

The present work involves formulating an optimal control path prescription problem of a UAV using DAE formulation and applying the modified alpha method and a solver for its solution. This approach can be used for mission simulation studies of a UAV prior to control law design and can act as a good tool for bringing out system deficiencies with regard to control saturations, non-reachability, feasibility of a particular flight path, etc.

2.0 MODEL DETAILS

2.1 Simulation model of UAV

The UAV model used for this study is of a single engine Medium-Altitude Long-Endurance (MALE) UAV of approximately 800 kg class with various payloads under development. It has an endurance of more than 10 hours.

The dynamics of this plant can be represented as the basic rigid body 6-DOF equations that govern the motion of the aircraft. In generic form, the equations of this model may be

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

written as

$$\dot{x} = f(x, u, t), \qquad \ldots (8)$$

where

$x = (u, v, w, p, q, r, \varsigma_0, \varsigma_1, \varsigma_2, \varsigma_3, x, y, z)$ represent state vector

$u = (\delta e, \delta a, \delta r, \delta T)$ represent control vector

There are 13 state variables in vector x and 4 control variables in vector u. State variables include velocities u, v and w in three-dimensional space, pitch, roll and yaw rates (p, q, r) in body axes; four quaternion variables $(\varsigma_0, \varsigma_1, \varsigma_2, \varsigma_3)$, which can give Euler angles (θ, ϕ, ψ); and x, y and z positions of the aircraft in earth fixed frame of reference. The control variables include elevator (δe) for pitch control, aileron (δa) for roll control, rudder (δr) for yaw control, and throttle (δT) for thrust generation from engine.

The set of governing non-linear differential equations representing system dynamics are given by the following equations:

$$\dot{p} = R_M/I_{xx} + (I_{yy} - I_{zz}) \cdot q \cdot r/(I_{xx}),$$

$$\dot{q} = P_M/I_{yy} + (I_{zz} - I_{xx}) \cdot r \cdot p/(I_{yy}),$$

$$\dot{r} = Y_M/I_{zz} + (I_{xx} - I_{yy}) \cdot p \cdot q/(I_{zz}),$$

$$\dot{u} = Fx_{Total}/m + (r \cdot v - q \cdot w),$$

$$\dot{v} = Fy_{Total}/m + (p \cdot w - r \cdot u),$$

$$\dot{w} = Fz_{Total}/m + (q \cdot u - p \cdot v),$$

$$\dot{\varsigma_0} = -0.5(p \cdot \varsigma_0 + q \cdot \varsigma_1 + r \cdot \varsigma_3),$$

$$\dot{\varsigma_1} = -0.5(-p \cdot \varsigma_0 - r \cdot \varsigma_2 + q \cdot \varsigma_3),$$

$$\dot{\varsigma_2} = -0.5(-q \cdot \varsigma_0 + r \cdot \varsigma_1 - p \cdot \varsigma_3),$$

$$\dot{\varsigma_3} = -0.5(-r \cdot \varsigma_0 - q \cdot \varsigma_1 + p \cdot \varsigma_2),$$

$$\dot{x} = (2 \cdot ({\varsigma_0}^2 + {\varsigma_1}^2) - 1) \cdot u + 2 \cdot (\varsigma_0 \cdot \varsigma_1 - \varsigma_0 \cdot \varsigma_3) \cdot v + 2.0 \cdot (\varsigma_1 \cdot \varsigma_3 - \varsigma_0 \cdot \varsigma_2) \cdot u,$$

$$\dot{y} = (2 \cdot ({\varsigma_0}^2 + {\varsigma_2}^2) - 1) \cdot v + 2 \cdot (\varsigma_0 \cdot \varsigma_3 - \varsigma_0 \cdot \varsigma_3) \cdot u + 2 \cdot (\varsigma_2 \cdot \varsigma_3 - \varsigma_0 \cdot \varsigma_1) \cdot w,$$

$$\dot{z} = (2 \cdot ({\varsigma_0}^2 + {\varsigma_3}^2) - 1) \cdot w + 2 \cdot (\varsigma_1 \cdot \varsigma_3 - \varsigma_0 \cdot \varsigma_2) \cdot v + 2 \cdot (\varsigma_2 \cdot \varsigma_3 - \varsigma_0 \cdot \varsigma_1) \cdot v$$

The aerodynamic angles are computed as:

$$\alpha = \tan^{-1}(w/u),$$

$$\beta = \tan^{-1}(v/u)$$

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

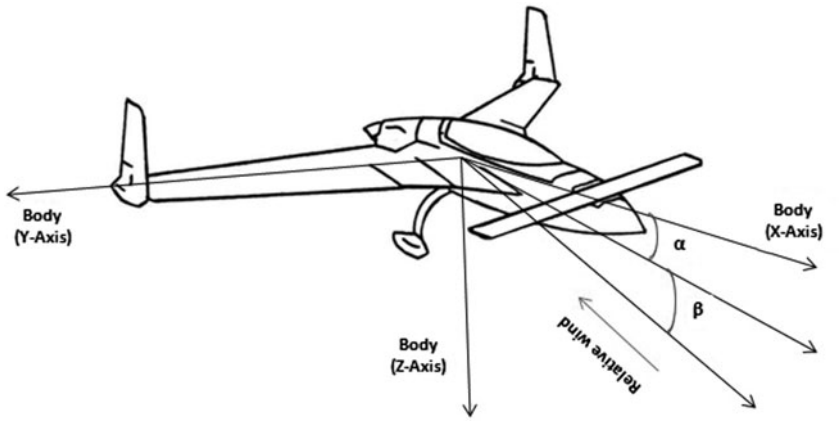


Figure 2. Axes systems for a general aircraft.

