LPVTools 2.0 and its Application to Spacecraft Attitude Control

Harald Pfifer * **Emily Burgin** *

* *Technische Universität Dresden, Dresden, 01062 Germany (e-mail: {harald.pfifer,emily.burgin}@tu-dresden.de).*

Abstract: This paper describes the newly included features in the LPVTools software suite and their application to the design of a flexible spacecraft attitude controller in the linear fractional transformation (LFT) framework. LPVTools is a Matlab Toolbox for modelling, analysis and controller synthesis for linear parameter-varying systems. It was originally developed by MUSYN Inc. and covers both LFT and gridded (Jacobian linearization) types of LPV system. The original version of LPVTools focused heavily on the latter. This paper presents newly added functionalities of the toolbox, which focus on the LFT framework.

Keywords: Linear parameter-varying systems, robust control, convex optimization, aerospace

1. INTRODUCTION

LPVTools is a Matlab Toolbox for working with grid-based and linear fractional transformation (LFT) based linear parameter varying (LPV) systems. Despite not being updated for 10 years, LPVTools is still contributing to the success of LPV methods in current control research and applications. For instance, the tools have been used recently for the design of fault tolerant flight control in Marcos et al. (2022), the active vibration control of elastic roof structures in Jirasek et al. (2024), observer design for vehicle control purposes in Fényes et al. (2021), control of guided projectiles in Vinco et al. (2023) and spacecraft attitude control in Burgin et al. (2023). The first two examples contain successful hardware tests, highlighting further the practical use of LPV control.

The toolbox was originally developed specifically with the design of aeroservoelastic control laws for lightweight, flexible Unmanned Aerial Vehicles (UAVs) in mind, see Hjartarson et al. (2015). These types of system heavily favour a grid-based approach as it is a natural extension of commonly used gain-scheduling approaches for flight control. Pfifer et al. (2015) provides a detailed overview of LPV applications to aeroservoelastic control. Hence, it comes as no surprise that the original release of LPVTools moslty provided algorithms applicable for grid-based LPV systems and only rudimentary support for LFT-based systems.

The release version of LPVTools was developed by MUSYN Inc. and published open source after the unfortunate passing of Dr. Gary Balas in 2014. Last year, after discussions with Dr. Pete Seiler, the authors of this paper took over maintenance and development of the toolbox. This paper presents the first major update since the original release ten years ago. The newest version of the tools, LPVTools 2.0, can now be found at

`https://github.com/Pharald/LPVTools`.

LPVTools 2.0 presented in this paper puts a major focus on updating the LFT functionalities of the toolbox and bringing them on par with the grid-based ones; described in Section 3. For many dynamical systems, e.g., ones based on multi-body dynamics, generating an LFT-based LPV model is more natural than a grid-based one. Examples for this include spacecraft dynamics, see Sanfedino et al. (2023) or industrial manipulators Saupe and Pfifer (2012). To this end, the full block S-Procedure from Scherer (1997) has been implemented in LPVTools 2.0. This adds so-called parameter dependent storage functions to the LPV controller synthesis for LFT-based systems on the basis of Wu and Dong (2006), which already existed for grid-based systems. In addition, an LFT-based state-feedback synthesis has been added to mimic the existing grid-based state-feedback synthesis. Finally, contractive coprime factorizations from Wood (1995) and a structured controller synthesis from Theis and Pfifer (2020) have been developed and implemented both for grid-based and LFT-based systems. The new methods are demonstrated on a spacecraft attitude control design problem in Section 4.

2. THE LPVTOOLS SOFTWARE SUITE

LPV systems are a class of systems whose state-space matrices are continuous functions of a time-varying parameter vector ρ that can be written in state space form:

$$P : \begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\rho(t)) & B(\rho(t)) \\ C(\rho(t)) & D(\rho(t)) \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}, \tag{1}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state vector, $u(t) \in \mathbb{R}^{n_u}$ the input vector, $y(t) \in \mathbb{R}^{n_y}$ the output vector and $A : \mathcal{P} \to \mathbb{R}^{n_x \times n_x}$, $B : \mathcal{P} \to \mathbb{R}^{n_x \times n_u}$, $C : \mathcal{P} \to \mathbb{R}^{n_y \times n_x}$, and $D : \mathcal{P} \to \mathbb{R}^{n_y \times n_u}$ are continuous functions of ρ. The trajectories of the parameter ρ are assumed to take values in a known compact subset $\mathcal{P} \subseteq R^{n_\rho}$, i.e $\rho \in \mathcal{T} := \{\rho : \mathbb{R} \to \mathcal{P} \subseteq R^{n_\rho}\}$. Additionally, they can have bounds on their variation rate, i.e. $\underline{\nu} \leq \dot{\rho}(t) \leq \bar{\nu}$ or are rate unbounded denoted by $-\infty \leq \dot{\rho}(t) \leq \infty$.

