TRAJECTORY GENERATION ON APPROACH & LANDING FOR A RLV USING NOC APPROACH

JAYAKRISHNAN, S

R HARIKUMAR

M.Tech(2010-2012), Deppartment of Electrical Engineering, College of Engineering Trivandrum, Thiruvananthapuram

Email: jksv88@gmail.com

Abstract—A major objective of next generation reusable launch vehicle (RLV) programs includes significant improvements in vehicle safety, reliability, and operational costs. In this paper, trajectory generation on approach/landing (A&L) for RLVs using motion primitives (MPs) and neighboring optimal control (NOC) is discussed. The proposed trajectory generation approach at A&L phase is based on an MP scheme which consists of trims and maneuvers. From an initial point to a given touchdown point, all feasible trajectories that satisfy certain constraints are generated and saved into a trajectory database. An optimal trajectory can then be found off-line by using Dijkstra’s algorithm. After perturbations are imposed on the initial states of the off-line optimal trajectory, it is reshaped into a neighboring feasible trajectory on-line by using NOC approach. At this point, a neighboring feasible trajectory existence theorem (NFTET) is investigated and its proof is provided as well. The results show that the vehicle with stuck effectors can be recovered from failures in real time. Finally, robustness issues on NOC approach are briefly discussed.

I. INTRODUCTION

The increased demand for commercial and military utilization of space is a substantial driver for the development of new technologies to improve space vehicle economics. Reusable launch vehicles (RLVs) have the potential to increase space launch efficiencies far beyond those achieved by current systems. Second generation (and future generation) RLVs may eventually take the place of the space shuttles, but not before scientists perfect the technologies that make RLVs safer, more reliable, and less expensive than the shuttle fleet. To achieve this goal, a variety of new RLV trajectory design approaches have recently been proposed. Generally, a RLV mission is composed of four major flight phases: ascent, re-entry, terminal area energy management (TAEM), and approach and landing (A&L). Some results on trajectory generation in ascent and re-entry phases were presented in [1]. Some methods of trajectory planning for TAEM were presented in [2]. A&L is a critical flight phase that brings the unpowered vehicle from the end of the TAEM phase to runway touchdown. A few trajectory design approaches in A&L phase were discussed in [3]. The goal of this paper is to develop new approaches that can deliver a RLV to its landing site safely and reliably, recover the vehicle from some failures, and avoid mission abort as much as possible. A feasible trajectory database under nominal conditions for the RLV is generated by using motion primitive (MP) scheme. Neighboring optimal control (NOC) is used to generate neighboring feasible trajectories in order to recover the vehicle from some failure scenarios. This approach demonstrates good robust performance of the RLV system when system uncertainty and terminal disturbance are not severe.

Section II introduces a motion primitive scheme and shows how to generate feasible trajectories in the A&L phase based on the MP scheme. Section III shows that the trajectory retargeting can recover the vehicle from some failures by using NOC approach. The robust performance of NOC is briefly discussed in Section IV. Some results and discussions are given in Section V. The conclusions are drawn in Section VI.

II. TRAJECTORY GENERATION ON APPROACH AND LANDING USING MOTION PRIMITIVES

This section will first introduce the point mass equations of motion for A&L problem of a RLV and then briefly describe MP scheme. After that, how to generate feasible trajectories using MP scheme is discussed.

A. Point Mass Equations of Motion

For an unpowered RLV on A&L, it is assumed that the sideslip angle to be zero and that symmetric flight conditions exist. Therefore, only longitudinal motion is considered. The gliding flight in a vertical plane of symmetry is then defined by the following point mass equations

\[ V = \left( \frac{1}{W} - \sin \gamma \right) g \]  
\[ \dot{\gamma} = \left( \frac{L}{W} - \cos \gamma \right) \frac{g}{V} \]  
\[ \dot{h} = V \sin \gamma \]  
\[ \dot{x} = V \cos \gamma \]

where \( V \) is the vehicle velocity, \( \gamma \) is the flight-path angle, \( h \) is the altitude, \( x \) is the downrange, \( g \) is the gravitational acceleration, \( W \) is the vehicle weight, \( L \) is the lift, and \( D \) is the drag. \( L \) and \( D \) can be represented as