The total velocity of the UAV is computed from three velocity components as:

$$V = \sqrt{u^2 + v^2 + w^2}$$

The dynamic pressure is calculated using air density at a point and vehicle speed as:

$$P = \frac{1}{2}\rho V^2$$

The details of the symbols used above are given in the nomenclature section of this paper. There are three forces and three moments acting on the body. The inertia, propulsion, aerodynamics and gravitational components are considered for calculations of these as total forces and moments denoted by R_M, P_M, Y_M, Fx_{Total}, Fy_{Total} and Fz_{Total}, respectively. The force and moment coefficients are functions of various non-dimensional stability, dynamic and control derivatives, which are based on theoretical prediction/wind-tunnel tests. The forces and moments in three axes get altered using four control variables: elevator (δe), aileron (δa), rudder (δr) and engine throttle δT. A hybrid axes system (consisting of body and stability axis system) has been used during the model formulation. The basic conventions for a generic aircraft 6-DOF model are shown in Fig. 2. The X-axis is considered in the direction of aircraft motion, the Y-axis towards starboard side and the Z-axis points down towards the ground. Quaternion is used in this model because they preclude singularities. An extra numerical integration as compared to Euler angle approach is a negligible computational effort.

2.2 UAV control as a dynamic optimisation problem

Whereas the basic dynamic model of aircraft motion in set of initial values ODEs (Equation (8)), when combined with constraints (prescribed path, states and control constraints) and control saturations, the model takes a form of an index-5 DAE system. This model combined with the optimality condition forms a dynamic optimisation set. The dynamic optimisation problem for aircraft motion can be formulated using this system as follows:

Minimise optimality condition K:

$$\int_0^{t_f} K(x, u, t)dt \qquad \ldots (9)$$

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

such that the plant dynamics are represented as:

$$\dot{x} = f(x, u, t) \quad \ldots(10)$$

path constraints are given by:

$$0 \geq \phi(x, u, t) \quad \ldots(11)$$

subject to bounds:

$$b_{upper} \geq \begin{pmatrix} x \\ u \end{pmatrix} \geq b_{lower}, \quad \ldots(12)$$

$$\left\| \frac{u_{n+1} - u_n}{h} \right\| \leqslant \dot{u}_{max} \quad \ldots(13)$$

and consistent initial conditions:

$$x(t_0) = x_0, \quad \ldots(14)$$

where $x \in \mathbb{R}^m$ denotes state variables; $f : \mathbb{R}^m \times \mathbb{R}^{qc(t)} \times \mathbb{R} \to \mathbb{R}^m$; $u \in \mathbb{R}^{qc(t)}$, $q_c(t) \leq m$ is the vector of controls and $\phi : \mathbb{R}^m \times \mathbb{R}^{qc(t)} \times \mathbb{R} \to \mathbb{R}^q$ is the vector of scleronomic and path inequality constraints, referred to as path constraints only and $t \in [t_o, t_f] \subset \mathbb{R}$ where t_o is initial and t_f is final time. K is a general optimisation function and is represented to complete the formulation of the constraint satisfaction problem.

For the path prescription problem, this model is a high-index DAE when the path constraints becomes active. An aircraft in circle loop manoeuver has been considered an index-5 problem in literature[12]. The solution to the above optimal control problem can be obtained with the discretised modified alpha method combined with a Sequential Quadratic Programming (SQP) solver as explained in the following sections.

Remark 2. *If the optimality condition (Equation (9)) is omitted in the above formulation, then an SQP solver can also be programmed as the only numerical equation solver.*

3.0 SOLVER AND ITS PROPERTIES

3.1 Alpha method DAE solver

The numerical solution of a high-index DAE system is challenging since high index results in high condition number Jacobian matrix leading to solver instability. To date, the most widely used code for the solution of DAE has been DASSL and its variants. However, these codes cannot handle high-index DAEs (more than index-3). These codes also do not have features for handling inequality constraints. The problem under study is of at least index-5.

The approach applied for the results obtained in this paper is direct numerical integration of initial value DAE/DAI system using the alpha method time stepping and at each time step. The SQP method solver detects and satisfies the active path constraints. Like other variables, the control variables are also obtained as unknown variables in the solution, thus fulfilling the inverse simulation objective as brought out in Section 1.2 of this paper.

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

The alpha method is the core of this method, whose formulation is as follows:

$$x_{n+1} = x_n + \left(1 - \frac{\beta}{\gamma}\right) h f(x_n, u_n, w(t_n), t_n) + \frac{\beta}{\gamma} h f(x_{n+1}, u_{n+1}, w(t_{n+1}, t_{n+1}) + \left(\frac{1}{2} - \frac{\beta}{\gamma}\right) h^2 a_n, \quad \ldots (15)$$

$$a_{n+1} = \frac{1}{h\gamma}(f(x_{n+1}, u_{n+1}, w(t_{n+1}, t_{n+1}) - f(x_n, u_n, w(t_n), t_n))) + \left(1 - \frac{1}{\gamma}\right) a_n, \ldots (16)$$

$$0 \geq \phi(x_{n+1}, u_{n+1}, w(t_{n+1}), t_{n+1}), \quad \ldots (17)$$

$$b_{upper_{n+1}} \geq \begin{pmatrix} x_{n+1} \\ u_{n+1} \end{pmatrix} \geq b_{lower_{n+1}}, \quad \ldots (18)$$

where

$$\gamma = \frac{2}{\rho + 1} - \frac{1}{2}, \quad \ldots (19)$$

$$\beta = \frac{1}{(\rho + 1)^2} \quad \ldots (20)$$

$\rho \in [0, 1]$ is a user-selectable regularisation parameter and $h := t_{n+1} - t_n$ is the time step size on the normalised time interval [0, 1] ($t_0 = 0$ and $t_f = 1$). It may be noted that the alpha method approaches the conventional trapezoidal integration method as ρ approaches 0. 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 (Equations (15) and (16)) 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 $a_0 = a(t_0) = \dot{f}_0$. The alpha method and constraint equations are submitted to an SQP solver at each time step (mesh point). Algorithm 1 shows the outline of the method.

