JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 29, No. 2, March–April 2006

Combined Feedback Linearization and Constrained Model Predictive Control for Entry Flight

W. R. van Soest,* Q. P. Chu,† and J. A. Mulder‡
Delft University of Technology, 2629 HS Delft, The Netherlands

The advantages of constrained model predictive control over proportional integral derivative (PID) control applied to a feedback-linearized entry flight-control problem are discussed. The feedback linearization is based on the full rotational equations of motion rather than on a conventional model derived from time-scale separation. Input and state constraints are applied to avoid input saturations, to guarantee a minimum level of tracking performance, and to avoid physical vehicle state constraints violation. A constraint mapping algorithm is developed to map the input and state constraints on the new inputs after feedback linearization. The performance of the constrained model predictive control design is compared with that of two PID control designs. Simulation of a complete entry flight demonstrates the advantages of the constrained model predictive control design, because control surfaces do not saturate, control actions are more smooth and efficient, no gain scheduling is required, and all performance requirements are satisfied.

Nomenclature

C_{ij}	=	moment coefficient differentiated with respect to j
e_i	=	tracking error of i, deg
H_c	=	control horizon
H_p	=	prediction horizon
I	=	inertial matrix, kg · m^2
M	=	Mach number
$\boldsymbol{M}_B$	=	vector of moments in the body frame, N · m
p, q, r	=	angular rates in the body frame, rad/s
$\boldsymbol{Q}$	=	performance index weight matrix
q_{dyn}	=	dynamic pressure, Pa
$\boldsymbol{R}$	=	performance index weight matrix
T_{Mi}	=	thruster moment about body axis i, N · m
T_s	=	sample time, s
$\boldsymbol{u}$	=	original input vector
$\boldsymbol{x}$	=	original state vector
$\boldsymbol{y}$	=	output vector
$\boldsymbol{z}$	=	vector of nonlinear internal dynamics
α, β	=	aerodynamic flow angles, deg
$\boldsymbol{\Delta\nu}$	=	new input increment vector
δ_a	=	aileron deflection, deg
δ_e	=	elevator deflection, deg
δ_{et}	=	elevator deflection for vehicle trimming, deg
δ_r	=	rudder deflection, deg
$\boldsymbol{\eta}$	=	nonlinear internal dynamics state vector
$\boldsymbol{\nu}$	=	new input vector
$\boldsymbol{\xi}$	=	new state vector
σ	=	bank angle

Subscripts

c	=	commanded
t	=	trimmed condition

Superscripts

$\bar{}$	=	vector expanded over the control horizon
$\wedge$	=	predicted

Introduction

THE goal of entry guidance and control (GC) is to define and follow a trajectory that lies within an entry corridor, constrained by acceleration, heating, dynamic pressure, and controllability limits, and leads to a position and energy state allowing a safe approach and landing. Typically, the Mach number during reentry ranges from 28 to 1.5. The guidance tracking algorithm provides steering commands for attitude control, defined in terms of desired aerodynamic flow angles and the bank angle. Because the sideslip angle should be kept close to zero to prevent excessive heat buildup, the primary control variables are the angle of attack and the bank angle. In the design of reentry trajectories, heating constraints during the hypersonic phase are of prime concern, and the resulting trajectories may feature demanding rapid bank reversal maneuvers from -80 to $+80$ deg. This, together with the fact that entry vehicles are characterized by poor aerodynamic maneuverability, makes the entry attitude control system design problem a challenging task.

The conventional approach to the entry flight-control problem is gain scheduling.[1] The entry trajectory is first partitioned into several segments defined by Mach number intervals. For each of these segments, a linearized dynamic model is derived. Next, a corresponding set of linear controllers is defined. Rather than switching from one controller to the next, a single linear controller is designed with scheduled gains defined through interpolation of the gains of the original set of controllers. Although gain scheduling is a well-proven approach, it has disadvantages. The design of the individual controllers is time-consuming and requires intuitive engineering skills based on experience. Moreover, because the resulting controller is designed for only one specific entry trajectory, it lacks flexibility with respect to its applicability in other cases.

These disadvantages have encouraged the reevaluation of nonlinear control system design for entry flights. Steinberg[2] presents a comparison of intelligent, adaptive, and nonlinear control techniques. In Ref. 3, linear model predictive control is applied to a nonlinear aerospace model. A fuzzy logic attitude controller for entry flight control is proposed in Ref. 4. A widely investigated approach is nonlinear dynamic inversion or feedback linearization (FBL) in combination with a linear controller, such as PID[5–10] or μ synthesis.[11,12] In FBL a nonlinear state feedback control law is applied that, in principle, cancels all system nonlinearities. However, this assumes exact knowledge of the equations of motion, including

Received 9 November 2004; revision received 24 January 2005; accepted for publication 26 January 2005. Copyright © 2005 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/06 $10.00 in correspondence with the CCC.

*Graduate Student, Control and Simulation Division, Faculty of Aerospace Engineering.

†Associate Professor, Control and Simulation Division, Faculty of Aerospace Engineering; q.p.chu@lr.tudelft.nl. Member AIAA.

‡Professor and Chairman, Control and Simulation Division, Faculty of Aerospace Engineering; j.a.mulder@lr.tudelft.nl. Member AIAA.

the aerodynamic model. Although admittedly such an assumption can never be fully met in practice, an effort should be made to eliminate or reduce model errors as much as possible. Therefore, simplifications of the system model, such as those resulting from the commonly used time-scale separation of dynamics and kinematics,[6,12] should best be avoided. In Refs. 13–16, continuous nonlinear predictive control is proposed instead of FBL. Although this approach shows a certain degree of robust performance in the presence of parameter uncertainties and a tradeoff between tracking performance and control effort, input and state constraints cannot be taken into account.

In this paper, the combination of FBL and constrained linear model predictive control (LMPC) is proposed. This combination originated in process engineering.[17,18]

Dynamic inversion, the most basic form of FBL, can only be applied to time-scale-separated dynamics and kinematics, such that the two square subsystems can literally be inverted. Each inverted subsystem consists of three first-order decomposed channels, which can be easily controlled with PID controllers. Unfortunately, this separation introduces modeling errors as the system is originally coupled. General FBL, however, is well suited for complex nonsquare nonlinear models,[19] making time-scale separation unnecessary. Using Lie algebra, one nonlinear state feedback control law is derived which cancels the nonlinearities of both the rotational dynamics and kinematics. The inverted system now consists of three second-order decomposed and time invariant channels which can be controlled by either one multivariable controller or three independent PID controllers. The application of FBL leads to a more accurate linearized model than that resulting from dynamic inversion. It has therefore formed the motivations of applying FBL rather than dynamic inversion.

One drawback of FBL is that in many cases it produces unrealistically large control inputs for canceling the system nonlinearity. The combination of FBL and PID control may lead to actuator saturations, which destroy the linearization. The LMPC concept is capable of taking constraints on the input, state, and output into account. Unlike PID control, successful LMPC satisfies the input constraints and prevents saturations of control actuators. In principle, accomplishment of FBL is therefore guaranteed. The second advantage of LMPC is the fact that it performs an online constrained input optimization based on a performance index (PI) in terms of predicted future system behavior, whereas PID control of second-order systems requires subtle tuning. To implement input, state, and output constraints in the LMPC for a feedback-linearized system, original constraints have to be mapped onto the new constraints after the FBL. We will extend the mapping algorithm suggested in Refs. 17 and 18, based on the assumption that the original nonlinear system is decoupled, to the general case of multivariable systems.

The proposed combination is symbiotic; FBL allows application of the simple linear discrete model predictive control (MPC) concept, whereas constrained LMPC guarantees FBL and determines online constrained optimal inputs while satisfying state constraints.

The next sections of this paper will discuss the mathematical model of the entry vehicle, then the design of the FBL, followed by the design of the LMPC controller and the constraint mapping algorithms. The MPC–FBL concept is numerically compared with the PID–FBL concept with time-scale separation of Ref. 6 and a PID–FBL concept without time-scale separation by means of simulations of a complete entry flight.