LPVTools is a software suite implemented in Matlab that provides a wide range of functionalities to work with such LPV systems, be it for modelling, analysis or controller synthesis. It uses object oriented programming to provide class-based data structure for two types of widely used LPV systems, namely grid-based (Jacobian linearization-based) and LFT-based. A detailed overview of the existing functionality of the initial version of LPVTools can be found in Hjartarson et al. (2015).

One of the most commonly used features of LPVTools is the synthesis of LPV controllers in the induced $\mathcal{L}_2$-norm defined by

$$\|P\| = \sup_{u\in\mathcal{L}_2\setminus\{0\},\rho\in\mathcal{T},x(0)=0} \frac{\|y\|_2}{\|u\|_2}. \tag{2}$$

A generalisation of the Bounded Real Lemma (Wu et al., 1996) provides a sufficient condition to upper bound the norm of a system $\|P\|$.

Theorem 1. (Wu et al., 1996): P is exponentially stable and $\|P\| \leq \gamma$ if there exists a continuously differentiable symmetric matrix function $X : \mathcal{P} \to \mathbb{R}^{n_x\times n_x}$ such that $X(p) \geq 0$ and

$$\begin{bmatrix} X(p)A(p) + A(p)^TX(p) + \partial X(p,q) & X(p)B(p) \\ B(p)^TX(p) & -I \end{bmatrix} + \frac{1}{\gamma^2}\begin{bmatrix} C(p)^T \\ D(p)^T \end{bmatrix}\begin{bmatrix} C(p) & D(p) \end{bmatrix} \leq 0 \tag{3}$$

hold for all $p \in \mathcal{P}$ and $\underline{\nu} \leq q \leq \bar{\nu}$, where ∂X is defined as $\partial X(p,q) = \sum_{i=1}^{n_\rho} \frac{\partial X}{\partial \rho_i}(p)q_i$.

Theorem 1 forms the basis of synthesis conditions for an LPV controller K that minimizes the induced $\mathcal{L}_2$-norm of the closed loop interconnection, i.e.,

$$\min_K \|F_l(P,K)\|, \tag{4}$$

where $F_l(P,K)$ is the lower fractional transformation. The details of the approach are described in Wu et al. (1996). In short, two linear matrix inequalities (LMIs) need to be satisfied, namely a filter inequality and a state feedback inequality. Both are coupled by a spectral radius condition. The solution of these coupled LMIs can be used to compute K.

2.1 Grid-based Approach

A grid-based LPV model is a finite collection of LTI models. These LTI systems represent (1) evaluated at a set $\{\rho_k\}_1^{n_{gp}}$ that spans the parameter space. It can be written in the form

$$\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\rho_k) & B(\rho_k) \\ C(\rho_k) & D(\rho_k) \end{bmatrix}\begin{bmatrix} x(t) \\ u(t) \end{bmatrix}, k = 1,\ldots,n_{gp}, \tag{5}$$

where $(A(\rho_k), B(\rho_k), C(\rho_k), D(\rho_k))$ represent the dynamics of the LPV system when ρ_k is held constant.

In LPVTools, the LPV system is represented by a `pss` data object which stores the gridded domain and the array that defines the state-space data at each point in the domain.

The original release of toolbox includes a vast range of functionalities for grid-based LPV systems including `lpvsyn` which solves the induced $\mathcal{L}_2$-norm control described in the previous section. Additionally, pole placement constraints can be included or stochastic LPV bounds can be used as a cost function. Besides the output feedback problem, LPVTools also includes `lpvsfsyn` for state feedback synthesis and `lpvestsyn` for estimator synthesis.

