Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-35 (2022) 37–42

LPV Modeling of the Atmospheric Flight Dynamics of a Generic Parafoil Return Vehicle [⋆]

Matthis H. de Lange [*] Chris Verhoek [*] Valentin Preda [**]
Roland Tóth [*,***]

[*] Control Systems Group, Dept. of Electrical Engineering, Eindhoven University of Technology, Eindhoven 5600MB, The Netherlands.
[**] ESTEC, European Space Agency, Noordwijk 2200AG, The Netherlands.
[***] Systems and Control Lab, Institute for Computer Science and Control, Budapest 1111, Hungary.

Abstract: Obtaining models that can be used for flight control is of outmost importance to ensure reliable guidance and navigation of spacecrafts, like a *Generic Parafoil Return Vehicle* (GPRV). In this paper, we convert an existing, high-fidelity nonlinear model of the atmospheric flight dynamics of a GPRV to a *Linear Parameter-Varying* (LPV) form that enables high-performance navigation control design. Application of existing systematic conversion methods for such complicated nonlinear models often result in complex LPV representations, which are not suitable for controller synthesis. We apply and compare state-of-the-art conversion techniques on the GPRV model, including learning based approaches, to optimize the complexity and conservatism of the resulting LPV embedding. The results show that we can obtain an LPV embedding that approximates the complex nonlinear dynamics sufficiently well, where the balance between complexity, conservatism and model performance is efficiently chosen.

Keywords: linear parameter-varying systems, scheduling reduction, spacecraft modeling, aerodynamics, principle component analysis, deep neural networks.

1. INTRODUCTION

The *European Space Agency* (ESA) is currently developing a *Generic Parafoil Return Vehicle* (GPRV) to perform missions at low orbits. The vehicle is designed to re-enter Earth's atmosphere and land at a designated location on the surface to facilitate multiple reuse. An example of such a vehicle is the *Space Rider* reusable spacecraft. In the final stage of the landing process, the GPRV is navigated towards the landing point by a guided parafoil as shown in Figure 1. The navigation is challenging, as the flight dynamics of the GPRV are subject to changing aerodynamic effects, the parafoil is attached by flexible tension lines to the canopy whose motion is governed by complex fluid dynamics, and the overall vehicle is subject to harsh wind disturbances, while the only available actuation is steering of the parafoil lines (no active propulsion).

Reliable and accurate motion control of a GPRV is essential for proper navigation (heading and flight path tracking) and guaranteeing a safe landing. Hence, the development of an accurate model, useful for flight controller design, is crucial for the *Guidance, Navigation & Control* (GNC) development of the prototype. In this paper, we focus on obtaining a high-fidelity model of a GPRV and show how it can be converted to *Linear Parameter-Varying* (LPV) forms with various complexity levels, enabling high-performance flight controller synthesis.

Fig. 1. Space Rider reusable spacecraft navigating with its parafoil during the final phase of the re-entry and landing process. (Image source: `https://esa.int`.)

Common control design strategies in aerospace applications rely heavily on the LPV framework (Wu et al., 1995; Corti et al., 2012). With this framework, it is possible to embed complex nonlinear systems in a representation with *linear* signal relations. These relations however vary with a signal p, called the *scheduling*, which is assumed to be measurable. This allows to extend powerful methods of the *Linear Time-Invariant* (LTI) framework to design controllers with stability and performance guarantees and rely on efficient performance shaping concepts. While linearity of the resulting LPV surrogate models of the dynamics enables simplified control design and efficient performance shaping, the construction of the often multidimensional

[⋆] This work was supported by the European Space Agency in the scope of the 'AI4GNC' project with SENER Aeroespacial S.A. (contract nr. 4000133595/20/NL/CRS) and was also supported by the European Union within the framework of the National Laboratory for Autonomous Systems (RRF-2.3.1-21-2022-00002). The views expressed in this paper do not reflect the official opinion of the European Space Agency. Corresponding author: Matthis de Lange (`m.h.d.lange@student.tue.nl`)

p that describes the effect of nonlinearities is highly important. In LPV embedding, the constructed scheduling is inherently dependent on internal latent variables, like states and inputs of the system. This dependence is excluded from the LPV model, intentionally seeing p as an external independent variable (Tóth, 2010). By assuming all possible variations of p, the solution set of the LPV model will include the original trajectories of the nonlinear model, but possibly even more due to the disregarded relationship between p and its inducing variables. This is called *conservativeness* of the embedding, and its reduction is highly important to avoid deterioration of the achievable performance of LPV control based on the extracted surrogate model (Tóth, 2010). Furthermore, the dimension of the constructed p and functional dependence of the LPV model coefficients on p (e.g., affine, polynomial, etc. dependence of the matrices in a state-space representation), i.e., *complexity* of the LPV model, have major impact on the computability of model-based LPV controller synthesis (Hoffmann and Werner, 2014). Hence, reduction of such complexity is also a key objective of the LPV modeling toolchain. For this purpose, several conversion strategies, e.g., *substitution based transformation* (SBT) methods (Rugh and Shamma, 2000; Carter and Shamma, 1996; Marcos and Balas, 2004) and *automated conversion procedures* (Kwiatkowski et al., 2006; Hoffmann and Werner, 2015; Tóth, 2010) together with various complexity reduction methods, e.g., (Beck, 2006; Hecker and Varga, 2005), have been introduced and also applied for spacecraft models in Varga et al. (1998). However, only a limited number of methods have been derived to optimize the scheduling complexity in the conversion process, like the family of *Principle Component Analysis* (PCA) methods (Kwiatkowski and Werner, 2008; Rizvi et al., 2016; Sadeghzadeh et al., 2020) and learning-based scheduling reduction methods discussed in Rizvi et al. (2018); Koelewijn and Tóth (2020).