X-38 Crew Return Vehicle

We consider the entry flight control of NASA/ESA's X-38 unpowered crew return vehicle (CRV) technology demonstrator. A highly detailed nonlinear model of the CRV is available, based on nonlinear aerodynamic tables, the nonlinear six-degrees-of-freedom (DOF) equations of motion, and actuator and sensor models. The CRV model is implemented in GESARED, an advanced atmospheric entry simulation environment.[20]

The primary task of the GC system is to guide the CRV along a predetermined entry reference trajectory corridor. Because we focus on the control part of the GC system, the translational equations of motion are not considered. The entry controller design is based only on the rotational equations of motion. Although separation of translational and rotational equations of motion is well accepted in space, the entry is partly performed in the atmosphere, which makes decoupling of translational and rotational equations of motions not preferable. By defining the rotational equations of motion in terms of the body angular rates p, q, and r and the aerodynamic angles α, β, and σ, the relation between both sets of equations is addressed.

The time derivatives of both the position and the direction of the velocity and Earth's angular velocity are considered to be negligible with respect to the rotational motion, resulting in the following set of equations:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = I^{-1} \left\{ \boldsymbol{M}_B - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left(I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \right\} \tag{1a}$$

$$\dot{\alpha} = -p\cos(\alpha)\tan(\beta) + q - r\sin(\alpha)\tan(\beta) \tag{1b}$$

$$\dot{\beta} = +p\sin(\alpha) - r\cos(\alpha) \tag{1c}$$

$$\dot{\sigma} = -p\cos(\alpha)\cos(\beta) - q\sin(\beta) - r\sin(\alpha)\cos(\beta) + \dot{\alpha}\sin(\beta) \tag{1d}$$

with the inertia tensor I defined as

$$I = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix} \tag{2}$$

due to mass symmetry about the x axis in the body frame. Note that the simplified equations are only used for controller design and the complete nonlinear model will be used for evaluation of the controller. Furthermore, we make the following assumptions. The navigation sensors are assumed to be ideal (i.e., sensor dynamics are not considered). The guidance controller is also assumed to be ideal. Therefore, the commanded aerodynamic angles α_c, β_c, and σ_c, which serve as steering commands for the flight controller, are equal to the desired aerodynamic angles. All translational motion parameters are derived directly from the reference trajectory.

The vector of total external moments, $\boldsymbol{M}_B$, in Eq. (1a) is defined by the actuator assignment. The CRV is controllable through a reaction control system, which consists of a system of thrusters that can generate variable moments based on pulse-width modulations, and through four independently controlled aerodynamic control surfaces (i.e., two flaps and two rudders). The traditional control surface deflections δ_e, δ_a, and δ_r (elevator, ailerons, and rudder) are defined by combination of the actual control surface deflections that produce pitch, roll, and yaw moments.

The availability of the control systems is scheduled with dynamic pressure. When dynamic pressure is low, only the thrusters are available. With increasing dynamic pressure, the aerodynamic surfaces will replace the thrusters. Actuator scheduling results in five different flight phases. The fifth phase is defined due to the fact that the rudders are only available when $M \leq 6$ (Table 1).

The aerodynamic moments are related to the control surface deflections through a nonlinear aerodynamic database presented in the form of lookup tables. A linear aerodynamic model is obtained by linearization of the database along a trimmed nominal trajectory using the central difference method. The result is the following time-varying linear relation between the control surface deflections

Table 1 Actuator assignment and constraints

Phase	T_{Mx}, N · m	T_{My}, N · m	T_{Mz}, N · m	δ_a, deg	δ_e, deg	δ_r, deg
1	±68	±494	±196	——	——	——
2	±22	±172	±330	±25	0/+50	——
3	——	——	±330	±25	0/+50	——
4	——	——	——	±25	0/+50	——
5	——	——	——	±25	0/+50	±10

and the aerodynamic moment coefficients:

$$C_l = C_{l\beta}\beta + C_{lp}p + C_{lr}r + C_{l\delta_a}\delta_a + C_{l\delta_r}\delta_r \tag{3a}$$

$$C_m = C_{m\alpha}(\alpha - \alpha_t) + C_{mq}q + C_{m\delta_e}(\delta_e - \delta_{\text{et}}) \tag{3b}$$

$$C_n = C_{n\beta}\beta + C_{np}p + C_{nr}r + C_{n\delta_a}\delta_a + C_{n\delta_r}\delta_r \tag{3c}$$

Lift and drag resulting from the control surfaces are neglected and we also do not consider force coefficients.[9] During phase 2, both control surfaces and thrusters are available, not including rudders. A bridging function is used to shift control from thrusters to control surfaces with increasing dynamic pressure.[21]

Feedback Linearization

In this section, the FBL theory is summarized. A thorough discussion can be found in Ref. 19. Consider an n-dimensional multivariable nonlinear system with m inputs u_i and m outputs y_i and which is affine in the input $\boldsymbol{u}$

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} \tag{4a}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \tag{4b}$$

where $\boldsymbol{x} \in X \subset \mathbb{R}^n$, $\boldsymbol{u} \in U \subset \mathbb{R}^m$, and $\boldsymbol{y} \in Y \subset \mathbb{R}^m$.

The functions $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}) \ \cdots \ f_n(\boldsymbol{x})]'$ and $\boldsymbol{h}(\boldsymbol{x}) = [h_1(\boldsymbol{x}) \ \cdots \ h_m(\boldsymbol{x})]'$ are assumed to be continuously differentiable on X and the functions $g(x) = [\boldsymbol{g}_1(\boldsymbol{x}) \ \cdots \ \boldsymbol{g}_m(\boldsymbol{x})] \in \mathbb{R}^{n\times m}$ are continuous functions of $\boldsymbol{x}$. The system has a vector of relative degree $[\rho_1, \ldots, \rho_m]'$ and total relative degree $\rho = \rho_1 + \cdots + \rho_m$. Then there exists a state feedback control law $\boldsymbol{u}$ defined as

$$\boldsymbol{u} = \boldsymbol{\varphi}(\boldsymbol{x}) + \boldsymbol{\vartheta}(\boldsymbol{x})\boldsymbol{\nu} \tag{5}$$

which results in closed-loop linear input/output behavior between the new input $\boldsymbol{\nu}$ and the output $\boldsymbol{y}$. The vector $\boldsymbol{\varphi}(\boldsymbol{x})$ and matrix $\boldsymbol{\vartheta}(\boldsymbol{x})$ are defined as

$$\boldsymbol{\varphi}(\boldsymbol{x}) = -\boldsymbol{\Lambda}^{-1}(\boldsymbol{x})\boldsymbol{l}(\boldsymbol{x}) \tag{6a}$$

$$\boldsymbol{\vartheta}(\boldsymbol{x}) = \boldsymbol{\Lambda}^{-1}(\boldsymbol{x}) \tag{6b}$$

where

$$\boldsymbol{\Lambda}(\boldsymbol{x}) = \begin{bmatrix} L_{g_1}L_f^{\rho_1-1}h_1(\boldsymbol{x}) & \cdots & L_{g_m}L_f^{\rho_1-1}h_1(\boldsymbol{x}) \\ \vdots & \ddots & \vdots \\ L_{g_1}L_f^{\rho_m-1}h_m(\boldsymbol{x}) & & L_{g_m}L_f^{\rho_m-1}h_m(\boldsymbol{x}) \end{bmatrix} \tag{7a}$$

$$\boldsymbol{l}(\boldsymbol{x}) = \begin{bmatrix} L_f^{\rho_1}h_1(\boldsymbol{x}) \\ \vdots \\ L_f^{\rho_m}h_m(\boldsymbol{x}) \end{bmatrix} \tag{7b}$$

and where $L_f^{\rho_j}h_j(\boldsymbol{x})$ and $L_{g_i}L_f^{\rho_j-1}h_j(\boldsymbol{x})$ are Lie derivatives of the scalar functions $h_j(\boldsymbol{x})$ with respect to the vectors $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}_i(\boldsymbol{x})$, with $j, i = 1$ to m. If the matrix $\boldsymbol{\Lambda}(\boldsymbol{x})$ is nonsingular, the control law is well defined and a coordinate transformation,

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \end{bmatrix} \tag{8}$$

defined as a local diffeomorphism based on the calculation of the relative degree of each output, yields the closed-loop system

$$\dot{\boldsymbol{\xi}} = A\boldsymbol{\xi} + B\boldsymbol{\nu} \tag{9a}$$

$$\boldsymbol{y} = C\boldsymbol{\xi} \tag{9b}$$

$$\dot{\boldsymbol{\eta}} = z(\boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\nu}) \tag{9c}$$

with state matrices A, B, and C in Brunovsky block canonical form and new input vector $\boldsymbol{\nu}$.[19] The new state vectors $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are of dimension ρ and $n - \rho$, respectively. The vector $\boldsymbol{z}$ contains the nonlinear internal dynamics, which only appears when the total relative degree ρ is smaller than the original state dimension n (partly feedback-linearized system). The zero dynamics, defined as the internal dynamics when the input is chosen such that the output is and remains zero $[\boldsymbol{z}(\boldsymbol{0}, \boldsymbol{\eta}, \boldsymbol{\nu}_{y=0})]$, should be stable to ensure closed-loop stability.[19]

To allow input constraint mapping, the actuator assignment is taken into account in the FBL controller design. Based on Table 1, this results in three FBL controllers: FBL_{12} for phases 1 and 2, however, they differ in control assignment, FBL_{35} for phases 3 and 5, and FBL_4 for phase 4.