It may be noted that the SQP method is not an optimiser but used for detecting and satisfying active path constraints. A local objective function J can be used for this purpose, as defined in Algorithm 1. It is emphasised that we are solving a constraint satisfaction problem and not an optimisation problem.

3.2 Solver properties

Multiple shooting and direct transcription are other popular established techniques for the optimal control problem. The DAI solver finds a feasible solution locally, in contrast to the multiple shooting method or the direct transcription method, where the dynamic optimisation problem discretised over the entire simulation interval enters the QP iterations of an SQP solver. For the aircraft prescribed path control and path feasibility studies, finding this local solution suffices. The method proposed is intended to be either a computationally cheap tool for generating feasible initial guess for the dynamic optimisation problem solved by multiple shooting or direct transcription with an interior

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

point method, or to be a solver for rapid trajectory planning using continuous constraint satisfaction.

The method used has a parameter ρ in Equation (14) and (15) that is varied in order to overcome numerical row rank deficiency and hence increases the smallest singular values (i.e. regularisation) of basis matrix.

3.3 Note

The regularisation process achieved through the proposed method is explained as follows. For a system of the form:

$$\dot{x}(t) = f(x, y, t), \quad \ldots(21)$$

$$0 = g(x, y, t) \quad \ldots(22)$$

Algorithm 1 UAV path prescription and regularisation

Initialise the solver with optimisation options: Max Iters, Max FunEv, TelCon, Tolfun
Define t_o and t_f as start and end simulation time
Define starting values of h and ρ
Define initial conditions for aircraft 6-DOF equations and aero data set
Check for consistency in initialisations
if inconsistent **then**
  flash error
end If; exit
while $t_0 < t_f$ **do**
Solve using SQP method following constraint satisfaction problem
  $\underset{u_{i+1}, x_{i+1}}{min} J(u_{i+1}, x_{i+1}) = 0$ such that
  $x_{i+1} = x_i + \left(1 + \frac{\beta}{\gamma}\right)h_i f_i + \frac{\beta}{\gamma} h f_{i+1} + \left(\frac{1}{2} - \frac{\beta}{\gamma}\right)h_i{}^2 a_i$
  $a_{i+1} = \frac{1}{h_i \gamma}(f_{i+1} - f_i) + \left(1 - \frac{1}{\gamma}\right)a_i$
  $0 \geq \phi(x_{i+1}, u_{i+1}, f_{i+1})$
  $u_{lower} \leq u \leq u_{upper}$
  if local minimum found **then**
    $t_i = t_i + h$ and use the current outputs as next guess values
else
  $\rho = \rho + \epsilon$
  if $\rho = 0$ **then**
    $h = h_{max}$
  end if
  end if
end while
Save values in data file for analysis and stop the simulation

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

The iteration matrix is of the form (refer to Ref. 17):

$$hJ_n = \begin{bmatrix} (\alpha_0 I - h\frac{\partial f}{\partial x}) & -h\frac{\partial f}{\partial y} \\ h\frac{\partial g}{\partial x} & h\frac{\partial g}{\partial y} \end{bmatrix} \quad \ldots (23)$$

The Jacobian of the DAI system consists of differential, algebraic and algorithmic variables at each time step (mesh point). The numerical rank of the basis matrix needs to be full for invertibility.

Theorem 3. *The condition number of the iteration matrix for a system with (local) index of* υ *is* $O(h^{-\upsilon})$ *(see Section 5.4.2 in Ref. 8 for proof).*

When an ill-conditioned basis matrix is detected, the present DAI solver tries to regularise the basis by varying ρ. Under certain conditions, a regularised and invertible basis can be obtained. (Refer to Ref. 17 for more details).

Remark 4. *For a system of the form*

$\dot{x}_t = f_t$

$g(x_t) = 0$

if (x_0, u_0) *is satisfying* $\dot{x}_0 = f_0$, $g(x_0) = 0$, *then, by implicit function theorem,*

$\tau \overset{Lt}{\rightarrow} 0^+ \begin{pmatrix} \tau - \frac{\delta f}{\delta x}\tau & -\frac{\delta f}{\delta u}\tau \\ \frac{\delta g}{\delta x} & 0 \end{pmatrix}$ *must be invertible.*

For a unique solution in an open set containing (x_0, u_0) *over* $[t_0, t_0 + \tau]$, $\frac{\delta g}{\delta x}$ $\frac{\delta f}{\delta u}$ *is not invertible for high index. Thus, we need to regularise. Differentiation with index reduction involves a lot of tuning variables, stabilisation and unnecessary smoothening of control. Also,* $g(x_t) = \epsilon$ *is not preferred because it violates path constraint.*

Thus, we take

$\dot{x}(t) = f(t) + \eta(t)$

$g(x(t)) = 0$

Such that $\|\eta(t)\| \ll \|f(t)\|$, *which is introduced by a very small change in* $u(t)$

This small change can be given as $\eta(t) = \gamma da_t$, *where*

$$a_t \approx e^{-\frac{1}{\gamma}t} a_0 + \int_0^t e^{-\frac{1}{\gamma}(t-s)} df_s \quad \ldots (24)$$

such that $df_s = \left(\frac{\delta f}{\delta x} f\, d_s + \frac{\delta f}{\delta u} d_u\right)$ *and* $\|a_t dt\| = O(dt^2)$.

A rank deficient basis is an indication of no feasible solution. The rank deficient Jacobian fails to attain SQP convergence. Sharp changes in the prescribed path may also cause this to happen. The failure of convergence is a direct indication of non-feasibility of the manoeuver under the prescribed constraints. Therefore, this approach can be easily applied to generate set of safe trajectories for the UAV.