In de Lange (2021), the 12 *Degree of Freedom* (DOF) motion dynamics of a GPRV with detailed aerodynamical effects have been derived in terms of a nonlinear dynamic model. To make this model suitable for LPV control, as a main contribution of the paper, we develop a global embedding of these dynamics in terms of an LPV representation, where both the conservativeness and complexity of the embedding are optimized. For this purpose, we apply and compare two data-based scheduling dimension reduction methods: (i) the PCA method in (Sadeghzadeh et al., 2020) that is the current state-of-the-art method in the PCA family of reduction techniques and the (ii) *Deep Neural Network* (DNN) method from (Koelewijn and Tóth, 2020) that has been reported to perform the best among the learning based methods. The accuracy of the obtained LPV models with various complexity levels is analyzed in simulation studies with the original model.

The paper is organized as follows. First in Section 2, the dynamic motion model of a GPRV is introduced and its LPV conversion based on direct factorization is explained. This is followed in Section 3 by a brief overview of the PCA and DNN methods used for complexity and conservativeness reduction of LPV models. In Section 4, the discussed methods are applied on the LPV modeling problem of the GPRV. Finally, the conclusions on the obtained results are given in Section 5.

2. MODELING OF THE FLIGHT DYNAMICS

In this section, a (simplified) dynamic model of a GPRV is introduced based on the detailed modelling discussed in de Lange (2021).

2.1 Simplified dynamic model of a GPRV

For the simplified model of a GPRV, the parafoil and vehicle body is approximated as a single rigid body. The model equations are given by

$$\dot{r} = R(\eta)v \tag{1a}$$

$$\dot{\eta} = J(\eta)\omega \tag{1b}$$

$$\dot{v} = -\omega \times v + \tfrac{1}{m}\left(f_\mathrm{a}(\eta, v, \omega, \delta, w) + f_\mathrm{g}(r, \eta)\right) \tag{1c}$$

$$\dot{\omega} = -I^{-1}\omega \times I\omega + I^{-1}\left(m_\mathrm{a}(\eta, v, \omega, \delta, w)\right) \tag{1d}$$

with translational position vector $r(t) \in \mathbb{R}^3$, Euler angles $\eta(t) \in \mathbb{R}^3$ of the GPRV and corresponding translational speeds $v(t) \in \mathbb{R}^3$ and angular rates $\omega(t) \in \mathbb{R}^3$ in the frame of the Earth, where $t \in \mathbb{R}$ denotes time. The input $\delta(t) \in [0,1]^2$ is the left and right deflection of the parafoil, while $w(t) \in \mathbb{R}^3$ denotes the wind velocity, which acts as disturbance. $R, J : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ are nonlinear functions of η as given in de Lange (2021). m is the cumulative mass and $I \in \mathbb{R}^{3\times 3}$ is the moment inertia at the center of mass of the spacecraft body, rigidly connected to the parafoil. f_g is the gravitational, while f_a is the aerodynamic force and m_a is the aerodynamic moment, which are all nonlinear functions of the states. The states, forces and moments are all defined in the body frame. We can write (1) as

$$\dot{x} = f(x, u, w), \tag{2a}$$

$$y = x, \tag{2b}$$

where $x(t) \in \mathbb{R}^{n_\mathrm{x}}$ is the composite state variable in terms of $x = [\, r^\top \ \eta^\top \ v^\top \ \omega^\top \,]^\top$ with $n_\mathrm{x} = 12$, while $u = \delta$ and w is the wind disturbance. The whole state is considered to be measurable: $y = x$. A typical operating region of the system is given in terms of the compact sets $\mathbb{X} \subset \mathbb{R}^{n_\mathrm{x}}$, $\mathbb{U} \subset \mathbb{R}^{n_\mathrm{u}}$ and $\mathbb{W} \subset \mathbb{R}^{n_\mathrm{w}}$, corresponding to

$$\begin{aligned} r_x(t), r_y(t) &\in [-3\cdot 10^3, 3\cdot 10^3] \ \text{[m]} \\ r_z(t) &\in [6.3\cdot 10^6, 6.4\cdot 10^6] \ \text{[m]} \\ \eta(t) &\in [-\pi, \pi]^3 \ \text{[rad]} \\ v(t) &\in [-50, 50]^3 \ \text{[m/s]} \\ \omega(t) &\in [-0.1, 0.1]^3 \ \text{[rad/s]}. \end{aligned}$$

Furthermore, $\mathbb{U} := [0,1]^2$ and the wind disturbance in $\mathbb{W}$ is considered to be bounded by 3D gusts of 18 [m/s], which is a realistic choice based on the data of the National Oceanic and Atmospheric Administration.