The original six-dimensional state vector $\boldsymbol{x}$ is defined as $\boldsymbol{x} = [p \ q \ r \ \alpha \ \beta \ \sigma]'$. The functions $\boldsymbol{f}(\boldsymbol{x})$, $\boldsymbol{g}(\boldsymbol{x})$ and vector $\boldsymbol{u}$ for each controller design are given by

$$\boldsymbol{f}_{12}(\boldsymbol{x}) = \begin{bmatrix} I_1pq + I_2qr \\ I_3(r^2 - p^2) + I_4pr \\ I_5pq - I_1qr \\ -p\cos(\alpha)\tan(\beta) + q - r\sin(\alpha)\tan(\beta) \\ p\sin(\alpha) - r\cos(\alpha) \\ -p\cos(\alpha)\cos(\beta) - q\sin(\beta) - r\sin(\alpha)\cos(\beta) + \dot{\alpha}\sin(\beta) \end{bmatrix} \tag{10a}$$

$$\boldsymbol{f}_{35,4}(\boldsymbol{x}) = \begin{bmatrix} C_{11}\beta + C_{12}p + C_{13}r + C_{14}pq + C_{15}qr \\ C_{20} + C_{21}\alpha + C_{22}q + C_{23}(r^2 - p^2) + C_{24}pr \\ C_{31}\beta + C_{32}p + C_{33}r + C_{34}pq + C_{35}qr \\ -p\cos(\alpha)\tan(\beta) + q - r\sin(\alpha)\tan(\beta) \\ p\sin(\alpha) - r\cos(\alpha) \\ -p\cos(\alpha)\cos(\beta) - q\sin(\beta) - r\sin(\alpha)\cos(\beta) + \dot{\alpha}\sin(\beta) \end{bmatrix} \tag{10b}$$

$$g_{12}(x) = \begin{bmatrix} I_6 & 0 & I_7 \\ 0 & I_8 & 0 \\ I_7 & 0 & I_9 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad g_{35}(x) = \begin{bmatrix} 0 & C_{16} & C_{17} \\ C_{25} & 0 & 0 \\ 0 & C_{36} & C_{37} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$g_4(x) = \begin{bmatrix} 0 & C_{16} \\ C_{25} & 0 \\ 0 & C_{36} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{10c}$$

$$\boldsymbol{u}_{12} = \begin{bmatrix} M_{\text{XB}} \\ M_{\text{YB}} \\ M_{\text{ZB}} \end{bmatrix}, \qquad \boldsymbol{u}_{35} = \begin{bmatrix} \delta_e \\ \delta_a \\ u_3 \end{bmatrix}, \qquad \boldsymbol{u}_4 = \begin{bmatrix} \delta_e \\ \delta_a \end{bmatrix} \tag{10d}$$

with all I_i combinations of inertia parameters and all C_i combinations of inertia parameters and aerodynamic moment derivative coefficients.[21] The actuator assignment for phases 1 and 2 and the assignment of the third input u_3 in $\boldsymbol{u}_{35}$ (i.e., T_{Mz} in phase 3 and δ_r in phase 5) are based on current dynamic pressure and are performed after FBL. In all other cases, actuator assignment is performed before FBL, such that the linearizing control law is a direct relation between the actual actuators and $\boldsymbol{\nu}$. This can be seen in Eq. (10b), where the aerodynamic model is already inserted in $\boldsymbol{f}(\boldsymbol{x})$.

Because the entry flight controller will track a reference trajectory defined in terms of commanded aerodynamic and bank angles, the design of the first two controllers is based on the following output vector:

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} \alpha \\ \beta \\ \sigma \end{bmatrix} \tag{11}$$

The total relative degree of the system configurations for controllers FBL_{12} and FBL_{35} is 6 ($\rho = 6$). Therefore, complete FBL can be achieved ($n = \rho = 6$) and no internal dynamics remain.

Because the number of inputs should equal the number of outputs, a redefined output vector is required for phase 4, when only the elevator and ailerons are available, resulting in only two controllable outputs. The suggested output vector in Ref. 6, that is, $\boldsymbol{h}_4(\boldsymbol{x}) = [\dot{\alpha} \;\; h_2(\beta, \sigma, \dot{\sigma})]'$ where h_2 is a linear combination of its arguments, leads to a three-dimensional linear subsystem and three-dimensional nonlinear internal dynamics, with stable nonlinear zero dynamics.[21] Contrary to Ref. 6, we suggest the following simpler output redefinition:

$$\boldsymbol{h}_4(\boldsymbol{x}) = \begin{bmatrix} \alpha \\ \sigma \end{bmatrix} \tag{12}$$

This output vector and the system configuration for controller FBL_4 result in a total relative degree of 4 ($\rho = 4$). The control law in Eq. (5) results in a partly linearized system: a four-dimensional linear subsystem and two-dimensional nonlinear internal dynamics. The zero dynamics, defined as follows, are now linear and can therefore be easily analyzed for stability:

$$\begin{bmatrix} \dot{r} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} C_1 & C_2 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} r \\ \beta \end{bmatrix} \tag{13}$$

The coefficients C_i are combinations of inertia parameters and aerodynamic moment derivative coefficients.[21]

The feedback-linearizing control law defined by Eq. (5) is derived algebraically with NonLinCon version 2.0,[22] a nonlinear control tool for Maple 6. The exact definition of the vectors $\boldsymbol{\varphi}(\boldsymbol{x})$ and matrices $\boldsymbol{\vartheta}(\boldsymbol{x})$ for each FBL controller can be found in Ref. 21.

The closed-loop systems with controllers FBL_{12} and FBL_{35} are defined as

$$\dot{\boldsymbol{\xi}}^{(6)} = A^{(6\times 6)}\boldsymbol{\xi}^{(6)} + B^{(6\times 3)}\boldsymbol{\nu}^{(3)} \tag{14a}$$

$$\boldsymbol{y}^{(3)} = C^{(3\times 6)}\boldsymbol{\xi}^{(6)} \tag{14b}$$

where the upper indices indicate the dimensions of the vectors and matrices. The new state vector $\boldsymbol{\xi}$ is defined in terms of the original state $\boldsymbol{x}$ as $\boldsymbol{\xi} = [\alpha \;\; \dot{\alpha} \;\; \beta \;\; \dot{\beta} \;\; \sigma \;\; \dot{\sigma}]'$ and A, B, and C are in Brunovsky block canonical form.

The closed loop with controller FBL_4 is defined as

$$\dot{\boldsymbol{\xi}}^{(4)} = A^{(4\times 4)}\boldsymbol{\xi}^{(4)} + B^{(4\times 2)}\boldsymbol{\nu}^{(2)} \tag{15a}$$

$$\boldsymbol{y}^{(2)} = C^{(2\times 4)}\boldsymbol{\xi}^{(4)} \tag{15b}$$

$$\dot{\boldsymbol{\eta}}^{(2)} = \boldsymbol{z}(\boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\nu})^{(2)} \tag{15c}$$

where the stable internal dynamics are defined as $\boldsymbol{\eta} = [r \;\; \beta]$; $\boldsymbol{\xi}$ is defined as $\boldsymbol{\xi} = [\alpha \;\; \dot{\alpha} \;\; \sigma \;\; \dot{\sigma}]$; and A, B, and C are again in Brunovsky block canonical form. The closed loops of Eqs. (14) and (15) represent the basic models for the MPC design in the next section.