2.2 LFT-based Approach

LPV systems (1) can be represented in the LFT framework. For example, the state matrix A is cast into an LFT with a partitioned constant matrix $\bar{A}$ and parameter-dependent matrix $\Delta \in \mathbb{R}^{m_1\times n_1}$, where

$$\bar{A} = \left[\begin{array}{c|c} \bar{A}_{11} & \bar{A}_{12} \\ \hline \bar{A}_{21} & \bar{A}_{22} \end{array}\right] \in \mathbb{R}^{(n_1+n_2)\times(m_1+m_2)}. \tag{6}$$

The upper LFT $F_u(\bar{A}, \Delta(\rho(t)))$ is defined as

$$F_u(\bar{A}, \Delta(\rho(t))) = \bar{A}_{22} + \bar{A}_{21}\Delta(\rho(t))(I - \bar{A}_{11}\Delta(\rho(t)))^{-1}\bar{A}_{12}, \tag{7}$$

where $\Delta(\rho(t))$ takes a diagonal form

$$\Delta(\rho(t)) = \begin{bmatrix} \rho_1(t)I_{s_1} & & \\ & \ddots & \\ & & \rho_{n\delta}(t)I_{s_{n\delta}} \end{bmatrix}. \tag{8}$$

Within the toolbox, an LFT-based LPV system is stored as a `plftss` object. It is given in the state-space formulation and can be constructed from parameter varying matrices `plftmat` such as (7). The parameter variation $\rho(t)$ is defined and stored as a time-varying object using `tvreal`, similar to an uncertain object `ureal`. Within the object, the range and rate-bounds of the parameter can be defined.

A transition between the grid-based and LFT-based formulations is possible using the functions `grid2lft` and `lft2grid`. By transforming from an LFT to a grid, the system is evaluated on a grid of parameter values. In the reverse direction, the gridded parameter dependency is approximated as a rational function of the parameter. In version 1 of the toolbox, the functionality compatible with `plftss` objects is limited as the focus was predominantly on grid-based approaches (see Hjartarson et al. (2015)). Moreover, the LFT-based controller synthesis is based on the work of Packard (1994), which lacks the capabilities of it's grid-based counterpart. More discussion on this is given in later sections.

3. NEW TOOLBOX CAPABILITIES

In this section the most significant new functionalities of LPVTools 2.0 are described. The main focus was on implementing and updating LFT-based functionalities, specifically for `lpvsyn` and `lpvsfsyn`. Moreover, novel structured synthesis algorithms for mixed sensitivity control design have been implemented in `lpvmixsyn`. Additionally, version 2.0 includes various bug fixes, e.g. in `connect` that were in the release version.

3.1 Contractive Coprime Factorization

A new function `lpvlccf` for calculating the left contractive coprime factorisation of a system was implemented. This function is compatible with both `pss` and `plftss` objects. It is an extension of `rncf`, a function for calculating the

right normalized coprime factorisation. The applications of `lpvlccf` are for order reduction methods for parameter-dependent systems (Wood, 1995). Additionally, within the toolbox, this function is used to synthesise structured mixed-sensitivity controllers, as described in Section 3.3.

A left contractive coprime factorisation of an LPV system $P = M^{-1}N$ provides a kernel representation of all stable input-output pairs, see Wood (1995). The set of all input-output pairs is characterized by the null space of $[M,\ -N]$. In order to be contractive, the factorisation must also satisfy $\|[M,\ N]\| \leq 1$.

A state-space representation of $[M,\ N]$ for an LPV plant (1) is

$$\begin{bmatrix} \dot{\mu} \\ v \end{bmatrix} = \left[\begin{array}{c|cc} A+LC & L & B+LD \\ \hline R^{-0.5}C & R^{-0.5} & R^{-0.5}D \end{array}\right] \begin{bmatrix} \mu \\ y \\ -u \end{bmatrix}, \tag{9}$$

(*dependency on ρ is omitted from (9) for brevity*) where $L(\rho)$ is a filter gain matrix and $R(\rho) := I + D(\rho)D^\top(\rho)$.

The conditions of the contractive left-coprime factorisation can be derived by applying a generalisation of the Bounded Real Lemma to the state-space realisation of the coprime factorisation (9). This was proven in Wood (1995).

3.2 Full Block S-Procedure

The new LFT-based implementation of `lpvsyn` was completely overhauled to be in-line with the more mature grid-based capabilities of the toolbox. Whereas the previous implementation (Packard, 1994) solved the rate-unbounded case with constant storage functions, the updated implementation enables the user to define parameter dependent storage functions and rate-bounds for an LFT-based synthesis. It uses the approach presented in Wu and Dong (2006), based on the Full Block S-Procedure.