\[ L = \frac{1}{2} \rho \cdot C_L \cdot s \cdot h \]  
\[ D = \frac{1}{2} \rho \cdot C_D \cdot s \cdot h \]

where \( \rho \) is the dynamic pressure, \( s \) is the aerodynamic reference area of the vehicle, \( C_L \) and \( C_D \) are the lift and drag coefficients, respectively. The dynamic pressure \( \frac{1}{2} \rho \cdot \dot{h} \) is given by

\[ \frac{1}{2} \rho \cdot \dot{h}^2 \]

where \( \rho \) is the air density at the altitude \( h \) which is approximated using an exponential model

\[ \rho = \rho_0 e^{-3h} \]
where \( \rho_0 \) is the air density at sea level and \( \beta \) is the atmospheric density scale.

Generally, the lift coefficient \( C_L \) is a linear function of \( \alpha \), where \( \alpha \) is angle of attack and the drag coefficient \( C_D \) is a quadratic function of \( C_L \), namely,
\[
C_L = C_{L0} + C_{L\alpha \alpha} \alpha \\
C_D = C_{D0} + K C_L^2
\]
(9) (10)
where \( C_{L0} \) is the lift coefficient at zero angle of attack, \( C_{L\alpha \alpha} \) is the lift slope coefficient, \( C_{D0} \) is the drag coefficient at zero lift, and \( K \) is a coefficient relative to induced drag.

Substituting (9) into (10) gives \( C_D \) as a function of \( \alpha \)
\[
C_D = k_{D0} + k_{D\alpha} \alpha + k_{D\alpha \alpha} \alpha^2
\]
(11)
where \( k_{D0}, k_{D\alpha}, \) and \( k_{D\alpha \alpha} \) are resulting coefficients with respect to \( \alpha \).

In the state equations (1)-(4), \( V, \gamma, h, \) and \( x \) are the four state variables and \( \alpha \) is the control variable.

The constraints at touchdown are
\[
h_{TD_{min}} \leq h_{TD} \leq h_{TD_{max}}
\]
(12)
\[
V_{TD_{min}} \leq V_{TD} \leq V_{TD_{max}}
\]
(13)
where \( h_{TD} \) is the sink rate at touchdown, \( h_{TD_{min}} \) and \( h_{TD_{max}} \) are its minimum and maximum values, respectively, \( V_{TD} \) is the touchdown velocity, and \( V_{TD_{min}} \) and \( V_{TD_{max}} \) are its minimum and maximum values, respectively.

Another constraint could be the Mach number over the trajectory
\[
M_{min} \leq M \leq M_{max}
\]
(14)
where the Mach number \( M \) can be easily calculated from the vehicle velocity and the atmospheric temperature [14].

B. Motion Primitives

A motion plan consists of two classes of MPs [15]. The trajectory generation in this paper involves transitioning from one class of MPs to the other. The first class of MPs is a special class of trajectories, known as trims. A trim is a steady-state or quasisteady flight trajectory which includes a set of steep glideslopes and a set of shallow glideslopes on A&L. The second class of MPs consists of transitions between trims. These trajectory segments are known as maneuvers. A maneuver is an “unsteady” trajectory which includes pull-up and flare maneuvers on A&L. A pull-up maneuver transitions a steep glide to a shallow glide and a flare maneuver is added after the shallow glide to dissipate the energy of the impact at landing to touchdown with an acceptable sink rate for the vehicle [16]. From an initial point to a given touchdown point, all feasible trajectories which satisfy certain constraints are generated and saved into a trajectory database [12].