Model Predictive Control

Depending on different dimensions of the linearized subsystems, two MPC controllers must be designed: MPC_1 for phases 1, 2, 3, and 5 ($n = 6$) and MPC_2 for phase 4 ($n = 4$). Internal dynamics as present in phase 4 are not observable in the outputs and thus do not play a role in the MPC design. Consequently, the designs of MPC_1 and MPC_2 are identical except for their dimensions and therefore only the design of MPC_1 is discussed here.

First, the linearized closed loop of Eq. (14) is discretized with sample time T_s, which yields

$$\boldsymbol{\xi}(k+1) = A_d\boldsymbol{\xi}(k) + B_d\boldsymbol{\nu}(k) \tag{16a}$$

$$\boldsymbol{y}(k) = C\boldsymbol{\xi}(k) \tag{16b}$$

where system matrices A_d and B_d follow directly from their continuous-time equivalents. The discrete time linear model of Eq. (16) is the prediction model within the MPC concept. Although MPC stability is generally ensured by application of endpoint constraints and infinite prediction horizons, here a finite-prediction-horizon MPC concept without endpoint constraint is applied. While the choice of feasible initial conditions and a sufficiently long but still finite prediction horizon will serve to guarantee feasibility and stability, the problem of dealing with an infinite prediction horizon is avoided.[23]

The PI is defined in terms of tracking errors. To introduce integral action, an extra term that considers the input increments (in combination with a proper disturbance model) is added. The disturbance is considered to be constant; that is, the current mismatch between the actual and predicted output is measured and extended over the prediction horizon. The PI is defined as

$$\boldsymbol{J} = \sum_{i=1}^{H_p} \boldsymbol{e}'(k+i)\boldsymbol{Q}\boldsymbol{e}(k+i) + \sum_{i=1}^{H_c} \Delta\boldsymbol{\nu}'(k+i-1|k)\boldsymbol{R}\Delta\boldsymbol{\nu}(k+i-1|k) \tag{17}$$

where $\boldsymbol{e}(k+i) = \hat{\boldsymbol{y}}(k+i|k) - \boldsymbol{y}_c(k+i)$, $\boldsymbol{Q} \geq 0$, $\boldsymbol{R} > 0$, and k refers to actual time t_k. Beyond the control horizon H_c, the input is assumed to be fixed: $\Delta\boldsymbol{\nu}(k+H_c|k) = \boldsymbol{0}$. The reference output $\boldsymbol{y}_c$ and the input increments $\Delta\boldsymbol{\nu}$ are defined as $\boldsymbol{y}_c = [\alpha_c \;\; \beta_c \;\; \sigma_c]'$ and $\Delta\boldsymbol{\nu} = [\Delta\nu_1 \;\; \Delta\nu_2 \;\; \Delta\nu_3]'$, in which $\Delta\nu_j$ is the jth input increment of the new input of the feedback-linearized system, defined by $\Delta\nu_j(k+i|k) = \nu_j(k+i|k) - \nu_j(k+i-1|k)$.

The future predicted output $\hat{\boldsymbol{y}}(k+i|k)$ is determined with the discrete prediction model of Eq. (16), based on the current state $\boldsymbol{x}(k)$ (assuming that full earlier state measurement is possible) and the actual input of the former time step, $\nu(k+i-1|k)$. The new state $\boldsymbol{\xi}(k)$ is determined with the coordinate transformation $\boldsymbol{\Phi}(\boldsymbol{x})$ of Eq. (8); that is, $\boldsymbol{\xi}(k) = \boldsymbol{\Phi}(\boldsymbol{x}(k))$. The vector of predicted outputs at time k is then determined by

$$\hat{\bar{\boldsymbol{y}}}(k) = \tilde{C}\boldsymbol{\xi}(k) + \tilde{D}_1\boldsymbol{\nu}(k-1) + \tilde{D}_2\Delta\bar{\boldsymbol{\nu}}(k) \tag{18}$$

where

$$\hat{\bar{\boldsymbol{y}}}(k) = [\hat{\boldsymbol{y}}(k+1|k) \quad \hat{\boldsymbol{y}}(k+2|k) \quad \cdots \quad \hat{\boldsymbol{y}}(k+H_p|k)]' \tag{19a}$$

$$\Delta\bar{\boldsymbol{\nu}}(k) = [\Delta\boldsymbol{\nu}(k|k) \quad \Delta\boldsymbol{\nu}(k+1|k) \quad \cdots \quad \Delta\boldsymbol{\nu}(k+H_c-1|k)]' \tag{19b}$$

and $\tilde{C}$, $\tilde{D}_1$, and $\tilde{D}_2$ are properly defined matrices in terms of the system matrices A_d, B_d, and C.[21] Because the continuous-time system matrix A is in Brunovsky block canonical form, the eigenvalues of A_d lie on the unit circle. The system is unstable, but the pair (A_d, B_d) is controllable. Therefore, as long as the elements of the weighing matrix $\boldsymbol{R}$ in Eq. (17) are not infinitely large, a stabilizing MPC controller can be found.

Constraint Mapping

Both input and state constraints are considered. The input constraints are defined as maximum and minimum thruster moments and aerodynamic control surface deflections (Table 1). State constraints are defined in terms of maximum and minimum tracking errors (for all phases, $e_\alpha = \pm 0.5$ deg, $e_\beta = \pm 0.5$ deg, $e_\sigma = \pm 5.0$ deg). The input and state constraints are transformed into constraints on the input increments.

Input Constraint Mapping

Because of the nonlinear FBL controller, the constraints on $\boldsymbol{\nu}$ are nonlinear and state dependent and thus not suitable for optimization through quadratic programming. Therefore, we apply a mapping algorithm that transforms the input constraints on $\boldsymbol{u}$ into input increment constraints on $\Delta\boldsymbol{\nu}$. This transformation must be performed at each time k, because the mapping is state dependent as a result of the state-dependent linearizing control law. Moreover, the transformation must be performed over the entire control horizon to fit into the MPC concept.

In this paper we will extend the input constraint-mapping algorithms for nonlinear single-input/single-output (SISO) and decoupled multivariable nonlinear systems, suggested in Refs. 17 and 18, respectively. In the SISO case, the scalar input u is related to the scalar output y. As a result, input constraints can be mapped exactly to the new scalar input v. This is also the case for decoupled multivariable nonlinear systems. The CRV multivariable nonlinear model is coupled, which requires a different approach than the one presented in Refs. 17 and 18.

Consider the vectors of upper and lower bounds on $\boldsymbol{u}$:

$$\boldsymbol{u}_{\min} \le \boldsymbol{u} \le \boldsymbol{u}_{\max} \tag{20}$$

The feedback-linearizing control law of Eq. (5) can be substituted for $\boldsymbol{u}$ in Eq. (20). This results in the following expression:

$$\boldsymbol{u}_{\min} \le \boldsymbol{\varphi}(\boldsymbol{x}) + \boldsymbol{\vartheta}(\boldsymbol{x})\boldsymbol{\nu} \le \boldsymbol{u}_{\max} \tag{21}$$

which can be redefined as

$$\boldsymbol{u}_{\min} - \boldsymbol{\varphi}(\boldsymbol{x}) \le \boldsymbol{\vartheta}(\boldsymbol{x})\boldsymbol{\nu} \le \boldsymbol{u}_{\max} - \boldsymbol{\varphi}(\boldsymbol{x}) \tag{22}$$

Premultiplication of Eq. (22) by $\boldsymbol{\vartheta}^{-1}(\boldsymbol{x})$ is only allowed when matrix $\boldsymbol{\vartheta}(\boldsymbol{x})$ is nonsingular and positive definite: $\boldsymbol{\vartheta}(\boldsymbol{x}) > 0$. Nonsingularity is always required by definition of the feedback-linearizing control law [see Eq. (6b)]. Generally, however, $\boldsymbol{\vartheta}(\boldsymbol{x})$ is not a priori positive definite, which could alter some of the inequality signs. To maintain generality of the mapping algorithm, the premultiplication is not allowed.