The synthesis LMIs, see Section 2, are transformed using the Full Block S-Procedure to formulate equivalent LMIs where the parameter dependent constraints are satisfied as a result of the transformation. Therefore the optimisation problem is finite dimensional and solvable. This works because it is possible to reformulate the LMIs into equivalent quadratic inequalities with LFT outer factors $G(\rho)$ such that:

$$X(p) = G^\top(p)X_0G(p), \text{ and } G(p) = F_u(G_0, \Delta(p)), \tag{10}$$

The following theorem states the Full Block S-Procedure. *For clarity off-diagonal blocks of block-diagonal matrices are omitted from notation.*

Theorem 2. (Full Block S-Procedure Wu and Dong (2006)). The quadratic matrix inequality, $X(p) < 0$, that satisfies (10) and $X_0 \in \mathbb{R}^{n_2 \times n_2}$, holds for all $p \in \mathcal{P}$ if and only if there exists a symmetric full block multiplier $\Pi \in \mathbb{R}^{(n_1+m_1)\times(n_1+m_1)}$ such that

$$\left[\begin{array}{cc} G_{0,11} & G_{0,12} \\ I & 0 \\ \hline G_{0,21} & G_{0,22} \end{array}\right]^\top \left[\begin{array}{c|c} \Pi & \\ \hline & X_0 \end{array}\right] \left[\begin{array}{cc} G_{0,11} & G_{0,12} \\ I & 0 \\ \hline G_{0,21} & G_{0,22} \end{array}\right] < 0 \tag{11}$$

and $\forall p \in \mathcal{P}$

$$\begin{bmatrix} I \\ \Delta(p) \end{bmatrix}^\top \Pi \begin{bmatrix} I \\ \Delta(p) \end{bmatrix} \geq 0. \tag{12}$$

where

$$\Pi = \begin{bmatrix} -\tilde{R} & \tilde{S} \\ \tilde{S}^\top & \tilde{R} \end{bmatrix}, \tag{13}$$

$\tilde{R}$ is diagonal, $\tilde{R} \leq 0$, $\tilde{S}$ is skew symmetric $\tilde{S} = -\tilde{S}^\top$ and both are of appropriate dimensions.

Proof. The proof is provided in the work of Wu and Dong (2006). The additional restriction on the multiplier Π ensures that the infinite dimensional constraint (12) is automatically satisfied, and thus not included in the computation.

By reformulating the LFT-based synthesis in this way, it is possible to define parameter-dependent storage functions. The parameter dependence must be rational (or reasonably approximated as rational). A common choice is an even numbered polynomial, for example, a 2nd order parameter dependent storage function would be

$$X(p) = [\, I \;\; Ip \,]\, X_0 \begin{bmatrix} I \\ Ip \end{bmatrix} \tag{14a}$$

$$= X_{11} + (X_{12} + X_{12}^T)p + X_{22}p^2 \tag{14b}$$

3.3 Mixed sensitivity control design

In the original release of LPVTools the function `lpvmixsyn` was included to mimic Matlab's robust control toolbox function `mixsyn`. The objective of the function was to find an LPV controller K that minimizes the induced L_2-norm of the weighted closed-loop input/output mapping

$$\begin{bmatrix} W_1S \\ W_2KS \\ W_3T \end{bmatrix}, \tag{15}$$

where $S = (I+PK)^{-1}$ is the sensitivity function, $T = I-S$ the complementary sensitivity function and W_1, W_2, W_3, and W_4 weighting filters. This function served mostly academic purposes and was rarely useful for designing controllers for real life applications. In LPVTools 2.0, it has been updated to minimize the following weighted 4-block problem

$$\begin{bmatrix} W_1 & 0 \\ 0 & W_3 \end{bmatrix} \begin{bmatrix} S & -SP \\ KS & -KSP \end{bmatrix} \begin{bmatrix} W_2 & 0 \\ 0 & W_4 \end{bmatrix}. \tag{16}$$

This 4-block problem has been successfully applied to many industrial problems, e.g. for spacecraft attitude control (Burgin et al., 2023).

The function `lpvmixsyn` can use both `pss` or `plftss` objects as an input and is fully compatible with the full block S-procedure-based synthesis described in the previous section. In addition to the standard output feedback solution, a novel structured synthesis based on a filter/state-feedback problem is also included following the results from Theis and Pfifer (2020).

If the flag '`struc`' is used when calling `lpvmixsyn`, the following structure is imposed on the controller K:

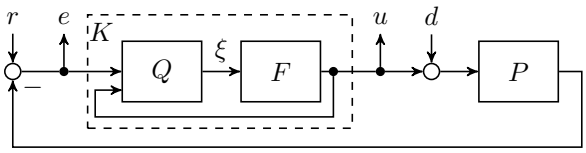

Fig. 1. Unweighted four-block mixed-sensitivity problem

$$\begin{bmatrix} \dot{\xi} \\ \xi \end{bmatrix} = \begin{bmatrix} A(\rho) + L(\rho)C(\rho) & L(\rho) & B(\rho) \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi \\ e \\ u \end{bmatrix}, \tag{17a}$$

$$u = F(\rho)\,\xi, \tag{17b}$$

where the dynamics of the controller are prescribed by a filter Q given by (17a) and a feedback gain F, as depicted graphically in Fig. 1. By using this specific structure in the 4-block problem, the filter and state-feedback LMI can be decoupled, i.e., the spectral radius condition can be removed. This provides a computationally significantly more efficient algorithm that still provides guaranteed performance bounds and an exploitable controller structure, e.g. for anti wind-up.