C. Trajectory Planning under Nominal Conditions

1) Trims: The steady or quasi-steady approach is applied during trims. Since the flight path angle \( \gamma \) is a constant, \( \tan \gamma = \frac{\gamma}{h} \) can be easily calculated. If the altitude \( h \) is chosen to be the independent variable during trims, and note that the gliding is in a constant flight path angle, (1)-(4) now become [12]
\[
\frac{dV}{dh} = \frac{(-D/W - \sin \gamma)g}{V \sin \gamma}
\]
(15)
\[
\frac{dx}{dh} = \frac{1}{\tan \gamma}
\]
(16)
From \( \frac{\gamma}{h} = 0 \), (2) becomes
\[
L = W \cos \gamma
\]
(17)
Substituting (5) into (17) gives the lift coefficient
\[
C_L = \frac{W \cos \gamma}{gS}
\]
(18)
where \( \dot{q} \) is from (7).
From (9), the control input \( \alpha \) can now be evaluated
\[
\alpha = \frac{C_L - C_{L0}}{C_{L\alpha \alpha}}
\]
(19)
In (15), the drag \( D \) can be obtained from (6), (7), and (11) once \( \alpha \) is acquired.

2) Maneuvers: For the maneuvering flights, the nonsteady approach has to be used. For convenience, time is not chosen to be the independent variable. Instead, \( \gamma \) can be used as the independent variable during pull-up. For the pull-up maneuver, (1)-(4) now become [12]
\[
\frac{dV}{dt} = \frac{V(-D/W - \sin \gamma)}{V^2 \sin \gamma}
\]
(20)
\[
\frac{dh}{dt} = g(L/W - \cos \gamma)
\]
(21)
\[
\frac{dx}{dt} = \frac{V^2 \cos \gamma}{g(L/W - \cos \gamma)}
\]
(22)
Since the load factor \( n_x = \frac{V}{h} \) is fixed during the pull-up, the lift coefficient \( C_L \) can be calculated by the following equation
\[
C_L = \frac{n_x W}{gS}
\]
(23)
The control \( \alpha \) is then obtained by (19).

The flare maneuver is modeled by a cubic polynomial that gives the altitude as a function of downrange [12]
\[
h(x) = ax^3 + bx^2 + cx + d
\]
(24)
where \( a, b, c, \) and \( d \) are constants to be determined from the desired initial altitude of the flare and the amount of downrange that is desired for the maneuver. Without loss of generality, it is assumed that the maneuver starts at \( x = 0 \). Letting \( x_3 \) be the downrange at which the flare starts (recall that \( x_3 = 0 \)), it turns out that \( h_3 = h(x_3) \) and \( \tan \gamma_3 = \tan \gamma_2 \) where \( \gamma_2 \) is the flight path angle for shallow glide. Given \( h(x_{TD}) = 0 \) and \( \frac{dh}{dx} = \tan \gamma_{TD} \) where \( \gamma_{TD} \) is the flight path angle at touchdown, the coefficients of the cubic polynomial are as follows
\[
a = 2h_3 + (\tan \gamma_2 + \tan \gamma_{TD})x_{TD}
\]
(25)
\[
b = -3h_3 + (2 \tan \gamma_2 + \tan \gamma_{TD})x_{TD}
\]
(26)
\[
c = \tan \gamma_2
\]
(27)
\[
d = h_3
\]
(28)

Similarly, \( x \) is chosen to be the independent variable during flare maneuver. According to (24)-(28), (1)-(4) now become
\[
\frac{dV}{dx} = \frac{g(-D/W - \sin \gamma)}{V \cos \gamma}
\]
(29)
\[
\frac{dy}{dx} = \frac{\cos \gamma}{d\gamma} \frac{d^2 h}{dx^2}
\]
(30)
\[
\frac{dh}{dx} = \tan \gamma = 3ax^2 + 2bx + c
\]
(31)
When a failure significantly affects the forces on a RLV, or modifies flight envelope constraint boundaries, trajectory retargeting may be used to recover it. Failures considered here correspond to the case of a stuck effector so that the nominal states and controls can be used to reshape trajectory [17]. Among the feasible trajectory database, an optimal one is always assumed to be found off-line. When a failure occurs, NOC is used to determine its neighboring feasible trajectory on-line. If such a neighboring feasible trajectory exists, the RLV can then be recovered from that failure in real time.

### III. RAJECTORY RETARGETING USING NEIGHBORING OPTIMAL CONTROL

When a failure significantly affects the forces on a RLV, or modifies flight envelope constraint boundaries, trajectory retargeting may be used to recover it. Failures considered here correspond to the case of a stuck effector so that the nominal states and controls can be used to reshape trajectory [17]. Among the feasible trajectory database, an optimal one is always assumed to be found off-line. When a failure occurs, NOC is used to determine its neighboring feasible trajectory on-line. If such a neighboring feasible trajectory exists, the RLV can then be recovered from that failure in real time.