4.0 SIMULATION STUDIES

4.1 Path prescription

For a Proof of Concept (POC) study for this approach, the circle manoeuver and a random movement of UAV are considered. Circle is most common trajectory taken by

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

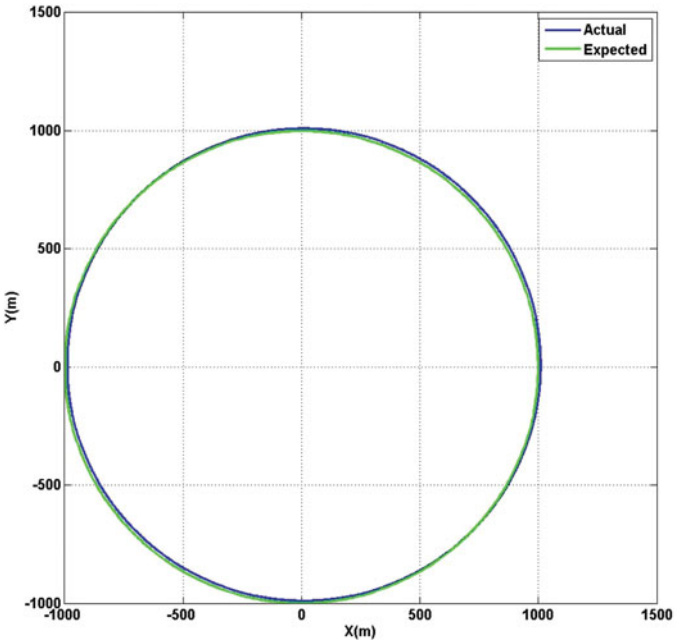


Figure 3. (Colour online) UAV trajectory during circle manoeuver (radius = 1,000 m).

UAV during surveillance operations, as it easily meets the payload operation specifications. The path constraint requires the aircraft to move on a circular trajectory at fixed altitude defined by:

$$(x - x_0)^2 + (y - y_0)^2 = R^2, \quad \ldots (25)$$

where x_0, y_0 are the coordinates of the center of circle and R is its radius. The aircraft needs to make a circle at almost a constant bank angle (ϕ) and very low sideslip angle (β). This trajectory is to be obtained such that aircraft attitudes and aircraft body rates are within the safe limits. The control variables (elevator, aileron, rudder and throttle) and state variables such as roll angle and body rates are suitably constrained within respective desired min-max bands. The SQP solver is programmed with suitable constraint tolerance limits and maximum iteration limits for convergence. The digitised set of equations are submitted to the SQP solver as a *constraint satisfaction problem* at every time step to ensure local convergence. The optimality condition is chosen to minimise the change in control from one step to other. This is done to minimise control chatter. The outputs of one iteration are the guess values for the next mesh point. The UAV model discussed in Section 2 is numerically integrated with the alpha method and submitted to the SQP solver. In this way, a local solution is obtained at every mesh point. The ρ and h values are varied as per the algorithm described to ensure regularisation at each mesh point. The procedure is repeated till the UAV completes the prescribed trajectory successfully or no solution message starts coming consecutively for a predefined number of simulation iterations.

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

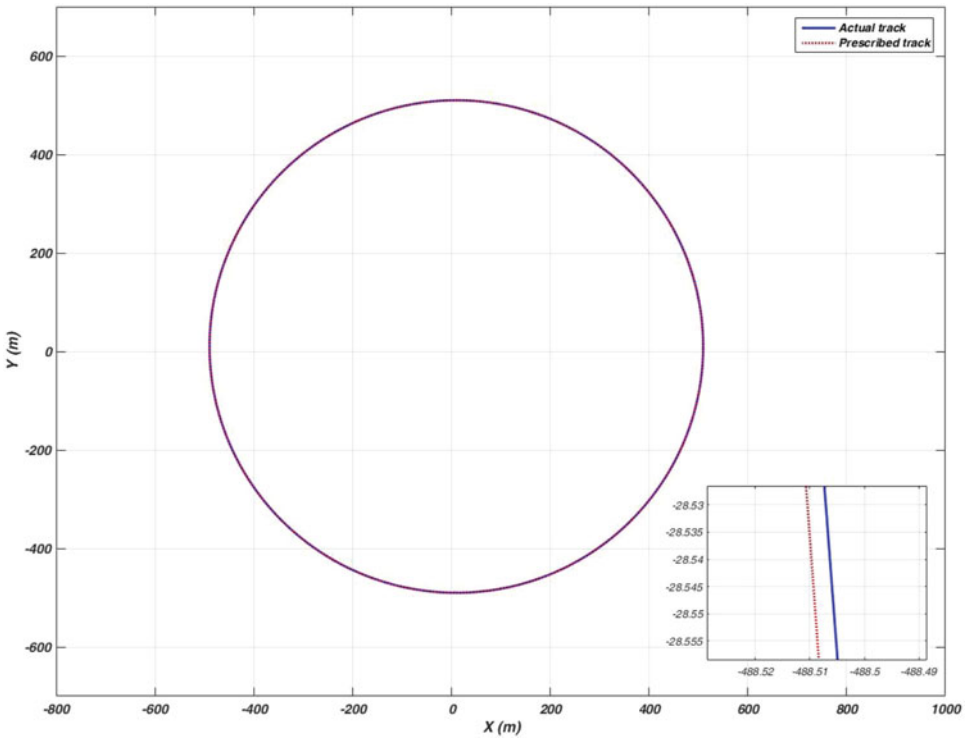


Figure 4. (Colour online) UAV trajectory during circle manoeuver (radius = 500 m).