In the case of SISO and decoupled multivariable nonlinear systems, the relations stated in Eq. (21) are m individual expressions, one for each input u_i. To determine the upper and lower bounds on $\boldsymbol{\nu}$, two simple and trivial optimizations are performed.

In the general nonlinear multi-input/multi-output case, however, this is not possible. We continue from Eq. (22). The bounds on $\boldsymbol{u}$ are transformed into inequality constraints on $\boldsymbol{\nu}$:

$$\begin{bmatrix} \boldsymbol{\vartheta}(\boldsymbol{x}(k)) \\ -\boldsymbol{\vartheta}(\boldsymbol{x}(k)) \end{bmatrix} \boldsymbol{\nu}(k) \le \begin{bmatrix} \boldsymbol{u}_{\max}(k) - \boldsymbol{\varphi}(\boldsymbol{x}(k)) \\ \boldsymbol{\varphi}(\boldsymbol{x}(k)) - \boldsymbol{u}_{\min}(k) \end{bmatrix} \tag{23}$$

Then $\boldsymbol{\nu}(k)$ is replaced by $\boldsymbol{\nu}(k-1) + \Delta\boldsymbol{\nu}(k)$, resulting in the inequality constraint on $\Delta\boldsymbol{\nu}(k)$:

$$\begin{bmatrix} \boldsymbol{\vartheta}(\boldsymbol{x}(k)) \\ -\boldsymbol{\vartheta}(\boldsymbol{x}(k)) \end{bmatrix} \Delta\boldsymbol{\nu}(k) \le \left\{ \begin{bmatrix} \boldsymbol{u}_{\max}(k) - \boldsymbol{\varphi}(\boldsymbol{x}(k)) \\ \boldsymbol{\varphi}(\boldsymbol{x}(k)) - \boldsymbol{u}_{\min}(k) \end{bmatrix} - \begin{bmatrix} \boldsymbol{\vartheta}(\boldsymbol{x}(k)) \\ -\boldsymbol{\vartheta}(\boldsymbol{x}(k)) \end{bmatrix} \boldsymbol{\nu}(k-1) \right\} \tag{24}$$

Expanding Eq. (24) over the control horizon results in the inequality constraint

$$\boldsymbol{\Theta}(\bar{\boldsymbol{x}}(k))\Delta\bar{\boldsymbol{\nu}}(k) \le \boldsymbol{\phi}(\bar{\boldsymbol{x}}(k)) \tag{25}$$

in which $\boldsymbol{\Theta}(\bar{\boldsymbol{x}}(k))$ and $\boldsymbol{\phi}(\bar{\boldsymbol{x}}(k))$ are defined as

$$\boldsymbol{\Theta}(\bar{\boldsymbol{x}}(k)) = \begin{bmatrix} \boldsymbol{\Theta}_k & 0 & \cdots & 0 \\ \boldsymbol{\Theta}_{k+1} & \boldsymbol{\Theta}_{k+1} & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ \boldsymbol{\Theta}_{k+H_c-1} & \cdots & \boldsymbol{\Theta}_{k+H_c-1} & \boldsymbol{\Theta}_{k+H_c-1} \end{bmatrix} \tag{26a}$$

$$\boldsymbol{\phi}(\bar{\boldsymbol{x}}(k)) = \begin{bmatrix} \boldsymbol{\phi}_k \\ \boldsymbol{\phi}_{k+1} \\ \vdots \\ \boldsymbol{\phi}_{k+H_c-1} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Theta}_k \\ \boldsymbol{\Theta}_{k+1} \\ \vdots \\ \boldsymbol{\Theta}_{k+H_c-1} \end{bmatrix} \boldsymbol{\nu}(k-1) \tag{26b}$$

and where $\boldsymbol{\Theta}_{k+i}$ and $\boldsymbol{\phi}_{k+i}$ are defined as the left-hand-side matrix and right-hand-side vector of Eq. (24), respectively, for time $k+i$. Given the future state $\boldsymbol{x}(k+i)$, the inequality constraint of Eq. (25) is defined. However, the future state $\boldsymbol{x}(k+i)$ is not known until the LMPC problem is solved, and the LMPC problem is not solved until the constraints are specified. To address this problem, the future constraints are approximated using the inputs calculated at the former time step to approximate the future state $\boldsymbol{x}(k+i)$.[17,18] We present two constraint approximations.

1) Constant constraint mapping: the inequality constraint for time k of Eq. (24) is valid for the entire control horizon; that is, $\hat{\boldsymbol{x}}(k+i) = \boldsymbol{x}(k)$ for $i = 1 : H_c - 1$ and, thus, matrix $\boldsymbol{\Theta}_{k+i} = \boldsymbol{\Theta}_k$ and vector $\boldsymbol{\phi}_{k+i} = \boldsymbol{\phi}_k$ for $i = 1 : H_c - 1$.

2) Varying constraint mapping: the future state is approximated by using the optimal control sequence of the former time step $\bar{\boldsymbol{\nu}}(k-1|k-1)$ and the prediction model of Eq. (16), resulting in a vector of predicted states, $\hat{\bar{\boldsymbol{x}}}(k|k-1)$. Consider the vector of optimal control inputs calculated at time $k-1$: $\bar{\boldsymbol{\nu}}(k-1|k-1)$. To estimate the future states, the first element $\boldsymbol{\nu}(k-1|k-1)$ is removed and an arbitrary input $\boldsymbol{\nu}(k+H_c-1)$ is inserted at the end, resulting in

$$\bar{\boldsymbol{\nu}}(k|k-1) = [\boldsymbol{\nu}(k|k-1) \quad \boldsymbol{\nu}(k+1|k-1) \quad \cdots \quad \boldsymbol{\nu}(k+H_c-2|k-1) \quad \boldsymbol{\nu}(k+H_c-1)]' \tag{27}$$

Together, the input vector of Eq. (27), the transformed measured state $\boldsymbol{\xi}(k)$, and the prediction model of Eq. (16) give the vector of future states in the normal form, defined as

$$\boldsymbol{\xi}(k|k-1) = [\boldsymbol{\xi}(k) \quad \boldsymbol{\xi}(k+1|k-1) \quad \cdots \quad \boldsymbol{\xi}(k+H_c-1)]' \tag{28}$$

The actual approximated future state is determined with the inverse diffeomorphism $\boldsymbol{\Phi}^{-1}(\boldsymbol{x})$, resulting in

$$\hat{\bar{\boldsymbol{x}}}(k|k-1) = [\boldsymbol{x}(k) \quad \boldsymbol{x}(k+1|k-1) \quad \cdots \quad \boldsymbol{x}(k+H_c-1)]' \tag{29}$$

With the approximated state of Eq. (29), $\boldsymbol{\Theta}(\hat{\bar{\boldsymbol{x}}}(k|k-1))$ and $\boldsymbol{\phi}(\hat{\bar{\boldsymbol{x}}}(k|k-1))$ are determined and the inequality constraints of Eq. (25) are specified.

In both cases, the constraints on $\Delta\boldsymbol{\nu}(k)$ depend on the actual measured state. Only the constraints on $\Delta\boldsymbol{\nu}(k+i)$ for $i = 1 : H_c - 1$ are approximations. As result, the implemented inputs $\boldsymbol{\nu}(k)$ are feasible to satisfy the original input constraints at time k. The actual implemented input is not affected by the approximation, except for the fact that the inputs are optimized over the control horizon based on approximated future constraints.

The five entry phases dictate five constraint-mapping algorithms. In phase 1, varying constraint mapping is used. In the other phases, constant constraint mapping is suggested to avoid computationally complex interpolations of the aerodynamic derivatives of Eq. (3) for each $k+i$ during the online constraint mapping. Whenever H_p will cross the phase borders, the accompanying change of constraints is taken into account. In phase 2, the most restrictive of either thruster moments or aerodynamic moments, rated by the aforementioned bridging function, defines the actual constraints.