#### A. Off-line Trajectory Optimization

A shortest path problem in computer graph and algorithms theories is used for the off-line optimal trajectory search. At this point, the source is fixed while the target is unknown. To find the shortest path, several algorithms can be used [18], [19]. Dijkstra’s algorithm is a reasonable choice since it is a best-first search and uses greedy strategy. It is guaranteed to find an optimal path by repeatedly selecting the vertex with the minimum shortest-path estimate or cost. The cost function or weight is defined by the mean value of \( \alpha = \alpha^* \) where \( \alpha^* \) is a nominal value of angle of attack \( \alpha \), e.g., \( \alpha^* = 5^\circ \) which corresponds to the base pitching moment coefficient \( C_m \) = 0 due to the wing body. During search, a priority queue is used.

#### B. On-Line Trajectory Optimization Using NOC

Based on the off-line optimal trajectory, a small perturbation is imposed on each state variable and new control can be obtained by NOC. The RLV can be recovered from the aforementioned failures with stuck effectors.

1. **NOC Approach Description:** The NOC approach is discussed in [20]-[22]. However, the system parameters are under consideration in this paper and it is necessary to make some changes.

Considering the following general nonlinear system

\[
\dot{x}(t) = f(t, x(t), u(t), \lambda(t)), \quad x(t_0) = x_0
\]  

(34)

where \( x(t) \in C \) are state variables, \( u(t) \in \Omega \) are control inputs, \( \lambda(t) \in \Omega \) contains system parameters, \( C \subset R^n, \Omega \subset R^m, \) and \( \Omega \subset R^p \). With the convex cost function

\[
J(u) = \phi(t_f, x_f, \lambda_f)
\]  

(35)

where the subscript “f” denotes the final time, it is assumed that the optimal solution \( (x^*(t), u^*(t)) \) can be found, where the optimal control

\[
u^*(t) = g(t, x^*(t), \lambda(t))
\]  

under the state constraints and control constraints

\[
x_{\text{min}} \leq x^*(t) \leq x_{\text{max}}
\]  

(37)

\[
u_{\text{min}} \leq u^*(t) \leq u_{\text{max}}
\]  

(38)

When small perturbations \( \delta x(t) \) and small variations \( \delta u \) are imposed on initial states \( x_0 \) and system parameters \( \lambda \), respectively, the neighboring feasible trajectory \( z(t) \) and new optimal control \( v(t) \) can then be determined by \( \delta x(t), \delta u, \lambda, \) and \( \delta x(t) \). Since system parameters are considered in the NOC approach of this paper, the following theorem is then investigated.

1. **Neighboring Feasible Trajectory Existence Theorem (NFTET):** It requires two following assumptions.

   **Assumption 1:** For the nonlinear system (34), the following hypotheses must be followed:

   (a) \( f \) is continuous in \((t, x, \lambda)\);

   (b) \( f \) is local Lipschitz in \( x \) and uniformly Lipschitz in \( t \) and \( \lambda \);

   (c) \( C \subset R^n, \Omega \subset R^m, \) and \( P \subset R^p \) are convex, connected sets.

   **Assumption 2:** The open-loop optimal solution \( z(t) \) exists and optimal control \( u^*(t) \) is well defined for the system (34).

   **Theorem 1 (NFTET):** (1) For an initial condition \( z(t_0) \), which can be equal to \( x_0 \), there exist neighboring feasible state trajectories \( z(t) \) and neighboring optimal control \( v(t) \), satisfying

\[
sup_t \|z(t) - z^*(t)\| \leq K
\]  

(39)

where \( K \) is an upper bound.