4.2 Results

In the first simulation, the UAV was directed to fly along a predefined path of circular trajectory of 1,000-m radius starting from initial points. The required trajectory is defined in Equation (25). Inverse simulation has been carried out for 260 sec. The UAV flight dynamics model was coded in Matlab software version 2014a, and the model is executed on a dual core 2.4 Ghz workstation, with 8 GB RAM on a 64-bit Windows 7 operating system. Figures 5 and 6 show various aircraft parameters and trajectory taken by the UAV during path prescribed control of the 1,000-m radius circle. A close match between prescribed and actual trajectory can be seen in Fig. 3. It was observed that the aircraft completes the motion along the defined path within safe values of angle-of-attack (α) (stall angle being less than 11°). Aircraft speed and other critical parameters such as Euler angles and body rates are shown in Fig. 5. It may be noted that the aircraft maintains roll and sideslip (less than 2°) within acceptable limits and completes circular trajectory. Small chatter is observed at few mesh points. The results can be improved upon with better constraint formulation. When under the same constraints the circle manoeuver was tried at very high altitude, the simulation failed due to non-availability of trim elevator showing non-feasibility. Table 1 shows the various h and ρ combinations; S indicates success and F indicates failure. In another simulation, the radius of the circle was prescribed as 500 m and the aircraft moved on the this prescribed circular trajectory; the resultsare shown in Figs 4, 7–9. When the aircraft was prescribed to from a circle of less than 300 m, the simulation failed indicating that the circle of this radius cannot be made under the prescribed constraint values. The random trajectory prescription results are shown in Figs 11

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

Table 1
Simulation outcome with different ρ and *h* values

	Simulation with combinations for ρ and *h*			
h	**ρ = 0**	**ρ = 0.25**	**ρ = 0.5**	**ρ = 1**
0.04	F	F	F	F
0.05	S	F	F	F
0.075	S	F	S	F
0.1	S	F	S	F

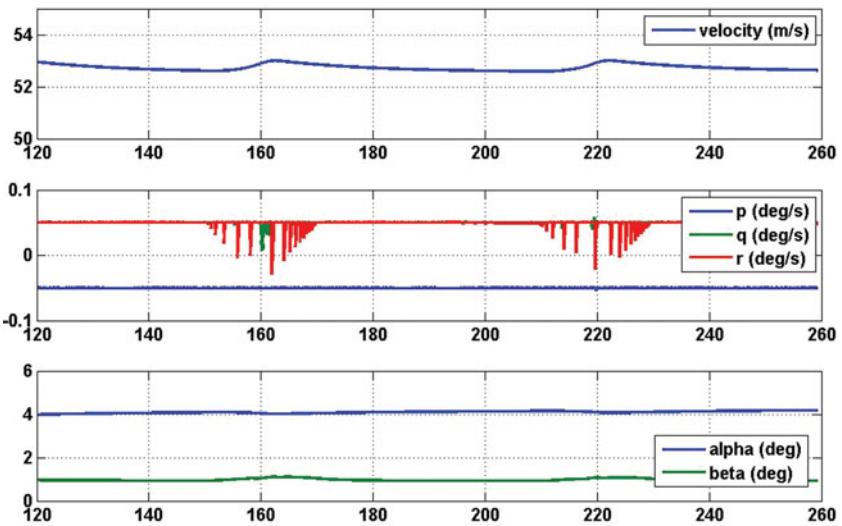


Figure 5. (Colour online) Velocity and other aircraft parameters during circle manoeuver (radius = 1,000 m).

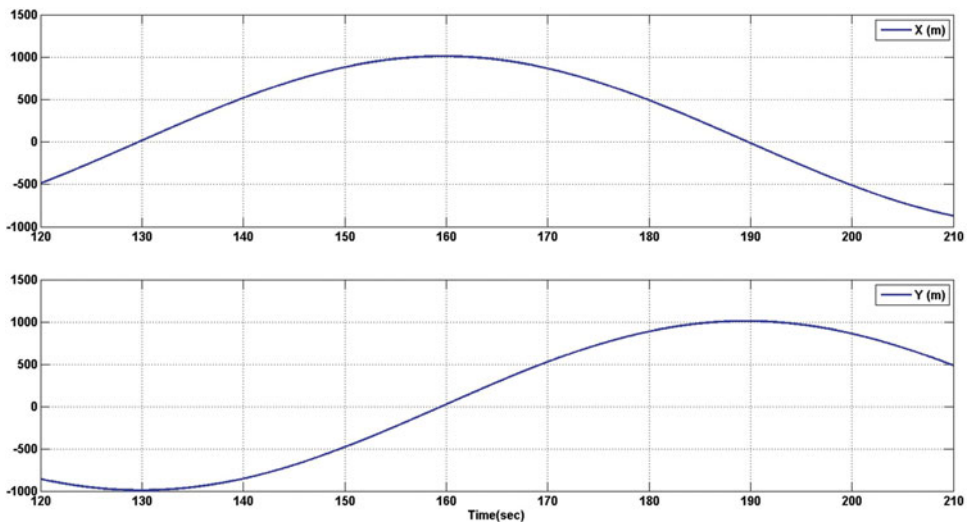


Figure 6. (Colour online) X and Y position of aircraft during circle manoeuver (radius = 1,000 m).

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

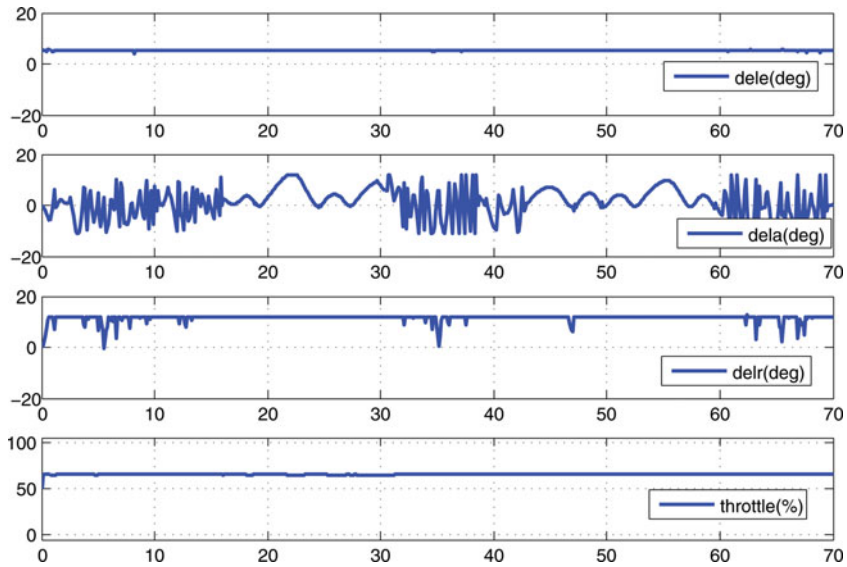


Figure 7. (Colour online) Control parameters during 500 m radius circle.