2.2 LPV conversion by factorization

A common technique to embed a nonlinear SS model (2) into an LPV description is to factorize the state transition function f to obtain

$$\dot{x} = \mathcal{A}(x, u, w)x + \mathcal{B}_\mathrm{u}(x, u, w)u + \mathcal{B}_\mathrm{w}(x, u, w)w, \tag{3}$$

where the matrix functions $\mathcal{A}, \mathcal{B}_\mathrm{u}, \mathcal{B}_\mathrm{w}$ are assumed to be bounded and to have appropriate argument and image dimensions. Note that the output function $y = x$ is already linear, hence it is left out out from the model conversion. To obtain (3) based on (2a), the aerodynamic forces and moments are factorized w.r.t. the trans. and angular velocities and the input. The gravity force is factorized w.r.t. the position vector. As a second step, the scheduling is extracted by constructing $p(t) = \psi(x(t), u(t), w(t)) \in \mathbb{R}^{n_\mathrm{p}}$ such that the resulting LPV model is

$$\dot{x} = A(p)x + B_\mathrm{u}(p)u + B_\mathrm{w}(p)w, \tag{4a}$$
$$y = x, \tag{4b}$$

where $A : \mathbb{R}^{n_\mathrm{p}} \to \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{x}}$, $B_\mathrm{u} : \mathbb{R}^{n_\mathrm{p}} \to \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{u}}$ and $B_\mathrm{w} : \mathbb{R}^{n_\mathrm{p}} \to \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{w}}$ belong to a given function class like affine (that is, $A(p) = A_0 + \sum_{i=1}^{n_\mathrm{p}} A_i p_i$), polynomial, etc., and $\mathcal{A} = A \circ \psi$, $\mathcal{B}_\mathrm{u} = B_\mathrm{u} \circ \psi$, $\mathcal{B}_\mathrm{w} = B_\mathrm{w} \circ \psi$, with $\circ$ denoting the function composition operator. As affine dependence of A, B_u and B_w is generally preferred for controller design, $\mathcal{A}$, $\mathcal{B}_\mathrm{u}$ and $\mathcal{B}_\mathrm{w}$ is converted to $A \circ \psi$, $B_\mathrm{u} \circ \psi$ and $B_\mathrm{w} \circ \psi$ by extracting every nonlinearity as a new scheduling variable. Finally, the scheduling region $\mathbb{P}$ is computed, by taking the smallest hypercube around the extreme values of each component of ψ over $(\mathbb{X}, \mathbb{U}, \mathbb{W})$.

While this constitutes to a simple LPV model conversion process, where the obtained model is an exact representation of the original nonlinear system, the conservativeness and complexity of the representation are maximized, achieving a scheduling dimension $n_\mathrm{p} = 71$. As a next step, we will reduce n_p and optimize the conservativeness of the LPV representation of the GPRV dynamics. Furthermore, we will show that the wind w can be excluded from ψ without significant deterioration of the model accuracy.

3. SCHEDULING REDUCTION METHODS

For reducing the conservativeness and complexity of the converted LPV model, we will first briefly introduce the PCA method of Sadeghzadeh et al. (2020) and the DNN approach in Koelewijn and Tóth (2020), after which they are applied to our GPRV model Section 4.

3.1 PCA-based scheduling dimension reduction

The PCA-based scheduling dimension reduction method of Sadeghzadeh et al. (2020) is an improved version of Kwiatkowski and Werner (2008). The idea of the PCA method is to extract the *principle components* of the model variations that contribute most to the system behavior under typical operation. The principle components are extracted with *Singular Value Decomposition* (SVD), allowing for the selection of an effective number of components with which the reduced model is scheduled.

We capture system variations along the typical operation trajectories of the GPRV in a data-set $\mathcal{D}_N = \{x(k), u(k), w(k)\}_{k=1}^N \in \mathbb{R}^{(n_\mathrm{x}+n_\mathrm{u}+n_\mathrm{w}) \times N}$, on which the PCA is performed. The variation of $A, B_\mathrm{u}, B_\mathrm{w}$ along $p(k)$ (in $\mathcal{D}_N$) is represented with Γ,

$$\Gamma(p(k)) = \mathrm{vec}\left([A(p)\ B_\mathrm{u}(p)\ B_\mathrm{w}(p)](k)\right), \tag{5}$$

where $\mathrm{vec}(\cdot)$ denotes column vectorization of a matrix and $p(k) \in \mathbb{P}_N = \psi(\mathcal{D}_N)$. Note that $\mathcal{D}_N$ should represent the solution space the GPRV encounters during typical operation as much as possible. The variation of Γ over the entries in $\mathcal{D}_N$ is collected in Π_N, i.e.,

$$\Pi_N = [\Gamma(p(1))\ \Gamma(p(2))\ \cdots\ \Gamma(p(N))] \in \mathbb{R}^{n_\Pi \times N}. \tag{6}$$

where $n_\Pi = (n_\mathrm{x} + n_\mathrm{u} + n_\mathrm{w})n_\mathrm{x}$.

To improve numerical conditioning, the data is centered and normalized

$$\bar{\Pi}_N = \mathcal{N}(\Pi_N) := S_\mathrm{scale} \cdot (\Pi_N - \Pi_\mathrm{c} \otimes 1_{n_\Pi \times N}), \tag{7}$$

where $\Pi_\mathrm{c} \in \mathbb{R}^{n_\Pi}$ is the column average, i.e, mean, of Π_N, $S_\mathrm{scale} \in \mathbb{R}^{n_\Pi \times n_\Pi}$ is a diagonal scaling matrix and $\otimes$ denotes the Kronecker product. In this work, min-max normalization and standard deviation-based normalization are considered. For min-max normalization, we have $S_\mathrm{scale} := \mathrm{diag}^{-1}(d(\Pi_{N,1}), \ldots, d(\Pi_{N,n_\Pi}))$, with $d(\Pi_{N,i}) = \max(\Pi_{N,i}) - \min(\Pi_{N,i})$. For standard deviation-based normalization, we have $S_\mathrm{scale} := \mathrm{diag}^{-1}(\mathrm{std}(\Pi_{N,1}), \ldots, \mathrm{std}(\Pi_{N,n_\Pi}))$, with $\mathrm{std}(\cdot)$ the square root of the sample variance.