(2) When the system parameters change from \( \lambda_1 \) to \( \lambda_2 \) \( (\lambda_2 = \lambda_1 + \delta \lambda) \), the neighboring feasible state trajectories \( z(t) \) and the neighboring optimal control \( v(t) \) still exist, where \( v(t) \) is determined by

\[
u(t) = u^*(t) + G(t)(Z(t) - \lambda_1 \Delta(t))
\]  

(40)

or

\[
u(t) = u^*(t) + \beta(t)(\Delta(t) X(t))
\]  

(41)

where

\[
Z(t) = \begin{bmatrix} x(t) \end{bmatrix}, \quad X(t) = \begin{bmatrix} x(t) \\ \lambda_1 \end{bmatrix}
\]  

\[
\Delta X(t) = \begin{bmatrix} \beta(t) \end{bmatrix}
\]  

(42)

and \( \Delta X(t) \) is a deviation vector of the new state vector \( Z(t) \) from the open-loop optimal (nominal) state trajectory \( \lambda_1 \). The matrix \( G(t) \) is the feedback gain matrix which is given by

\[
G(t) = G(t) D(t)\Delta X^{-1}(t)
\]  

(43)

where \( D(t) \) are the control errors caused by the perturbations imposed on the state vector \( X(t) \) and \( \delta X(t) \) is a perturbation matrix due to perturbations in the initial states \( x(t_0) \) and initial system parameters \( \lambda_1 \), where \( i = 1, 2, \ldots, n + p \).

**Proof:** According to the Assumptions 1 and 2, the optimal solution of the initial value problem (IVP) (34) does exist and the Part (1) has already been proved when no system parameters are considered [23].

Based on the consequence of Part (1), Part (2) can be proved as follows. For convenience, the independent variable \( t \) in parentheses will be neglected in the subsequent notations. Assume the control \( u \) is of the form

\[
u = g(t, x, \lambda)
\]  

(44)

where the system parameters \( \lambda \) are included. The system (34) then becomes the following unforced system

\[
\dot{x} = f(t, x, \lambda), \quad x(t_0) = x_0
\]  

(45)

where \( x \in C \) and \( \lambda \in P \).

Under the above conditions, the controller must exist satisfying the constraints of (37) and (38).

First, prove the existence of the optimal feasible solution of the IVP (34) with the new control input \( v \) of (40). By the Theorem 3.5 in [24], all the conditions of NFTET are satisfied, therefore, there exists a unique solution of (34) defined on \([t_0, t_f]\).

Now that the solution of the IVP (34) exists and its optimal solution \( (x^*, u^*) \) can be found according to Assumption 2, the perturbations can be imposed on \( x^* \). In the neighborhood of \( x^* \), the new state vector \( x \) can be defined as follows

\[
x = x^* + \delta x
\]  

(46)

where \( \delta x \) is the resulting perturbation vector from the optimal state variables due to perturbations in the initial states \( x_0 \). Recall that the system parameters change from \( \lambda_1 \) to \( \lambda_2 \), i.e.,

\[
\lambda_2 = \lambda_1 + \delta \lambda
\]  

(47)
By Taylor series expansion, the new control is now
\[ v = g(t, z, \lambda_2) = g(t, x^* + \delta x, \lambda_1 + \delta \lambda) \]
\[ = g(t, x^*, \lambda_1) + \frac{\partial g}{\partial x} \delta x + \frac{\partial g}{\partial \lambda} \delta \lambda + \text{H.O.T.} \]
\[ = u^* + \frac{\partial g}{\partial x} (z - x^*) + \frac{\partial g}{\partial \lambda} (\lambda_2 - \lambda_1) + \text{H.O.T.} \]
\[ = u^* + \left[ \frac{\partial g}{\partial x} \frac{\delta x}{\partial \lambda_1} \left( \frac{z}{\lambda_2} - \frac{x^*}{\lambda_1} \right) \right] + \text{H.O.T.} \] (48)
where H.O.T. denotes the higher order terms. If the perturbation is small enough, i.e., \( \delta x \) is small enough or \( x \) is in the feasible neighborhood of \( x^* \), in other words, (39) is satisfied, then the higher order products of \( \delta x \) and \( \delta \lambda \) must be small enough as well. Moreover, \( \delta \lambda \) is basically small and then its higher order products are small enough. Therefore, the H.O.T. in (48) can be neglected at this point. After the linearization, the new controller is of the form of (40), i.e.,
\[ u = u^* + G(Z - X^*) \] (49)
where \( Z \in R^{n+p} \) and \( X^* \in R^{n+p} \) are defined in (42). For the closed-loop system, the perturbations \( \delta x \) and \( \delta \lambda \) are actually the measured deviations. Therefore, (49) or (46) is equivalent to (41) where \( \delta \lambda \) denotes a deviation vector of the closed-loop state vector from the open-loop nominal (optimal) state trajectory and the gain matrix \( G \) is defined by
\[ G = \left[ \frac{\partial g}{\partial x} \frac{\delta x}{\partial \lambda_1} \right] \in R^{n+p} \] (50)