The idea is briefly sketched here for the unweighted problem. The interested reader is referred to Theis and Pfifer (2020) for a detailed derivation. Recall that the performance of the controller is given by the induced L_2-norm of the closed loop input-output map, i.e.

$$\left\| \begin{bmatrix} S & -SP \\ KS & -KSP \end{bmatrix} \right\|_2. \tag{18}$$

Replacing the plant P by its contractive left coprime factorization yields

$$\left\| \begin{bmatrix} S \\ KS \end{bmatrix} M^{-1} \begin{bmatrix} -N M \end{bmatrix} \right\|_2 \leq \left\| \begin{bmatrix} S \\ KS \end{bmatrix} M^{-1} \right\|_2. \tag{19}$$

The right hand side of (19) represents an input weighted two-block mixed sensitivity problem, which bounds the norm of the original 4-block problem (18). Computing the filter Q based on the contractive left coprime factorization turns the input weighted two-block problem into a simple state-feedback problem for the feedback gain F in (17). In other words, the output-feedback mixed sensitivity problem can be solved sequentially by first computing a contractive left coprime factorization of the generalized plant `lpvlccf` and then solving a specific state-feedback problem with `lpvsfsyn` . The performance bound of the state-feedback problem provide a bound on performance of the original output-feedback problem.

4. APPLICATION BENCHMARK: CONTROLLER DESIGN FOR AN OBSERVATION SATELLITE

The additional capabilities of the toolbox were tested on a small satellite model.

4.1 Satellite Model

The satellite in question weighs 95 kg, and has a comparatively large flexible solar array. The lowest undamped frequency of its flexible modes is 5.6 rad/s. Additionally, the spacecraft is modelled with a draining fuel tank (no sloshing dynamics) positioned at the centre of mass of the central body. Hence, the mass properties of the

Table 1. Parameters and notation used in (20) and figure 2.

Parameters	Description
F_{ext}	Force vector applied to central body
$T_{ext,\mathrm{B}}$	Torque vector applied to central body at B
$F_{\mathcal{B}/\mathcal{A}}$	Force vector applied by $\mathcal{B}$ on $\mathcal{A}$
$T_{\mathcal{B}/\mathcal{A},\mathrm{P}}$	Torque vector applied by $\mathcal{B}$ on $\mathcal{A}$ at point P
$\ddot{r}$	Linear acceleration
$\ddot{\theta}$	Rotational acceleration
m	Mass
J	Inertia
τ_{AP}	Kinematic model between points A and P
Notation	**Denoting**
$\mathcal{B}$	Central body
B	Center of mass of central body
P	Position of revolute joint
$\mathcal{A}$	Solar array
A	Center of mass of solar array

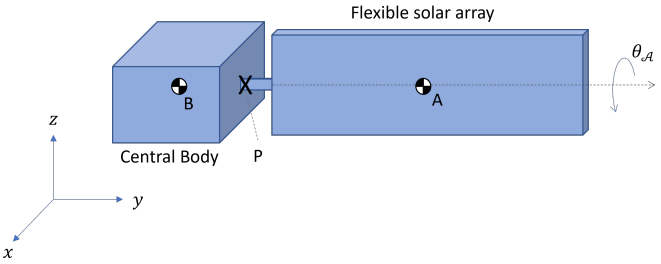

Fig. 2. Visual of spacecraft model

spacecraft are time-varying. The spacecraft is in a sun-synchronous, 400 km altitude orbit. At this altitude, the considered disturbances (such as solar radiation pressure torque, aerodynamic torque, gravity gradient torque and magnetic torque) reach magnitudes of 1×10^{-3} Nm (see, e.g., Hughes (2012)).