Taking the SVD of $\bar{\Pi}_N$ allows us to find the principle components in the variations of $A, \ldots, D_\mathrm{w}$, i.e.,

$$\bar{\Pi}_N = U\Sigma V^\top = [U_\mathrm{s}\ U_\mathrm{r}] \begin{bmatrix} \Sigma_\mathrm{s} & 0 \\ 0 & \Sigma_\mathrm{r} \end{bmatrix} \begin{bmatrix} V_\mathrm{s}^\top \\ V_\mathrm{r}^\top \end{bmatrix}, \tag{8}$$

where Σ has the principle components, i.e., singular values, in descending order in its diagonal. $\bar{\Pi}_N$ is projected to a lower dimension, while the n_s most significant components contributing to the variation are retained:

$$\hat{\bar{\Pi}}_N = U_\mathrm{s}\Sigma_\mathrm{s}V_\mathrm{s}^\top = U_\mathrm{s}U_\mathrm{s}^\top\bar{\Pi}_N. \tag{9}$$

The core idea is to use $U_\mathrm{s}^\top\bar{\Pi}_N$ as the new scheduling map, whose dimension is equal to the selected principal components, i.e., $n_\mathrm{s} = n_{\hat{\mathrm{p}}}$. Here U_s describes the linear combination of these variables which describe the variation, i.e., how these new scheduling variables will compose a new affine dependency of the LPV model. Note that such a decomposition is based on normalized and centered variations, hence the approximation of the original matrix variations is found with the inverse of the normalization $\mathcal{N}^{-1}(\hat{\bar{\Pi}}_N) = S_\mathrm{scale}^{-1}\hat{\bar{\Pi}}_N + \Pi_\mathrm{c} \otimes 1_{n_\Pi \times N}$, which is still an affine operator, preserving the affine dependency structure of the LPV model. Based on these, the reduced scheduling variable $\hat{p}$ is given as

$$\hat{p}(t) = U_\mathrm{s}^\top\mathcal{N}(\Gamma(p(t))) = \underbrace{U_\mathrm{s}^\top\mathcal{N}(\Gamma(\psi(x(t), u(t), w(t))))}_{\hat{\psi}(x(t),u(t),w(t))}. \tag{10}$$

The new linear affine dependency on $\hat{p}$ that approximates the original $\mathcal{A}, \ldots, \mathcal{D}_\mathrm{w}$ is reconstructed from the approximated matrix variations $\Gamma(p) \approx \hat{\Gamma}(\hat{p}) = \mathcal{N}^{-1}(U_\mathrm{s}\hat{p})$, i.e.,

$$\hat{\Gamma}(\hat{p}) = \mathrm{vec}\Big(\underbrace{[\hat{A}(\hat{p})\ \hat{B}_\mathrm{u}(\hat{p})\ \hat{B}_\mathrm{w}(\hat{p})]}_{\hat{M}_0 + \sum_{i=1}^{n_{\hat{\mathrm{p}}}} \hat{p}_i\hat{M}_i}\Big), \quad n_{\hat{\mathrm{p}}} < n_\mathrm{p} \tag{11}$$

where $\hat{M}_0 = [\,A_0\ B_{\mathrm{u},0}\ B_{\mathrm{w},0}\,] + \mathrm{vec}^{-1}(\Pi_\mathrm{c})$ and $\hat{M}_i = \mathrm{vec}^{-1}(S_\mathrm{scale}^{-1}U_{\mathrm{s},i})$, with vec^{-1} the inverse operation of (5) and $U_{\mathrm{s},i}$ representing the i^th column of U_s with $1 \le i \le n_\mathrm{s}$.

The final step is to determine the reduced scheduling region $\hat{\mathbb{P}}$ in which $\hat{p}$ is varying. The region $\hat{\mathbb{P}}$ can be defined as a hypercube denoted as

$$\hat{p}_i^{\min} \le \hat{p}_i \le \hat{p}_i^{\max},$$

where $\hat{p}_i^{\min}$ and $\hat{p}_i^{\max}$ are obtained as the minimum and maximum values of $\hat{p}_i(t)$ over $\psi(\mathcal{D}_N)$. This does not result in a $\hat{\mathbb{P}}$ with minimum volume, which introduces conservatism. Sadeghzadeh et al. (2020) discusses methods to find a minimum-volume hypercube that defines $\hat{\mathbb{P}}$ using the Kabsch algorithm (when $n_{\hat{\mathrm{p}}} \le 3$) and hyper-ellipse fitting (when $n_{\hat{\mathrm{p}}} > 3$).