**Note:** If the perturbation is too large to be satisfied for (39), then H.O.T. cannot be neglected and no neighboring feasible trajectories may be found.

Second, find a numerical method to evaluate the gain matrix \( G \) so that no partial derivatives are evaluated. The gain matrix \( G \) is defined by
\[ G = \frac{\partial g}{\partial x} \frac{\delta x}{\partial \lambda_1} \in R^{n+p} \] (51)
where \( \delta x = X_{\text{perturbed}} - X^* \in R^{n+p} \)
\[ \delta u = u_{\text{perturbed}} - u^* \in R^{n+p} \] (52)
Due to linear approximation, \( \delta u \) is obtained by
\[ \delta u = G \delta X \] (53)
Substituting (50) into (53), \( \delta u \) is then determined by
\[ \delta u = G \delta X \] (54)
In order to evaluate \( G \), a \((n+p) \times (n+p)\) square matrix of \( \delta X \) is required. From the known perturbation instances, two matrices are constructed by
\[ \delta X = \left[ \delta X^1, \ldots, \delta X^{n+p} \right] \in R^{n+p,n+p} \] (55)
\[ \delta U = \left[ \delta u^1, \ldots, \delta u^{n+p} \right] \in R^{n+p,n+p} \] (56)
where \( \delta X^i, i = 1, \ldots, n+p \) and \( \delta u^i, i = 1, \ldots, n+p \) are \( \delta \lambda \)-known perturbation instances of \( \delta X \) and \( \delta u \) respectively. Now the matrix expression of (54) is of the form
\[ \delta U = G \delta X \] (57)
When the known perturbation instances are chosen values yet still small enough, the invertibility of \( \delta X \) can be guaranteed at almost every time point. In case it is singular at a time \( t_j \in [t_0, t] \), that point is simply excluded in order that the rest \( \delta X(t_j) \)’s are nonsingular where \( t_j \in [t_0, t] \) and \( t \neq t_j \). From (57), the gain matrix \( G \) can then be evaluated by (43).

### 3) Application of NFTET to the RLV Trajectory Retargeting

According to NFTET, it can be easily shown that the neighboring feasible trajectory for a RLV system exists when applying the NOC method to the RLV system at A&L phase.

In this system, \( V, g, h, \) and \( x \) are the four state variables and \( \alpha \) is the control variable, \( C_L \) and \( C_D \) are two system parameters. If assuming \( C_L \) and \( C_D \) are two special state variables and the variations of \( C_L \) and \( C_D \) between nominal and those failed conditions are viewed as perturbations as well, then the neighboring feasible trajectory does exist if all the perturbations on the six state variables are small enough according to NFTET. Since the optimal (nominal) controller \( \alpha^* \) has already been obtained by Dijkstra’s Algorithm from the off-line feasible trajectory database using MP scheme, the new controller \( \alpha_{\text{perturbed}} \) for the closed-loop system is then obtained by
\[ \alpha_{\text{perturbed}} = \alpha^* + G(Z - X^*) \] (58)
where \( G, Z \) and \( X^* \) have the same meanings as described in NFTET: \( G \) is the feedback gain matrix, \( Z \) is the perturbed state vector including the system parameters, \( X \) is the original state vector including the system parameters, and \( X^* \) is corresponding off-line optimal (nominal) state vector to \( X \).