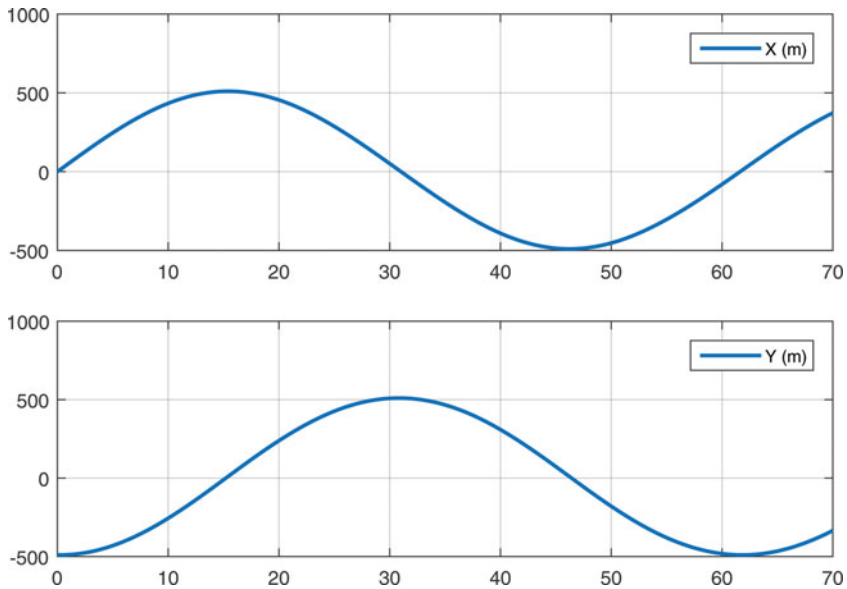


Figure 8. (Colour online) X and Y during 500 m radius circel.

and 12. These trajectories are shown as proof of concept for this method; the approach is applicable to any other trajectory represented as polynomial/spline equation in state variables. The results shown in the figures herein have been normalised. A failed trajectory tracking is shown in Fig. 10.

Note 1 *If the Newton iteration fails to converge at a mesh point, if conventional strategy to reduce h is applied, then it would lead to increasing ill conditioning of Jacobian (Refer*

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

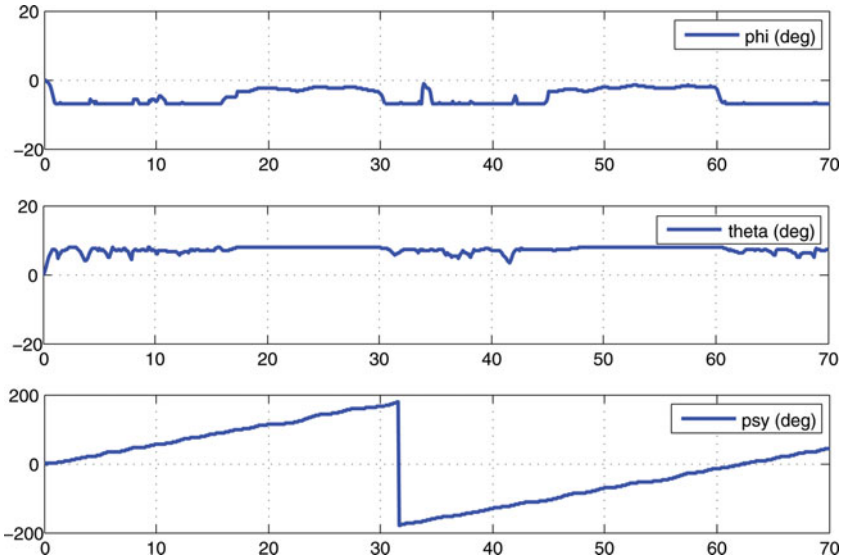


Figure 9. (Colour online) Euler angles during 500 m radius circle.

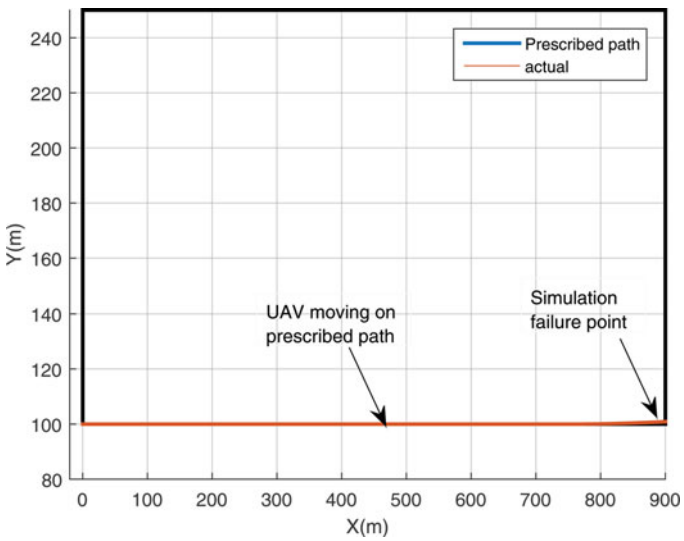


Figure 10. (Colour online) A non-feasible trajectory.