The spacecraft P is modelled with a cuboid, rigid central body $P_{\mathcal{B}}$ and the flexible solar array $P_{\mathcal{A}}$ is modelled as a cantilever beam attached by a revolute joint at point P; pictured in figure 2. The position of the solar array's centre of mass is denoted A. The following spacecraft model was generated using the Satellite Dynamics Toolbox (Alazard and Sanfedino, 2020); see the documentation for extensive explanation.

$$\begin{aligned} P_{\mathcal{B}}: \begin{bmatrix} F_{ext} \\ T_{ext,\mathrm{B}} \end{bmatrix} &= \begin{bmatrix} m_{\mathcal{B}}(m_f) I & 0 \\ 0 & J_{\mathcal{B}}(m_f) \end{bmatrix} \begin{bmatrix} \ddot{r}_{\mathrm{B}} \\ \ddot{\theta} \end{bmatrix} + \\ &\quad \begin{bmatrix} 0 \\ \dot{\theta} \times J_{\mathcal{B}} \dot{\theta} \end{bmatrix} \\ P_{\mathcal{A}}: \begin{bmatrix} F_{\mathcal{B}/\mathcal{A}} \\ T_{\mathcal{B}/\mathcal{A},\mathrm{P}} \end{bmatrix} &= \tau_{\mathrm{AP}}^{\top} \begin{bmatrix} m_{\mathcal{A}} I & 0 \\ 0 & J_{\mathcal{A}} \end{bmatrix} \tau_{\mathrm{AP}} \begin{bmatrix} \ddot{r}_{\mathrm{P}} \\ \ddot{\theta} \end{bmatrix} + \\ &\quad L_{\mathrm{P}}^{\top} \ddot{\eta} \\ -L_{\mathrm{P}} \begin{bmatrix} \ddot{r}_{\mathrm{P}} \\ \ddot{\theta} \end{bmatrix} &= \ddot{\eta} + diag\{2\xi_i\omega_i\}_{i=1}^{3} \dot{\eta} + diag\{\omega_i^2\}_{i=1}^{3} \eta \end{aligned} \tag{20a}$$

See Table 1 for details of the notation. The central body experiences external forces F_{ext} and torques T_{ext} about the centre of mass. The dynamic coupling between the two bodies is transmitted via the joint at point P. The flexible

modes of the solar array are described by their modal contributions L_{P}. Each second-order mode (denoted by subscript i) has damping ξ_i and natural frequency ω_i.

The central body mass $m_{\mathcal{B}}$ is a function of the fuel mass m_f, which is calculated from the commanded torque u and specific impulse $\mathrm{I_{sp}}$ using the thrust equation. The central body inertia about the centre of mass $J_{\mathcal{B}}$ also changes with fuel mass. To exaggerate the effects of the changing mass properties, it is assumed that 25% of the spacecraft's total mass is fuel, so a full fuel tank weighs 23.75 kg. For this application, only rotational dynamics are considered for attitude control, so the translational motion r of the central body is disregarded. For the control problem, the system output is the attitude θ which must track a target attitude relative to an inertial frame. The external disturbances are modelled as a summative input disturbance on the commanded torque $T_{ext} = d+u$. Hence, the linear plant takes the general LPV plant structure.

The application case considers a single-axis slew manoeuvre using thrusters. The controller for this case is synthesized with the fuel mass m_f as the scheduling parameter ρ. This has a range of 0 to 23.75 kg, which translates to a total satellite mass range of 71.25 kg to 95 kg. Consider three thrusters provide torque about each axis of the spacecraft. They are each positioned with lever arm of 10 cm from the centre of mass of the central body. The maximum force produced by each thruster is assumed to be 20 N with an $\mathrm{I_{sp}}$ of 100 s as this is reasonable for a small satellite (Zakirov et al., 2001). This results in a maximum torque about each axis of 2 Nm and maximum mass loss rate of 0.02 kg/s, which is used as the rate-bound in the synthesis. The guidance profile, which the spacecraft must follow for the slew, is generated as a rectangular acceleration signal, then integrated twice to provide a reference attitude signal.

4.2 Comparison of Functions

The results from the newly implemented LFT-based synthesis functions were compared to the grid-based and LFT-based results in version 1 of the Toolbox. Both grid- and LFT-based implementations of the new structured synthesis approach were also analysed.