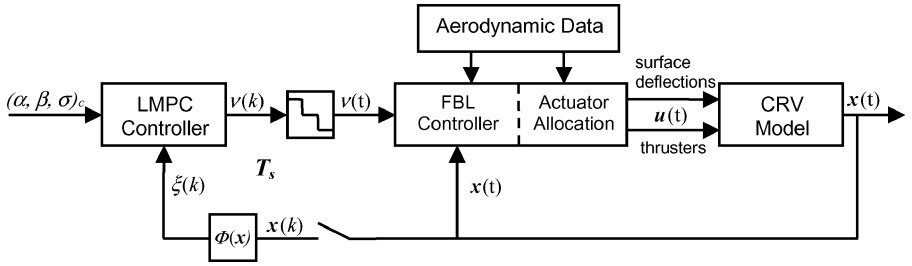

Fig. 1 MPC–FBL control concept.

State Constraints

Because the performance criteria are defined in terms of maximum tracking errors of the states α, β, and σ with respect to the reference trajectories α_c, β_c, and σ_c, we can implement them in the MPC controller in the form of state constraints. The new states of the feedback-linearized system are again the aerodynamic angles, together with their time derivatives. The bounds on the tracking errors can be defined in terms of $\boldsymbol{\xi}$. With the tracking error bound defined as

$$-\boldsymbol{e}_{\max} \le \boldsymbol{\xi}(k+i) - \boldsymbol{\xi}_c(k+i) \le \boldsymbol{e}_{\max} \tag{30}$$

and the state-space model as

$$\boldsymbol{\xi}(k+i) = A_d\boldsymbol{\xi}(k+i-1) + B_d\boldsymbol{\nu}(k+i-2) + B_d\Delta\boldsymbol{\nu}(k+i-1) \tag{31}$$

for each $i = 1 : H_c - 1$, the state constraints are transformed into input increment constraints, resulting in the inequality constraint

$$\boldsymbol{S}\Delta\bar{\boldsymbol{\nu}}(k) \le \boldsymbol{s}(\boldsymbol{x}(k)) \tag{32}$$

where $\boldsymbol{S}$ and $\boldsymbol{s}(\boldsymbol{x}(k))$ are defined in terms of A_d, B_d, $\boldsymbol{e}_{\max}$, the reference trajectory, the measured state $\boldsymbol{x}(k)$ [transformed to $\boldsymbol{\xi}(k)$], and the applied input of the former time step $\boldsymbol{\nu}(k-1)$. The exact definition is trivial and is not be discussed here. Together with Eq. (25), Eq. (32) forms the final inequality constraint used in the optimization of $\Delta\boldsymbol{\nu}(k)$:

$$\begin{bmatrix} \boldsymbol{\Theta}(\hat{\tilde{\boldsymbol{x}}}(k|k-1)) \\ \boldsymbol{S} \end{bmatrix} \Delta\bar{\boldsymbol{\nu}}(k) \le \begin{bmatrix} \phi(\hat{\tilde{\boldsymbol{x}}}(k|k-1)) \\ \boldsymbol{s}(\boldsymbol{x}(k)) \end{bmatrix} \tag{33}$$

The MPC–FBL control design is shown in Fig. 1. The entry vehicle is modeled in GESARED.[20] The full state $\boldsymbol{x}(t)$ is measured and sampled with T_s, resulting in $\boldsymbol{x}(k)$. The MPC controller calculates the optimal inputs $\boldsymbol{\nu}(k)$ based on the reference trajectory in terms α_c, β_c, σ_c, and $\boldsymbol{x}(k)$. The continuous time input $\boldsymbol{\nu}(t)$ is defined by a zero-order hold function on $\boldsymbol{\nu}(k)$. Based on $\boldsymbol{\nu}(t)$ and $\boldsymbol{x}(t)$ and aerodynamic data, the FBL controller calculates the linearizing and reference tracking input $\boldsymbol{u}(t)$, which is the actual input to the CRV.

Benchmark Results

In this section the MPC–FBL control concept is numerically evaluated and compared with PID control. The design is tested by simulation of an atmospheric entry along the White Sands reference trajectory,[24] which is defined in terms of desired aerodynamic angles, as shown in Fig. 2. The total length of the entry simulation is 1504 s. For convenience the time scale of all figures is normalized. The angle of attack is set to a large angle and decreases at the end. The desired sideslip angle is zero during the complete entry. The bank-angle trajectory is characterized by bank reversals. The definitions of the entry phases in normalized time are as follows: phase 1 = [0, 0.10], phase 2 = [0.10, 0.14], phase 3 = [0.14, 0.17], phase 4 = [0.17, 0.93], and phase 5 = [0.93, 1.00]. The simulations are conducted in MATLAB®/Simulink using the GESARED simulator environment. Note again that GESARED is based on the full six-DOF nonlinear equations of motion, the actual nonlinear aerodynamic database, and an accurate atmospheric model.[20] The performance requirements are defined as $e_\alpha = \pm 0.5$ deg, $e_\beta = \pm 0.5$ deg,

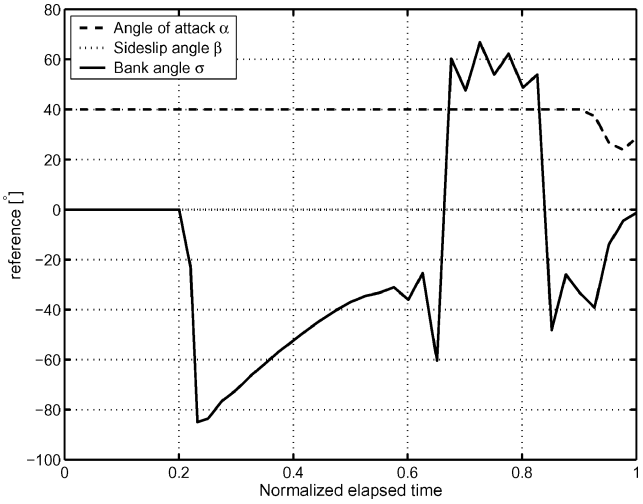

Fig. 2 White Sands reference trajectory.

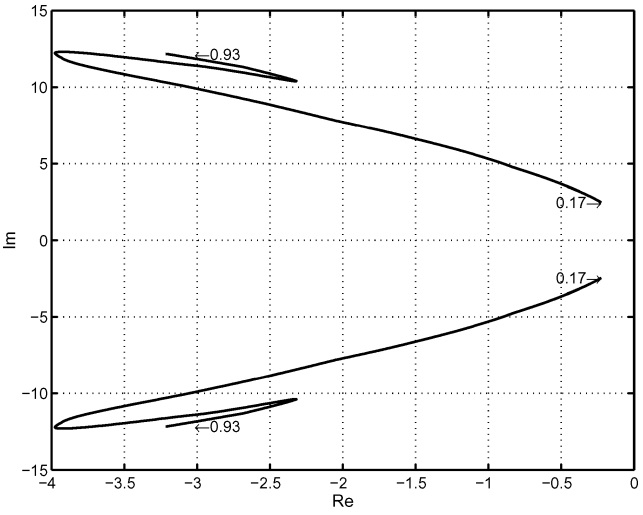

Fig. 3 Eigenvalue time history of the zero dynamics (phase 4).

and $e_\sigma = \pm 5.0$ deg. In Fig. 3, the eigenvalue time history of the zero dynamics of Eq. (13) is presented, showing that the zero dynamics are and remain stable in phase 4 and closed-loop stability is accomplished.

The MPC parameters T_s, H_p, and H_c and the matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are set. Sample time $T_s = 0.1$ s resulted in a satisfying tradeoff between computation speed and tracking performance. The prediction horizon and control horizon are set to 3 and 2 s, respectively ($H_p = 30$, $H_c = 20$). The values for $\boldsymbol{Q}$, $\boldsymbol{R}$, and the disturbance model are defined for each entry phase.[21] To determine whether the input constraints are satisfied, we did not implement saturators for the control surfaces. The deflection rates were limited to 30 deg/s. The thruster saturators are implemented in GESARED.[20]

The tracking errors of the aerodynamic angles are presented in Fig. 4. During the complete entry flight, the tracking errors of all

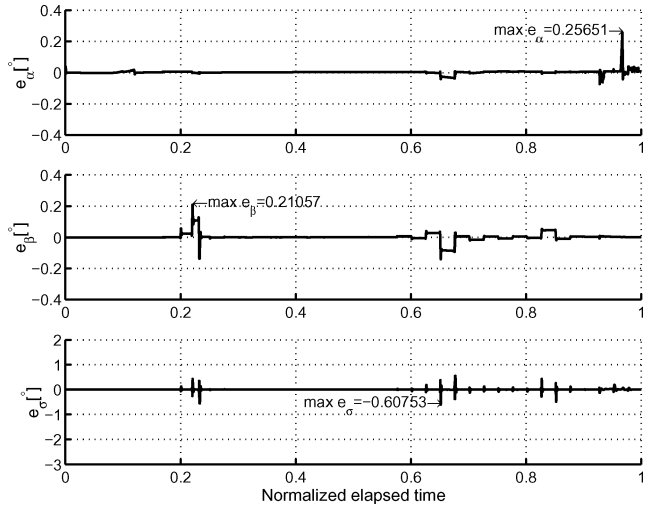

Fig. 4 Aerodynamic angle tracking errors (MPC–FBL).