3.2 DNN-based scheduling dimension reduction

The PCA method uses a linear mapping from a fixed set of model variations governed by p to the reduced scheduling vector $\hat{p}$. With the DNN method, proposed in Koelewijn and Tóth (2020), the scheduling map ψ is learned, i.e., optimized, simultaneously along the reduction step. Figure 2 shows a schematic overview of the DNN architecture. The

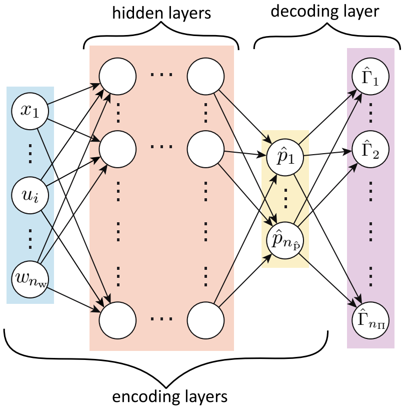


Fig. 2. DNN architecture for learning-based scheduling reduction (adopted from Koelewijn and Tóth (2020)).

encoding layers encode (x, u, w) into the reduced scheduling vector, while the decoding layer decodes the reduced scheduling vector to the approximated model variations, resulting in the linear affine dependency structure. The DNN consists of an input layer, n_{hl} hidden layers and an output layer. The input and hidden layers are given as

$$l^{[\tau]} = g^{[\tau]}\big(W^{[\tau]} l^{[\tau-1]} + b^{[\tau]}\big), \quad \tau = 0, \ldots, n_{\mathrm{hl}}, \tag{12}$$

where $g^{[\tau]}(\cdot)$ is the activation function (e.g., hyperbolic tangent, ReLU, sigmoid), $l^{[\tau]}$ is the output and $W^{[\tau]}$ and $b^{[\tau]}$ are the weight matrix and the bias vector of the τ^{th} layer, respectively. The input to the DNN, i.e., (x, u), is thus $l^{[-1]} := \mathrm{vec}(x, u)$. The reduced scheduling vector is the output of the $n_{\mathrm{hl}}{}^{\mathrm{th}}$ hidden layer, i.e., $\hat{p} := l^{[n_{\mathrm{hl}}]}$. The associated vectorized system matrices $\hat{\Gamma}$ follow from the last layer, which is affine, i.e., $\hat{\Gamma} = W^{[\Gamma]}\hat{p} + b^{[\Gamma]}$. $\hat{A}, \ldots, \hat{B}_{\mathrm{w}}$ are obtained as in the PCA method. The weightings and biases of the DNN are optimized by minimizing

$$\min_{W^{[k]}, b^{[k]}} \tfrac{1}{N}\textstyle\sum_{j=1}^{N} \|\hat{\Gamma}(\hat{p}(j)) - \Gamma(p(j))\|_2^2. \tag{13}$$

The optimization problem is solved with back-propagation combined with stochastic gradient descent, which are implemented in popular solvers such as Adam, or AdaBound (Kingma and Ba, 2014; Luo et al., 2019). Multiple techniques exist to prevent overfitting, like weight regularization and early stopping (Goodfellow et al., 2016). Based on the obtained $\hat{\psi}, \hat{p}$, the scheduling region $\hat{\mathbb{P}}$ can be determined using the same methods as discussed in Section 3.1.

4. LPV MODELING OF THE GPRV

Armed with the PCA and DNN methods introduced in Section 3, we optimize the scheduling complexity in the LPV modeling of the GPRV flight dynamics together with the conservativeness of the embedding.

4.1 Data-generation

We perform the optimization of the complexity and conservativeness based on trajectory data from typical operation of the GPRV. SENER Aerospace presented in Cacciatore et al. (2019) a baseline solution of the GNC problem on a simplified model of the GPRV. From the associated simulator, typical initial conditions are obtained for the states and input trajectories of δ to simulate our high-fidelity model (as presented in Section 2.1) in open-loop. These trajectories navigate the GPRV from ∼5.5 km above Earth's surface to a predefined landing location. As the wind cannot be measured during operation, we are not able to schedule the model based on the wind. For this reason, we exclude the wind from the model i.e., $w = 0$, and we assume that the GNC will be able to reject the disturbance in closed-loop. The simulation is computed with an ODE4 solver running at $f_{\mathrm{s}} = 400$ Hz. The resulting $\mathcal{D}_N$ data-set with $N = 10^5$ is chosen as randomly picked samples of the simulated trajectory.

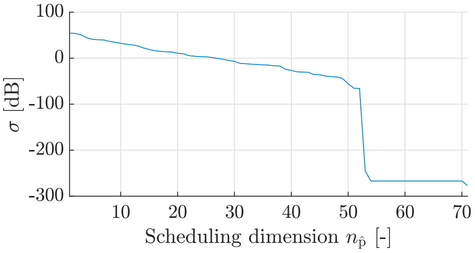


Fig. 3. The singular values of $\bar{\Pi}_N$ in the PCA when $\bar{\Pi}_N$ has been normalized with min-max normalization.

4.2 Reducing the scheduling dimension

We will now further optimize the direct LPV model (4) in both scheduling complexity and conservativeness. The former will be accomplished with the PCA and DNN-based scheduling reduction methods.