In all cases, the mixed-sensitivity weighting scheme (16) was applied, with identical weights W_1, W_2, W_3, W_4 for each. Furthermore, equivalent 2nd order grid-based and LFT-based basis functions were used such that the syntheses are directly comparable. The tuning of the controller was done using the process described by Burgin et al. (2023). A bode plot of the resulting 4-block problem performance with respect to the design weights is provided in Figure 3. It is clear from this plot that the new implementation of the LFT-based synthesis is more in-line with the grid-based synthesis, as the resulting loop-shapes align well. In contrast, the old implementation exhibits different behaviour to all other controllers. In addition, the structured synthesis approach is plotted for comparison. The plot demonstrates that the controllers again perform very similarly to the output-feedback approach, with the added advantage of faster synthesis time. To further support these results, the synthesis γ values (i.e. the upper bounds on the induced $\mathcal{L}_2$-norm of the closed loops) are provided in Table 2). The LFT-based synthesis

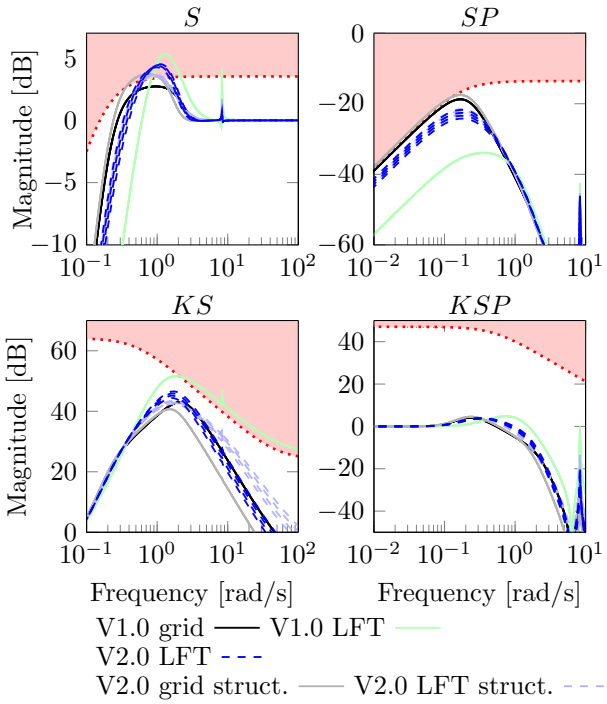

Fig. 3. Comparison of 4-block problem performance. Only x-axis transfer functions are plotted for clarity.

Table 2. Synthesis γ values

	Grid-Based	LFT-Based
V1.0	0.960	**1.98**
V2.0	0.960	1.53
V2.0 Structured	1.500	1.82

from V1.0 of the toolbox has a γ more than two times as high as the equivalent grid-based implementation, whereas the updated LFT implementation is only 1.6 times higher. The γ values for the structured synthesis approaches are also provided for comparison.

Simulations of the spacecraft performing the slew were also completed for all controllers. Plotted in Figure 4 is a comparison between the grid-based and LFT-based controllers implemented in V2.0 of the toolbox. The spacecraft model in the slew scenario is initialized with a full fuel tank. A second simulation intialized with an almost empty tank was also conducted to verify that the LPV controller works across the whole domain of m_f, but it is not plotted here to save space as the performance was almost identical. The discrepancy in the performance of the two implementations is very small. Furthermore, when compared to the structured implementations (for both grid and LFT), there is also minimal discrepancy, although it has been omitted from the plot for clarity. It is clear from this plot and the other results presented in this section that the updated LFT-based implementation has now been brought on par with the grid-based implementation.

5. CONCLUSION

To conclude, the Matlab Toolbox LPVTools has been updated by the integration of important LFT-based algorithms using the full block S-procedure. Moreover, a

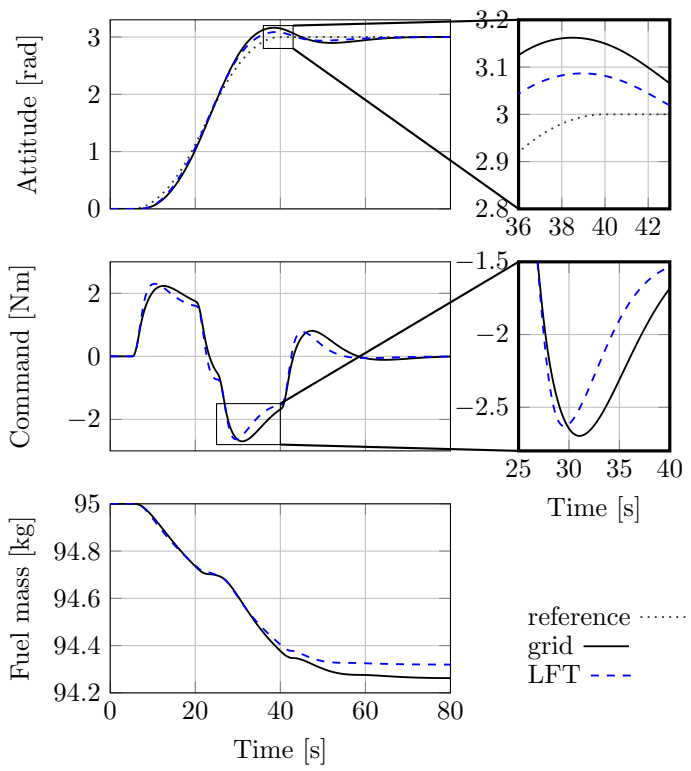

Fig. 4. Comparison of time-domain performance about the x-axis.

novel structured synthesis algorithm has been added. A brief description of the updates was provided and the new functionality was demonstrated on a spacecraft application. With the presented updates, the gap between grid-based and LFT-based capabilities within LPVTools has been bridged.