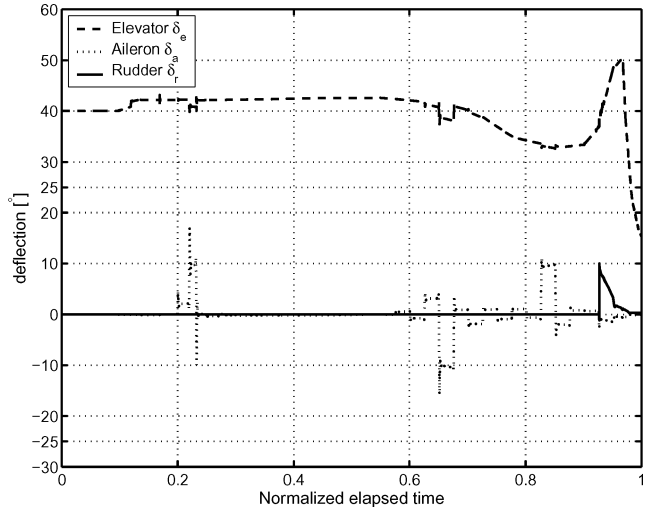

Fig. 5 Aerodynamic surface deflections (MPC–FBL).

aerodynamic angles are small and within the limits of the previously defined performance requirements. The maximum error in angle of attack is +0.26 deg and occurs during the last phase, when both the angle of attack and bank angle change. During the first bank reversal, e_β reaches its maximum value of +0.21 deg. This relatively large error with respect to the rest of the entry is explained by the fact that dynamic pressure is still low at the time of occurrence. The effectiveness of the aerodynamic surfaces is low and the use of rudder is still prohibited. The maximum bank-angle error is only 12% of its maximum allowable value, $e_\sigma = 0.61$ deg. In Fig. 5, the aerodynamic surface deflections are presented. The behavior of all three surfaces is smooth. The ailerons do not reach their maximum allowable deflection, whereas the elevator and rudders do in phase 5. Again it is noted that saturators were not implemented for the control surfaces during simulation. The maximum deflections of both the elevator and rudder were calculated by the MPC–FBL controller and thus FBL was never endangered. The effect of rudder activation in phase 5 can be seen in the almost absent e_β. All thrusters show smooth control performance. As an example, Fig. 6 shows the time history of phases 1, 2, and 3 for T_{Mz}. Overall, the MPC–FBL controller demonstrates the desired performance.

Next, we perform a benchmark comparison of the MPC–FBL control design with two PID–FBL control designs: one without time-scale separation (PID–FBL1) and one with time-scale separation (PID–FBL2). For PID–FBL1, the MPC controller in Fig. 1 is replaced by a continuous-time PID controller. For PID–FBL2, we refer to a design of earlier work presented in Ref. 6. Because the simulation environment, model, and reference trajectory in Ref. 6 are exactly equal to the ones used in this paper, the simulation results of Ref. 6 are duplicated in Table 2. Because PID does not take input constraints into account, saturators were put in place to limit the control surface deflections. Note that all three controllers were not specifically tuned for the White Sands reference trajectory.

Table 2 Benchmark results

Parameter	MPC-FBL	PID-FBL1[a]	PID-FBL2[b]
Max e_α	0.26 deg/1.00	0.30 deg/1.15	1.20 deg/4.61
Max e_β	0.21 deg/1.00	0.35 deg/1.67	0.23 deg/1.10
Max e_σ	0.61 deg/1.00	2.16 deg/3.54	25.0 deg/41.0
Av. norm. error	1.00	2.12	15.6
Aileron saturated	No	Yes	Yes
Elevator saturated	No	Yes	Yes
Rudder saturated	No	No	Yes
Thruster smoothness	Yes	No	Yes

[a]Without time-scale separation. [b]With time-scale separation.[6]

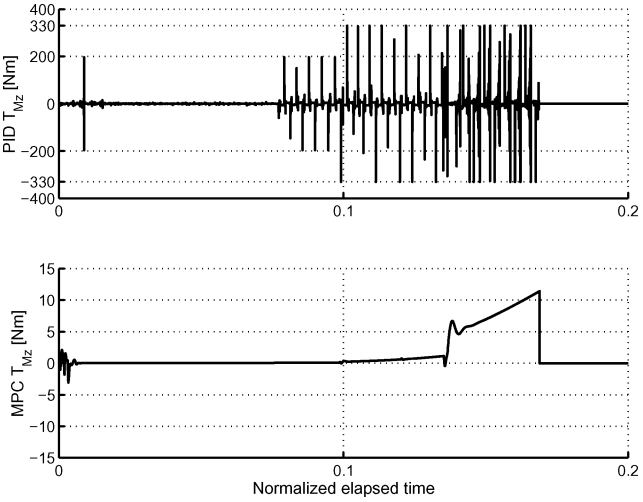

Fig. 6 Thruster moment T_{Mz} for MPC–FBL and PID–FBL1.

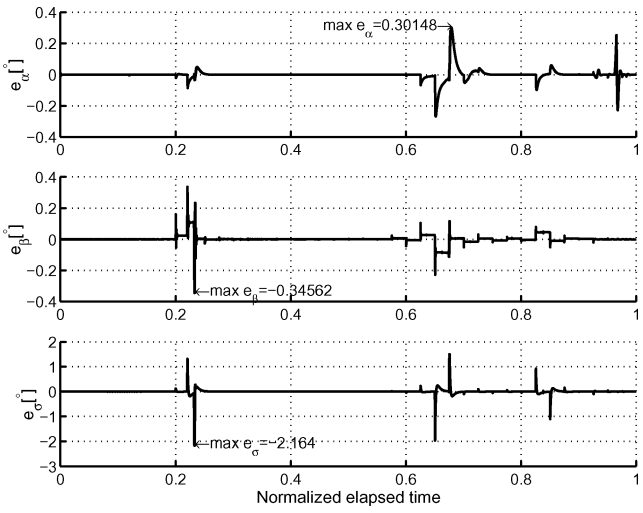

Fig. 7 Aerodynamic angle tracking errors (PID–FBL1).

In Fig. 7, the tracking errors of the aerodynamic angles for PID–FBL1 are presented. Again, all performance requirements are met. The maximum tracking errors are $e_\alpha = 0.30$ deg, $e_\beta = -0.35$ deg, and $e_\sigma = -2.16$ deg, respectively. The maximum value of e_α occurs during the bank maneuver but remains within limits. The maximum bank-angle and sideslip-angle errors occur during the first bank reversal. The less effective aerodynamic surfaces and the absence of the rudder are partly responsible for this, but, as demonstrated in Fig. 8, the saturation of ailerons also plays an important role. As a result of saturation, FBL cannot be accomplished, resulting in larger

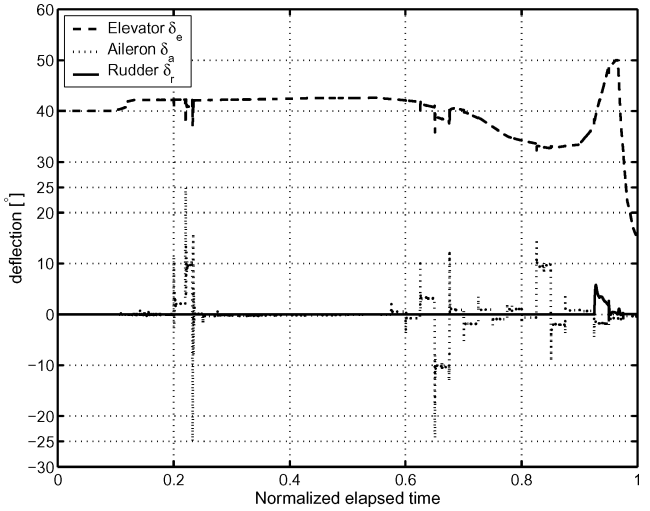

Fig. 8 Aerodynamic surface deflections (PID-FBL1).

tracking errors. In phase 5, the elevator saturates as well. The corresponding angle-of-attack error peaks can be found in Fig. 7. The rudder deflection remains within limits. Although the thruster behavior is not smooth and saturates (Fig. 6), the overall performance of PID–FBL1 is satisfactory.