4.2.1 PCA approach: Based on (4) and the design choice of $w = 0$, implying that $\mathcal{B}_{\mathrm{w}} \equiv 0$, $\Gamma(p(k))$ reduces to

$$\Gamma(p(k)) = \mathrm{vec}\left([A(p) \;\; B_{\mathrm{u}}(p)](k)\right), \tag{14}$$

from which Π_N is constructed. Normalization of Π_N is done with both std and min-max normalization The principle components of $\bar{\Pi}_N$ with min-max normalization are plotted in Figure 3, which shows an exponential decrease of the singular values. For the 52nd singular value till the last, the singular values drop below the numerical precision bound and can be considered zero. Hence, the full LPV embedding (4) requires 52 principal components to describe *all* variations of $\mathcal{A}$ and $\mathcal{B}_{\mathrm{u}}$ for typical operation. We now construct a compact scheduling map with $n_{\hat{\mathrm{p}}} \ll 52$ to economically represent the GPRV in an LPV form, which results in making a trade-off between model complexity and accuracy. To visualize this trade-off, we compute the reduced model for scheduling dimensions $n_{\hat{\mathrm{p}}} = 1, \ldots, 10$. This is often the range that is numerically manageable in controller synthesis for systems with $n_{\mathrm{x}} > 10$.

4.2.2 DNN approach: The factorization based LPV model in terms scheduling complexity is optimized for $n_{\hat{\mathrm{p}}} = 1, \ldots, 10$ using the DNN approach. The DNN is implemented with 4 hidden layers, each consisting of 128 neurons with tanh activation, which is sufficiently complex to capture the GPRV variations. Moreover, we apply a linear bypass between the input and output to allow for linear input-output relationships. The DNN input is $\left[\, r^\top \; \tilde{\eta}^\top \; V^\top \; \omega^\top \; \delta^\top \,\right]^\top$, where $\tilde{\eta}^\top := \left[\, \sin(\eta)^\top \; \cos(\eta)^\top \,\right]^\top$, resulting in 17 inputs. This decomposition of the Euler angles often helps in training. The decoding layer, i.e., the output layer of the DNN (as depicted in Figure 2), has $n_{\hat{\mathrm{p}}} = 1, \ldots, 10$ inputs[1] and the $n_\Pi = (n_{\mathrm{x}} + n_{\mathrm{u}})n_{\mathrm{x}} = 168$ matrix variations as output. Note that this is a linear layer, which results in a linear affine dependence on $\hat{p}$.

[1] Note that the network has to be retrained for every $n_{\hat{\mathrm{p}}}$.

The ℓ_2-weight regularization is set to 10^{-4}. The Adam optimizer (Kingma and Ba, 2014) is used during training with a learning rate of 10^{-5}. The batch-size is 128 and the network is trained to minimize (13) for 200 epochs. We did not perform any hyper-parameter optimization, as the initial parameter-set already gave good results. Both the input and output data are normalized before training with the aforementioned normalization methods. We want to stress here that the DNN-based scheduling reduction approach simultaneously learns a nonlinear map between the input (x, u) and a scheduling vector of size $n_{\hat{\text{p}}}$ under affine dependency.

4.2.3 Comparison of the results: We compare the outcomes of the scheduling reduction methods using two types of error measures on a validation data-set independent of the training data-set. The first error measure is the normalized approximation error of the elements of Γ, i.e., along the rows of Π. Let

$$e_{\Pi,i} := \frac{\|\Pi_{N,i} - \hat{\Pi}_{N,i}\|_2}{\|\Pi_{N,i}\|_\infty}, \quad i = 1, \dots, n_\Pi, \tag{15}$$

with $\|\cdot\|_2$ and $\|\cdot\|_\infty$ being the 2 and ∞ vector-norms. While this error measure indicates how $\left[\hat{A}(\hat{p})\ \hat{B}_\text{u}(\hat{p})\right]$ captures $[\mathcal{A}(x, u)\ \mathcal{B}_\text{u}(x, u)]$, we are mainly interested in how well the obtained LPV model represents the *true* solution space of the GPRV. An error measure for this is based on comparing the state-derivatives, i.e. $\mathcal{A}(x, u)x + \mathcal{B}_\text{u}(x, u)u$ with $\hat{A}(\hat{p})x + \hat{B}_\text{u}(\hat{p})u$. Let

$$f^N = \left[\, A(p(1))x(1) + B_\text{u}(p(1))u(1) \ \cdots \ A(p(N))x(N) + B_\text{u}(p(N))u(N) \,\right],$$

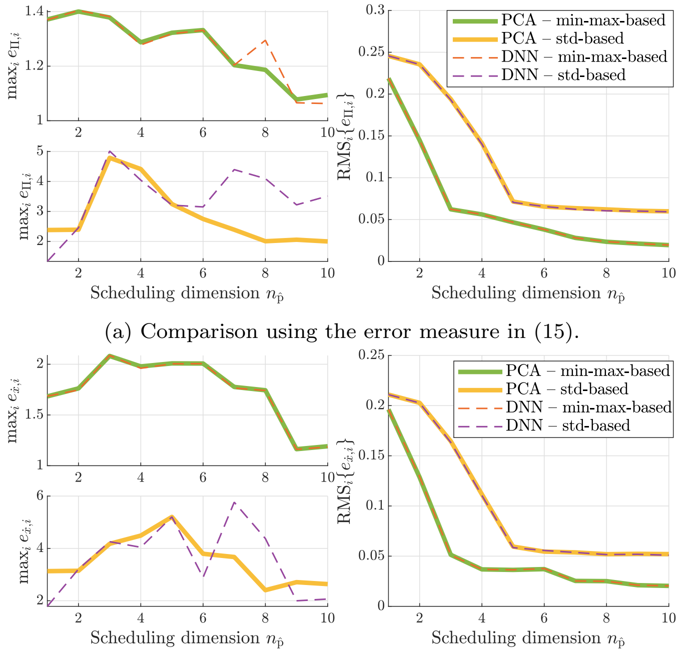
$$\hat{f}^N = \left[\, \hat{A}(\hat{p}(1))x(1) + \hat{B}_\text{u}(\hat{p}(1))u(1) \ \cdots \ \hat{A}(\hat{p}(N))x(N) + \hat{B}_\text{u}(\hat{p}(N))u(N) \,\right],$$

and define the second error measure as

$$e_{\dot{x},i} := \frac{\|f_i^N - \hat{f}_i^N\|_2}{\|f_i^N\|_\infty}, \quad i = 1, \dots, n_\text{x}. \tag{16}$$