ACKNOWLEDGEMENTS

This paper is dedicated to Dr. Gary Balas and Dr. Andy Packard who passed away too soon. Dr. Balas and Dr. Packard were strong advocates of the Linear Parameter Varying Framework and even stronger advocates of providing easy to use software tools to the practising control engineer. We hope that they appreciate our effort to continue developing LPVTools in their spirit.

REFERENCES

Alazard, D. and Sanfedino, F. (2020). Satellite dynamics toolbox for preliminary design phase. In *43rd Annual AAS Guidance and Control Conference*, volume 30, 1461–1472.

Burgin, E., Biertümpfel, F., and Pfifer, H. (2023). Linear parameter varying controller design for satellite attitude control. In *22nd IFAC World Congress*, 3455–3460.

Fényes, D., Hegedűs, T., Németh, B., and Gáspár, P. (2021). Observer design with performance guarantees for vehicle control purposes via the integration of learning-based and lpv approaches. In *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, 122–127. IEEE.

Hjartarson, A., Seiler, P., and Packard, A. (2015). Lpvtools: A toolbox for modeling, analysis, and synthesis of parameter varying control systems. *IFAC-PapersOnLine*, 48(26), 139–145.

Hughes, P.C. (2012). *Spacecraft attitude dynamics*. Courier Corporation.

Jirasek, R., Schauer, T., and Bleicher, A. (2024). Linear parameter-varying output-feedback for active vibration control of an elastic kinetic roof structure with experimental validation. *Engineering Structures*, 307, 117887.

Marcos, A., Waitman, S., and Sato, M. (2022). Fault tolerant linear parameter varying flight control design, verification and validation. *Journal of the Franklin Institute*, 359(2), 653–676.

Packard, A. (1994). Gain scheduling via linear fractional transformations. *Systems & Control Letters*, 22(2), 79–92.

Pfifer, H., Moreno, C.P., Theis, J., Kotikapuldi, A., Gupta, A., Takarics, B., and Seiler, P. (2015). Linear Parameter Varying Techniques Applied to Aeroservoelastic Aircraft: In Memory of Gary Balas. *IFAC-PapersOnLine*, 48(26), 103–108.

Sanfedino, F., Alazard, D., Kassarian, E., and Somers, F. (2023). Satellite dynamics toolbox library: a tool to model multi-body space systems for robust control synthesis and analysis. *IFAC-PapersOnLine*, 56(2), 9153–9160.

Saupe, F. and Pfifer, H. (2012). An Observer Based State Feedback LFT LPV Controller for an Industrial Manipulator. *IFAC Proceedings Volumes*, 45(13), 337–342.

Scherer, C.W. (1997). A full block s-procedure with applications. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 3, 2602–2607. IEEE.

Theis, J. and Pfifer, H. (2020). Observer-based synthesis of linear parameter-varying mixed sensitivity controllers. *International journal of robust and nonlinear control*, 30(13), 5021–5039.

Vinco, G.M., Theodoulis, S., Sename, O., and Strub, G. (2023). Uneven grid-based linear parameter-varying controller design for guided projectiles. *IFAC-PapersOnLine*, 56(2), 4496–4501.

Wood, G.D. (1995). *Control of Parameter-Dependent Mechanical Systems*. University Cambridge.

Wu, F. and Dong, K. (2006). Gain-Scheduling Control of LFT Systems Using Parameter-Dependent Lyapunov Functions. *Automatica*, 42(1), 39–50. doi:10.1016/j.automatica.2005.08.020.

Wu, F., Yang, X.H., Packard, A., and Becker, G. (1996). Induced L2-norm control for LPV systems with bounded parameter variation rates. *Int J Robust and Nonlinear Control*, 6(9-10), 983–998.

Zakirov, V., Sweeting, M., Erichsen, P., and Lawrence, T. (2001). Specifics of small satellite propulsion: Part 1. In *15th AIAA/USU Conference on Small Satellites*.