Now, it is needed to know the gain matrix \( G \) at first. To evaluate \( G \), some perturbations are imposed on \( X^* \). Note the altitude \( h \) is chosen to be the independent variable (instead of time \( t \)) so that the problem becomes a fixed “final time” problem (recall that \( h_{\text{D}} = 0 \)). Since the dimension of \( Z \) is five, five perturbation instances of \( V, g, x, C_L, \) and \( C_D \) are used. Similarly, there are five corresponding instances of \( \alpha \). Let
\[ \delta X := \begin{bmatrix} V^1 - V^* & \ldots & V^5 - V^* \\ \gamma^1 - \gamma^* & \ldots & \gamma^4 - \gamma^* \\ x^1 - x^* & \ldots & x^5 - x^* \\ C_{L1} - C_{L1} & \ldots & C_{L5} - C_{L5} \\ C_{D1} - C_{D1} & \ldots & C_{D5} - C_{D5} \end{bmatrix} \] (59)
\[ \delta \alpha := [\alpha^1 - \alpha^*; \ldots; \alpha^5 - \alpha^*] \] (60)
where superscript \( i \) denotes the \( i \)-th perturbation instance, \( i = 1, \ldots, 5 \).

The feedback gain matrix \( G \) can then be evaluated by
\[ G = \delta \delta X^{-1} \] (61)

Once \( G \) is obtained, the new controller \( \alpha_{\text{perturbed}} \) under a certain failure can then be determined by (58). In (58), the total number of state variables for the RLV system reduces to five since the altitude \( h \) is chosen to be the independent variable, i.e., \( Z = [V, g, x, C_L, C_D]^T \) and \( X^* = [V^*, g^*, x^*, C_{L1}^*, C_{D1}^*]^T \). The subscript “1” of \( C_{L1} \) and \( C_{D1} \) denotes nominal condition and “2” denotes failure condition. Unlike MP scheme, the perturbed control variable \( \alpha_{\text{perturbed}} \) is obtained by the same equation of (58) no matter what stage the vehicle is in – trims or maneuvers. The neighboring feasible trajectories are then determined by (61) and (58).

### IV. ROBUST PERFORMANCE OF NOC

Robust performance is defined as the low sensitivity of system performance with respect to system uncertainty and terminal disturbance [25], [26]. When the perturbations applied to initial states \( X \) (recall that \( X = [x^2, x^3]^T \) where \( x \) are system parameters) are small enough, the new controller can be obtained by (40) or (49). When the perturbations are not too large such as in some scenarios where the failures are not severe, the NOC method looks promising to find a neighboring feasible trajectory and hence recovers the vehicle from those failures.

Since \( C_L \) and \( C_D \) are viewed as two system parameters of \( \alpha \), the system performance on variation of \( C_L \) and \( C_D \) can present the robustness of the RLV system. When \( C_L \) and \( C_D \) change resulting from a stick-effector failure that is not severe, NOC can successfully reshape trajectory in real time and so recover the vehicle from that failure. Therefore, it shows good robust performance of the NOC approach on terminal disturbance.

When there is disturbance on system dynamics, how is the robust performance of NOC if the condition of NFTET is still met? Wind effect to the system is worth discussing as an example [27]. For the vertical wind shear model which is the variation of horizontal wind velocity with altitude, the wind effect can be given by [28]
\[ \Delta V_{\text{W}} = \frac{d V_{\text{W}}}{d h} \] (62)
where \( \frac{d V_{\text{W}}}{d h} \) can be a constant. If the wind effect is included, the gain matrix remains unchanged but the point mass equations of motion have to be modified in order to accommodate the wind effect. In (1-4), \( \Delta V_{\text{W}} \) is added to state variable \( V \). As a result, the other three state variables change as well. The new neighboring feasible trajectories and the new control are hence obtained in the same way as before. Therefore, the wind effect can be eliminated by using NOC. This also demonstrates good robustness of the NOC approach.

More vigorous analysis on robust performance and robustness enhancement is left to future research.
V. RESULTS AND DISCUSSIONS

In order to construct the feasible trajectory database under nominal conditions, some parameters and constraints used in MATLAB simulation are listed in Table I. Given initial and touchdown conditions for a RLV, a feasible trajectory database is built under nominal condition where the lift and drag contributions only depend on angle of attack \( \alpha \). The nominal lift and drag coefficients are given in Table II.