Theorem 5.4.1 in Ref. 8). Thus, ρ is the regularisation handle that can be varied to decrease the condition number of the Jacobian matrix. A low h cannot always guarantee a solution for a DAE system as obtained in Table 1.

Also, it may be noted that the path prescription problem does not have a unique solution and varies with ρ and *h* used for the simulation. In another simulation, the UAV was prescribed to fly on a path defined by a degree-3 polynomial. The results are shown in Figs 11 and 12. However, when a trajectory as shown in Fig. 10 was prescribed, the solver failed at the corner

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

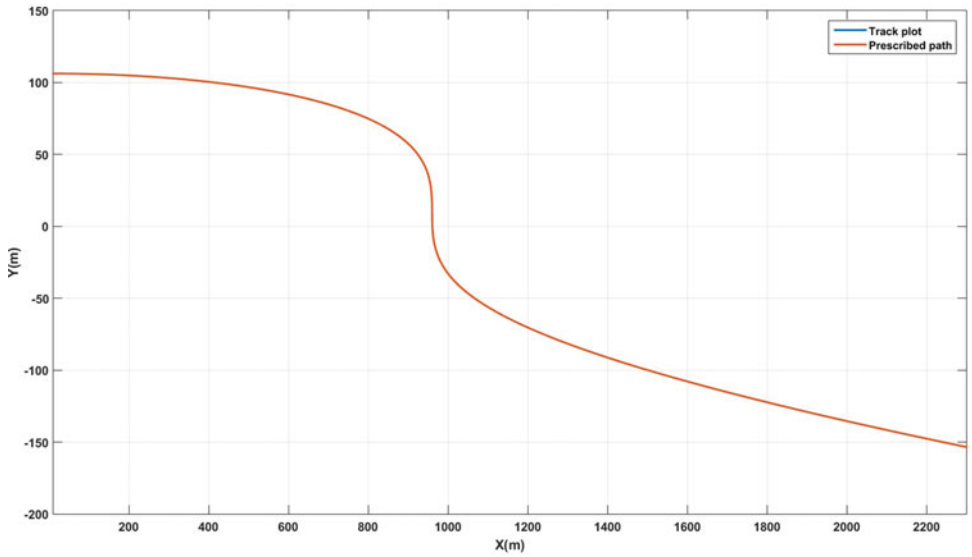


Figure 11. (Colour online) Prescribed and actual trajectory of aircraft during a random manoeuver.

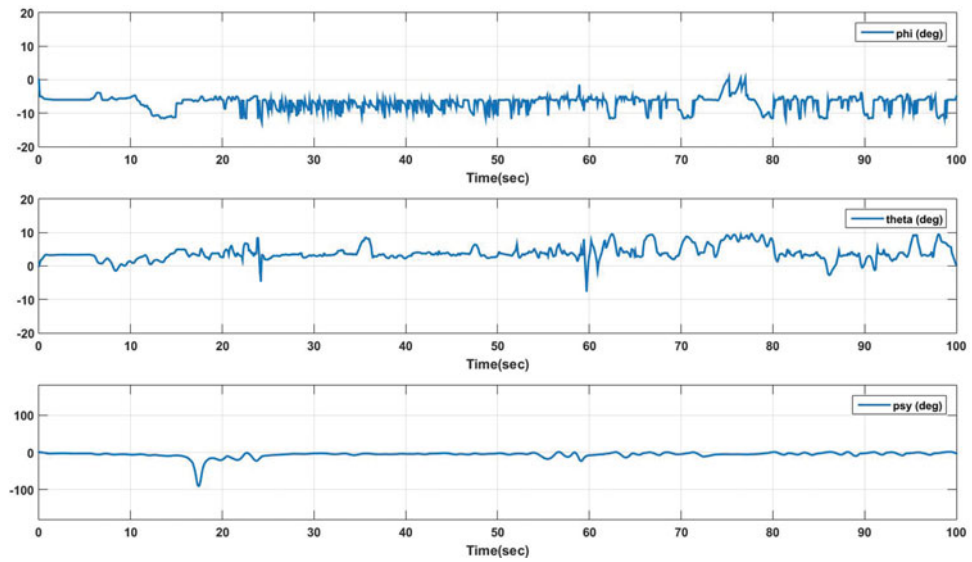


Figure 12. (Colour online) Parameters during random manoeuver.

point, indicating no feasible solution and displaying a low rank error. The large amount of chatter is observed due to high index at the point, and finally the SQP fails to converge due to the ill-conditioned Jacobian working set despite the best regularisation attempt. This showed that the prescribed trajectory shall not be feasible within the specified bounds in the given problem. Thus, the corner points needs to modelled as an arch.

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

5.0 CONCLUSION AND FUTURE SCOPE OF WORK

Trajectory optimisation via path prescription for a UAV has been demonstrated in this paper using the DAE approach. The numerical method for solving the resultant high-index DAE problem and regulariaation process have also been described. The following advantages are cited for this novel approach:

- It can be used for path planning for UAV and takes into account complete dynamical model of UAV.
- This approach can be used for UAV flight simulation at conceptual stages when control law of aircraft is not ready so as to ascertain possibilities of control saturation, limit exceedance, etc.
- This approach can be used with the traditional control laws based on classical control theory to improve their performance and can also be used as back-up mode information when stored as a look-up table in the on-board flight control computer.

The following enhancements are possible in the future:

- In the present work, delays in the control system due to uplink and downlink, actuator lags, and so on are not included in the simulation model because the current solver cannot handle delays. This solver needs to be extended to handle delays giving rise to Delayed Differential Algebraic Inequality (DDAI).
- Real-time simulation can be achieved so as to use this approach onboard and during pilot/operator in the loop simulation studies. By exploiting the matrix sparsity of constraint Jacobian, the method can be speeded up to complete speedy SQP convergence in the Matlab environment.
- Chatter observed in the control solution at few a points can be improved by revising the plant model and refinement of the algorithm.
- Wind effects on flight dynamics have been ignored to keep the formulation simple. The effect of wind can be studied separately.