The benchmark results are presented in Table 2. For convenience, the normalized tracking error with respect to the smallest error is presented next to the actual tracking errors. The average normalized error is the average value of the normalized tracking errors. Based on this number, it can be concluded that the MPC–FBL controller is superior, followed by PID–FBL1 and finally PID–FBL2. Not only the maximum tracking errors, but also the overall MPC–FBL tracking errors are smaller. Comparing the MPC–FBL controller with the two PID–FBL controllers, we notice the great effect of input constraint satisfaction—meaning avoidance of control saturation—that is, preservation of the FBL. Comparing PID–FBL1 and PID–FBL2, we can conclude that time-scale separation deteriorates FBL and results in worse tracking performance. The thruster behavior presented in Fig. 6 confirms MPC's superiority over PID control with respect to control of second-order systems. The online optimization results in smooth and efficient thruster and control surface behavior. Although the basic concept of dynamic inversion, time-scale separation, and PID control of first-order systems as presented in Ref. 6 leads to smooth control and is promising, the required time-scale separation and resulting control saturation degrade the FBL unnecessarily. In conclusion, we can state that the expected advantages of MPC–FBL over PID–FBL are demonstrated.

Conclusions

This paper has demonstrated the potential of combined FBL and constrained MPC in entry flight control. A successful output redefinition is demonstrated during the part of entry when only two control inputs are available. Analysis of the resulting internal dynamics shows that the zero dynamics are and remain stable during entry; that is, closed-loop stability is accomplished. Furthermore, we have developed a successful constraint-mapping algorithm for feedback-linearized multivariable nonlinear systems, which resulted in satisfaction of all constraints.

Entry simulations show that constrained MPC is superior to PID control in combination with FBL, with respect to both tracking performance and control behavior. The main reason for this is the fact that MPC recognizes the input constraints to preserve the FBL. The results show that large tracking errors occur when the PID controllers command excessive inputs. Compared with PID control, the secondary advantages of MPC are the easily accomplished optimal control and the possibility of implementing performance requirements in the controller design by means of state constraints.

As a final remark, we emphasize the questions that remain with regard to robustness properties of the feedback-linearizing control law. A small simplification, such as time-scale separation of the model, can already have a great impact on the performance. An important contribution in this area would be the investigation of FBL combined with robust MPC.

References

[1]Harpold, J. C., and Graves, C. A., "Shuttle Entry Guidance," *Journal of Astronautical Sciences*, Vol. 37, No. 3, 1979, pp. 239–268.

[2]Steinberg, M. L., "Comparison of Intelligent, Adaptive, and Nonlinear Flight Control Laws," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 4, 2001, pp. 693–699.

[3]Shearer, C. M., and Heise, S. A., "Constrained Model Predictive Control of a Nonlinear Aerospace System," *AIAA Guidance, Navigation, and Control Conference*, AIAA, Reston, VA, 1998, pp. 772–785.

[4]Wu, S. F., Engelen, C. J. H., Chu, Q. P., Babuska, R., Mulder, J. A., and Ortega, G., "Fuzzy Logic Based Attitude Control of the Spacecraft X-38 Along a Nominal Re-Entry Trajectory," *Control Engineering Practice*, Vol. 9, No. 7, 2001, pp. 699–707.

[5]Bharadwaj, S., Rao, A. V., and Mease, K. D., "Entry Trajectory Tracking Law via Feedback Linearization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 5, 1998, pp. 726–732.

[6]da Costa, R. R., Chu, Q. P., and Mulder, J. A., "Re-entry Flight Controller Design Using Nonlinear Dynamic Inversion," *Journal of Spacecraft and Rockets*, Vol. 40, No. 1, 2003, pp. 64–70.

[7]Devaud, E., Harcaut, J. P., and Siguerdidjane, H., "Three-Axes Missile Autopilot Design: From Linear to Nonlinear Control Strategies," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 1, 2001, pp. 64–71.

[8]Enns, D., Bugajski, D., Hendrick, R., and Stein, G., "Dynamic Inversion: An Evolving Methodology for Flight Control Design," *International Journal of Control*, Vol. 59, No. 1, 1994, pp. 71–91.

[9]Lane, S. H., and Stengel, R. F., "Flight Control Using Non-linear Inverse Dynamics," *Automatica*, Vol. 24, No. 4, 1988, pp. 471–483.

[10]Snell, S. A., Enns, D. F., and Garrard, W. L., Jr., "Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 976–984.

[11]Bennani, S., and Looye, G., "Flight Control Law Design for a Civil Aircraft Using Robust Dynamic Inversion," *Proceedings of the IEEE/SMC CESA'98*, Vol. 1, IEEE Publications, Piscataway, NJ, 1998, pp. 998–1004.

[12]Reiner, J., Balas, G. J., and Garrard, W. L., "Flight Control Design Using Robust Dynamic Inversion and Time-Scale Separation," *Automatica*, Vol. 32, No. 11, 1996, pp. 1493–1504.

[13]Lu, P., "Nonlinear Predictive Controllers for Continuous Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 3, 1994, pp. 553–560.

[14]Lu, P., "Entry Guidance and Trajectory Control for Reusable Launch Vehicle," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 143–149.

[15]Lu, P., and Hanson, J. M., "Entry Guidance for the X-33 Vehicle," *Journal of Spacecraft and Rockets*, Vol. 35, No. 3, 1998, pp. 342–349.

[16]Singh, S. N., Steinberg, M. L., and DiGirolamo, R. D., "Nonlinear Predictive Control of Feedback Linearizable Systems and Flight Control System Design," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 5, 1995, pp. 1023–1028.

[17]Henson, M. A., and Kurtz, M. J., "Input-Output Linearizing Control of Constrained Nonlinear Processes," *Journal of Process Control*, Vol. 7, No. 1, 1997, pp. 3–17.

[18]Kurtz, M. J., Zhu, G. Y., and Henson, M. A., "Constrained Output Feedback Control of a Multi-Variable Polymerization Reactor," *IEEE Transactions on Control Systems Technology*, Vol. 8, No. 1, 2000, pp. 87–97.

[19]Isidori, A., *Nonlinear Control Systems*, 3rd ed., Springer-Verlag, Berlin, 1995, pp. 207–275.

[20]da Costa, R. R., Silva, J., Wu, S. F., Chu, Q. P., and Mulder, J. A., "Atmospheric Reentry Modeling and Simulation," *Journal of Spacecraft and Rockets*, Vol. 39, No. 4, 2002, pp. 636–639.

[21]van Soest, W. R., "X-38 Nonlinear Flight Controller Design Using Model Predictive Control and On-line Genetic Algorithms," Faculty of Aerospace Engineering, Delft Univ. of Technology, Delft, The Netherlands, TR LR784128, Dec. 2001.

[22]van Essen, H., de Jager, B., Lemmen, M., and Polzer, J., "NonLinCon v2.0," Dynamics and Control Group, Faculty of Mechanical Engineering, Univ. of Eindhoven, The Netherlands, March 2001, URL: http://www.wfw.wtb.tue.nl/control/NonLinCon.html/.

[23]Primbs, J. A., and Nevistić, V., "Feasibility and Stability of Constrained Finite Receding Horizon Control," *Automatica*, Vol. 36, No. 7, 2000, pp. 965–971.

[24]Wiegand, A., Markl, A., Well, K. H., Mehlem, K., Ortega, G., and Steinkopf, M., "ALTOS—ESA's Trajectory Optimization Tool Applied to Re-entry Vehicle Trajectory Design," *Proceedings of International Astronautical Federation Congress*, International Astronautical Federation, Amsterdam, 1999, pp. 1–6.