Figure 4 shows the plots for $\max_i e_{\Pi,i}$, $\text{RMS}_i\{e_{\Pi,i}\}$, $\max_i e_{\dot{x},i}$ and $\text{RMS}_i\{e_{\dot{x},i}\}$, where $\text{RMS}_i\{a_i\} := \sqrt{\sum_{i=1}^{n_\text{a}} a_i^2}$, for both the PCA and DNN-based scheduling reduction methods. The plots show no significant difference between the PCA and the DNN methods with both error measures, where the main difference between the results is due to the type of normalization that is used on the data. This cause often plays a role in function approximation as well, when using basis functions such as splines or Chebyshev bases. The results show that, with min-max normalization, the RMS error drops below $\sim 5\%$ for $n_{\hat{\text{p}}} \geq 3$, which is excellent model accuracy for such a reduction. Moreover, this $n_{\hat{\text{p}}}$ is highly suitable for controller synthesis. Therefore, $n_{\hat{\text{p}}} = 3$ is the optimal trade-off between complexity and accuracy of the LPV model. When we compare the behavior along a nominal flight trajectory, we obtain the open-loop simulated model responses in Figure 5. The large differences between the trajectories are due to the integrator behavior, i.e., the small errors (quantified in Fig. 4) are magnified over time. This will not play a role during closed-loop operation, as the *qualitative* responses are quite similar.

From a computational perspective, the DNN approach is much more expensive than the PCA approach due to the training time. Especially when the reduction must be computed for multiple $n_{\hat{\text{p}}}$. On the other hand, large data-sets make the SVD computation for PCA intractable, while the DNN approach can handle large data-sets better.



(a) Comparison using the error measure in (15).

(b) Comparison using the error measure in (16).

Fig. 4. Comparison of the results on scheduling reduction: DNN (dashed lines) and PCA (solid line) under min-max-based and std-based normalization of the data.

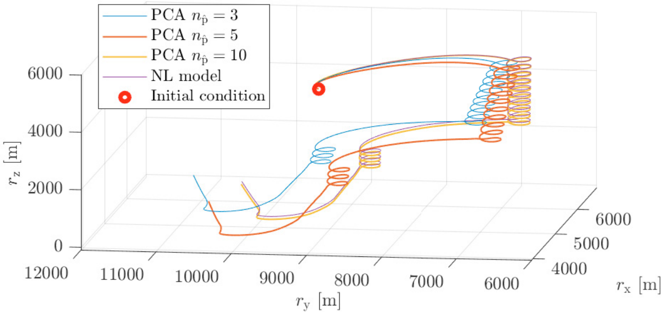
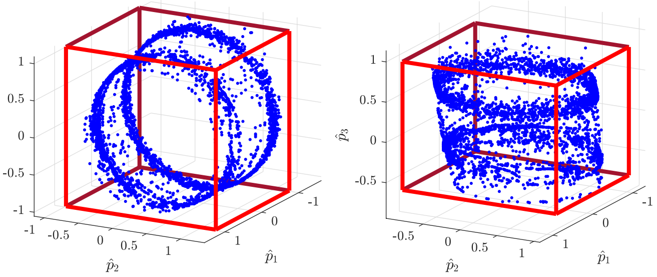


Fig. 5. Flight trajectory based on the original GPRV model compared to the simulated responses of the reduced LPV models with $n_{\hat{\text{p}}} \in \{3, 5, 10\}$.

4.3 Optimizing the conservativeness

We will briefly discuss the construction of $\hat{\mathbb{P}}$ when $n_{\hat{\text{p}}} = 3$. Defining $\hat{\mathbb{P}}$ as the convex hull around all $\hat{p}(k)$ generated with $\mathcal{D}_N$ reduces the conservatism, but likely results in a $\hat{\mathbb{P}}$ with many vertices, making LPV controller synthesis computationally intractable. Therefore, we use the Kabsch algorithm to construct a minimum-volume cube that encloses $U_{n_{\hat{\text{p}}}}^\top \bar{\Pi}_N$. The resulting cubes which define $\hat{\mathbb{P}}$ are depicted in Figure 6. To give an indication of the conservatism, we calculate the ratio between 'un-used' and 'used' space of $\hat{\mathbb{P}}$, i.e., $\frac{\text{Volume(cube)} - \text{Volume(polytope)}}{\text{Volume(polytope)}}$, with Volume(cube) the volume of the cube, and Volume(polytope) as the volume of the minimal convex hull around the scheduling trajectories $\hat{p}(k)$. The latter is computed with the Matlab function `convhulln`. The resulting ratios are 0.49 for the PCA based optimization and 0.54 for the DNN-based optimization for $n_{\hat{\text{p}}} = 3$, which implies that the conservativeness of both methods is similar.