ACKNOWLEDGEMENTS

The authors are thankful to Director Aeronautical Development Establishment (ADE) for granting permission to publish this research work. The authors would also like to thank Shri. Sushil Mohan Ratnaker, Shri. D Bhattacharya, Shri. Rakesh G for technical discussions on the aircraft modelling and manoeuvres and Shri. Sreenesh for helping with the word processor.

REFERENCES

1. MURRAY-SMITH, D.J. Feedback methods for inverse simulation of dynamic models for engineering systems applications, *Mathematical and Computer Modelling of Dynamical Systems*, 2011, **17**, (5), pp 515-541.
2. GOSZCZYŃSKI, J.A., BLAJER, W. and KRAWCZYK, M. The inverse simulation study of aircraft flight path reconstruction, *Transport*, 2002, **XVII**, (3), pp 103-107.
3. AZAM, M. and SINGH, S.N. Invertibility and trajectory control for nonlinear maneuvers of aircraft, *J. Guidance, Control, and Dynamics*, 1994, **17**, pp 192-200.

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press

4. KATO, O. and SUGIURA, I. An interpretation of aircraft general motion and control as inverse problem, *J. Guidance*, 1986, **9**, pp 198-204.
5. LANE, S.H. and STENGEL, R.F. Flight control design using non linear inverse dynamics, *Automatica*, 1988, **24**, pp 471-483.
6. SNELL, S.A. and STOUT, P.W. Flight control law using nonlinear dynamic inversion combined with quantitative feedback theory, *J. Dynamic Systems, Measurement, and Control*, 1998, **120**, pp 208-215.
7. FERON, E., POPOVIC, J., DEVER, C., METTLER, B. and MCCONLEY, M. Nonlinear trajectory generation for autonomous vehicles via parameterized maneuver classes, *J. Guidance, Control, and Dynamics*, 2006, **29**, (2), pp 289-302.
8. CAMPBELL, S.L., BRENAN, K.E. and PETZOLD, L.R. Numerical solution of initial-value problems in differential-algebraic equations, *Classics Applied Mathematics* , volume 4, 2nd revised ed, 1996, SIAM, Philadelphia, Pennsylvania, US.
9. HAIRER, E. and WANNER, G. Solving ordinary differential equations 2. stiff and differential-algebraic problems, *Springer Series in Computational Mathematics*, volume 14, 2nd revised ed, 2004, Springer-Verlag, Berlin, Germany.
10. ASCHER, U.M., SPITERI, R.J. and PAI, D.K. Programming and control of robots by means of dai, *IEEE Transcations on Robotics and Automation*, 2000, **16**, (2), pp 135-145.
11. BARRLUND, A, Numerical Solution of Higher Index Differential-Algebraic Systems, PhD Thesis, 1991, Department of Scientific Computing, Umea University, Umea, Sweden.
12. CAMPBELL, S.L. High-index differential algebraic equations, *J. Mechanics of Structures and Machines*, 1995, **23**, pp 199-222.
13. CAMPBELL, S.L., ENGELSONE, A. and BETTS, J.T. Direct transcription solution of higher-index optimal control problems and the virtual index, *Applied Numerical Mathematics*, 2007, **57**, pp 281-296.
14. PRYCE, J.D. and NEDIALKOV, N.S. Solving daes by taylor series: Daets code, *J. Numerical Analysis, Industrial and Applied Maths*, 2008, **3**, pp 61-80.
15. CHUNG, J. and HULBERT, G.M. Time integration algorithm for structural dynamics with improved numerical dissipation: The generalized- alpha method, *Transactions of ASME J. Applied Mechanics*, 1993, **60**, (2), pp 371-375.
16. PETZOLD, L., YEN, J. and RAHA, S. A time integration algorithm for flexible mechanism dynamics: The dae-alpha method, *Computer Methods in Applied Mechanics and Engineering*, 1998, **158**, pp 341-355.
17. CHANDRA PARIDA, N., PETER, J. and RAHA, S. The alpha-method for solving differential algebraic inequality *dai* systems, *International Journal of Numerical Analysis and Modeling*, 2010, **7**, (2), pp 240-260.
18. PARIDA, N.C. and RAHA, S. The alpha-method direct transcription in path constrained dynamic optimization, *SIAM J. Scientific Computing*, 2009, **31**, (3), pp 2386-2417.
19. BLAJER, W. and KOLODZIEJCZYK, K. A geometric approach to solving problems of control constraints: Theory and a dae framework, *Multbody System Dynamics*, 2004, **11**, pp 343-364.
20. RAHA, S. and PETZOLD, L.R. Constraint partitioning for stability in path-constrained dynamic optimization problems, *SIAM J. Scientific Computing*, 2001, **22**, pp 2051-2074.
21. RAHA, S. and PETZOLD, L.R. Constraint partitioning for structure in path-constrained dynamic optimization problems, *Applied Numerical Mathematics*, 2001, **39**, pp 105-126.
22. KOLODZIEJCZYK, K. and Blajer, W. A geometrical approach to solving problems of control constraints theory and a dae framework, *Multibody System Dynamics*, 2004, **11**, (4), pp 343-364.
23. MURRAY, W., GILL, P.E. and SAUNDERS, M.A. Snopt: An sqp algorithm for large-scale constrained optimization, *SIAM J. on Optimization*, 2002, **12**, pp 979-1006.
24. ALLGÖWER, F., FINDEISEN, R., IMSLAND, L. and FOSS, B. State and output feedback nonlinear model predictive control: An overview, *European J. Control*, 2003, **9**, (2-3), 190-206.

https://doi.org/10.1017/aer.2017.36 Published online by Cambridge University Press