The control effector failures can dramatically change its lift and drag characteristics [17]. When those failures are not severe, trajectory reshaping using NOC can basically recover the vehicle in real time. To observe the results, the following two failure cases are illustrated.

### TABLE I

<table>
<thead>
<tr>
<th>Some Parameters and Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aerodynamic reference area ( S ) (( ft^2 ))</td>
</tr>
<tr>
<td>Vehicle mass ( m ) (slug)</td>
</tr>
<tr>
<td>Gravitational acceleration ( g ) (( ft/s^2 ))</td>
</tr>
<tr>
<td>Density coefficient ( \beta ) (( 1/ft^2 ))</td>
</tr>
<tr>
<td>Air density at sea level ( \rho_0 ) (( slug/ft^3 ))</td>
</tr>
</tbody>
</table>

| Minimum touchdown downrange \( T_{D\min} (ft) \) | 0 |
| Maximum touchdown downrange \( T_{D\max} (ft) \) | 5,000 |
| Minimum touchdown velocity \( V_{T\min} (ft/s) \) | 260 |
| Maximum touchdown velocity \( V_{T\max} (ft/s) \) | 300 |
| Minimum sink rate \( h_{S\min} (ft/s) \) | -10 |
| Maximum sink rate \( h_{S\max} (ft/s) \) | 0 |
| Minimum Mach number \( M_{\min} \) | 0.25 |
| Maximum Mach number \( M_{\max} \) | 0.45 |

(1) A 35\(^{\circ}\) speedbrake failure
(2) The speedbrake fixed at 50\(^{\circ}\) and the body flap at \(-10^{\circ}\).

The coefficients of \( C_L \) and \( C_D \) for these two locked effector failures are listed in Table II.

When a RLV experiences Failure Case 1, it can be trimmed in a certain range of angle of attack. Fig. 2 indicates that the trajectory reshaping recovers the vehicle from such a failure in 2.714 sec. Since the perturbations applied to initial states and variations of system parameters \( (C_L \) and \( C_D \)) are small enough, NFTET applies and a neighboring feasible trajectory is then determined under Failure Case 1. The fact that the vehicle can be recovered from this failure in real time demonstrates good robust performance of the NOC approach. Fig. 3 shows how NOC approach eliminates the wind effect and compares the results between the effects with and without wind under Failure Case 2. For this failure (both with and without wind), a flight envelope with the available pitch control authority can always be found. If wind effect is included, \( \delta_{\text{W}} = 0.1 \text{sec}^{-2} \) and the elapsed time is 4.086 sec. Note that wind effects change the velocity in state equations (1)-(4) but the gain matrix remains unchanged. The results also demonstrate good robust performance of the RLV when using NOC.

### CONCLUSIONS

Under nominal condition for a RLV, a feasible trajectory database is built by using MP scheme. The database is well defined under the nominal condition. Under some failures such as stuck effectors, trajectory retargeting can recover the RLV. When those failures are not severe, NOC approach does help reshape the trajectory and hence recover the vehicle in real time. The neighboring feasible trajectory existence theorem (NFTET) has already been proved in this paper and guarantees that the RLV can be delivered to its landing site safely and reliably under those failures by reshaping the trajectory using the NOC approach. This also demonstrates good robust performance of NOC approach when wind effects are considered.

As a future direction, the NOC approach can be extended to more severe failure scenarios. Since it is not necessary to rebuild the feasible trajectory database, this can be expected to be verified easily but needs to do more work on improvement of the NOC approach. Moreover, more changes in plant dynamics could be included and more analyses on the robust performance for the RLV system (especially the analyses without linear approximation) can be done in the future research. At this point, the wind effect will then be discussed in more details.
ACKNOWLEDGMENTS

The authors would like to express thanks to Dr. Michael A. Bolender and Dr. David B. Doman in the Air Force Research Laboratory at Wright-Patterson Air Force Base for providing relevant aerodynamic models and enlightening ideas on the research. The authors would also like to acknowledge Dr. George Doyle in the Mechanical and Aerospace Engineering at the University of Dayton for his comments on this paper.

REFERENCES