(a) $\hat{\mathbb{P}}$ from PCA method.

(b) $\hat{\mathbb{P}}$ from DNN method.

Fig. 6. Construction of $\hat{\mathbb{P}}$, with the data-points $\hat{p}(k)$ in blue and the minimum-volume cube, defining $\hat{\mathbb{P}}$, in red.

5. CONCLUSIONS

This paper presents LPV modeling of the highly complex GPRV dynamics, where the LPV model is optimized over its complexity and conservativeness, such that it is suitable for LPV controller synthesis. From the results we conclude that we are able to obtain an affine LPV embedding of the complex nonlinear model of the GPRV with a scheduling dimension of 3. This is without requiring major simplifications of the dynamics, which is often done for tractability of GNC design. Both the PCA and DNN approaches obtain a good LPV model that is suitable for LPV controller synthesis. It must be noted that a clear advantage of the DNN method is that we have direct control over the variables participating in the new scheduling map, while there is no control over this for the PCA-based approach. For future work, we aim to design a high-performance LPV controller for the obtained LPV model and test its performance in closed-loop with a high-fidelity simulator of the vehicle.

ACKNOWLEDGEMENTS

We thank the SENER group and P.J.W. Koelewijn for useful discussions and simulation tools for the GPRV.

REFERENCES

Beck, C. (2006). Coprime factors reduction methods for linear parameter varying and uncertain systems. *Systems & Control Letters*, 55(3), 199–213.

Cacciatore, F., Ramos, H.R., Castellani, T.L., Figueroa, A., Veenman, A., Ramírez, S., Recupero, C., Kerr, M., and Béjar, J. (2019). The Design of the GNC of the Re-entry Module of Space Rider. In *Proc. of the 8th European Conference for Aeronautics and Space Sciences.*

Carter, L.H. and Shamma, J.S. (1996). Gain-scheduled bank-to-turn autopilot design using linear parameter varying transformations. *Journal of Guidance, Control, and Dynamics*, 19(5), 1056–1063.

Corti, A., Dardanelli, A., and Lovera, M. (2012). LPV methods for spacecraft control: An overview and two case studies. In *Proc. of the American Control Conference*, 1555–1560.

de Lange, M. (2021). Modeling of the Space Rider Flight Dynamics During the Terminal Descent Phase. Technical report, University of Technology Eindhoven.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning.* MIT Press.

Hecker, S. and Varga, A. (2005). Symbolic techniques for low order lft-modelling. In *Proc. of the 16th IFAC World Congress*, volume 38, 523–528.

Hoffmann, C. and Werner, H. (2015). LFT-LPV modeling and control of a Control Moment Gyroscope. In *Proc. of the 54th IEEE Conference on Decision and Control*, 5328–5333.

Hoffmann, C. and Werner, H. (2014). A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. *IEEE Transactions on Control Systems Technology*, 23(2), 416–433.

Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Koelewijn, P.J.W. and Tóth, R. (2020). Scheduling Dimension Reduction of LPV Models-A Deep Neural Network Approach. In *Proc. of the American Control Conference*, 1111–1117.

Kwiatkowski, A., Boll, M.T., and Werner, H. (2006). Automated Generation and Assessment of Affine LPV Models. In *Proc. of the 45th IEEE Conference on Decision and Control*, 6690–6695.

Kwiatkowski, A. and Werner, H. (2008). PCA-based parameter set mappings for LPV models with fewer parameters and less overbounding. *IEEE Transactions on Control Systems Technology*, 16(4), 781–788.

Luo, L., Xiong, Y., Liu, Y., and Sun, X. (2019). Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843.*

Marcos, A. and Balas, G.J. (2004). Development of Linear Parameter-Varying Models for Aircraft. *Journal of Guidance, Control, and Dynamics*, 27(2), 218–228.

Rizvi, S.Z., Abbasi, F., and Velni, J.M. (2018). Model reduction in linear parameter-varying models using autoencoder neural networks. In *Proc. of the American Control Conference*, 6415–6420.

Rizvi, S.Z., Mohammadpour, J., Tóth, R., and Meskin, N. (2016). A kernel-based PCA approach to model reduction of linear parameter-varying systems. *IEEE Transactions on Control Systems Technology*, 24(5), 1883–1891.

Rugh, W.J. and Shamma, J.S. (2000). Research on gain scheduling. *Automatica*, 36(10), 1401–1425.

Sadeghzadeh, A., Sharif, B., and Tóth, R. (2020). Affine linear parameter-varying embedding of non-linear models with improved accuracy and minimal overbounding. *IET Control Theory & Applications*, 14(20), 3363–3373.

Tóth, R. (2010). *Modeling and Identification of Linear Parameter-Varying Systems.* Lecture Notes in Control and Information Sciences, Vol. 403. Springer, Heidelberg.

Varga, A., Looye, G., Moormann, D., and Gräbel, G. (1998). Automated generation of LFT-based parametric uncertainty descriptions from generic aircraft models. *Mathematical and Computer Modelling of Dynamical Systems*, 4(4), 249–274.

Wu, F., Packard, A., and Balas, G. (1995). LPV control design for pitch-axis missile autopilots. In *Proc. of the 34th IEEE Conference on Decision and Control*, volume 1, 188–193.