Modelling and Control of LPV systems

Roland Tóth[a,b] and Chris Verhoek[a]
[a]Control Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
[b]Systems and Control Laboratory, HUN-REN Institute for Computer Science and Control, Budapest, Hungary

E-mail addresses: `r.toth@tue.nl, c.verhoek@tue.nl`

KEY POINTS & OBJECTIVES

- Introduction to the theoretic concept of *linear parameter-varying* (LPV) systems and its use as surrogate description of nonlinear and time-varying systems.
- Overview of the modelling methods in terms of model conversion and data-driven techniques, following either a local or a global modelling philosophy.
- Summary of popular LPV control synthesis methods for dynamic output and state-feedback design.

Abstract

The framework of *linear parameter-varying* (LPV) systems has been established with the purpose to represent complex nonlinear and/or time-varying behaviour in terms of simple linear surrogate models that can easily be used for analysis or design of controllers. It has become a manifestation of a dream in engineering about the extension of powerful and well-understood tools available for *linear time-invariant* (LTI) systems to allow easy, but efficient analysis and control design for applications with nonlinear/time-varying aspects. The resulting LPV methods provide stability and performance guarantees and have contributed seriously to the advancement of aerospace engineering, automotive technologies, and the mechatronic industry over the years. In this article, a concise overview of the main theoretic concepts behind of LPV system representations is given, together with an overview of the modelling methods in terms of model conversion and data-driven techniques, following either a local or a global modelling philosophy. Furthermore, we review LPV control synthesis methods for dynamic output and state-feedback design, whose effectiveness has made this framework a popular choice in practice.

Keywords: Control synthesis; Data-driven modeling; Gain scheduling; Generalized plants; Global modelling; Hinf control; LFRs; Linear Fractional Representations; Linear parameter-varying systems; LMIs; Local modelling; LPV control; LPV modelling; Nonlinear system modeling; Output feedback control; Polytopic systems; State feedback control; Surrogate modeling; System identification.

1 Background

Since the 1980s, rapid rise of performance expectations in engineering in terms of accuracy, energy efficiency, etc., resulted in the realisation that, to reach the desired performance targets, design principles that focussed on operating systems in ranges where the behaviour is approximately *linear time-invariant* (LTI) had to be abandoned and the commonly *nonlinear* (NL) and *time-varying* (TV) nature of the underlying physical systems must be handled by control design. This required better models and design approaches, which initiated a significant research effort spent on identification and modelling of NL and *linear time-varying* (LTV) systems, as well as rapid theoretical development of NL control. Despite many theoretical solutions, dealing with NL models without any structure has been found infeasible in practice, both in terms of identification and control.

Engineers working in the industry, even today, still prefer the application of LTI control design methods, due to intuitive and reliable approaches of loop shaping, sequential loop closing, optimal-gain, and robust control. These approaches are preferred as they guarantee high performance and reliability, have easy and intuitive performance shaping schemes, and there is a vast industrial experience in their application. Additionally, it has also been observed in practice that many NL systems can be well approximated by multiple LTI models that describe the behaviour of the plant around some operating points. This recognition manifested in the 1980s, that instead jumping into the deep-space of NL and TV systems, a model class is needed which can serve as an extension of the existing LTI control approaches, but is still able to incorporate NL and TV dynamical aspects. This has led to the birth of *linear parameter-varying* (LPV) systems through the idea of *gain-scheduling* (Shamma and Athans, 1992).

In gain-scheduling, the basic concept has been to linearise the NL system model at different operating points, resulting in a collection of local LTI descriptions of the plant. Then subsequently, LTI controllers are designed for each local aspect. These controllers are interpolated to give a control solution to the entire operation regime (Rugh, 1991). The used interpolation function is called the *scheduling function* in this framework and it is dependent on the current operating point of the plant. To describe the changes of the operating point, a signal is introduced, which is called the *scheduling signal*, often denoted by p. In this way, the parameters of the resulting controller are dependent on the varying signal p, hence the name *parameter-varying*, while the dynamic relation between the inputs and outputs of the controller remain *linear*. Due to many successful applications of this design methodology, gain-scheduling has become popular in industrial applications, even if guarantees for overall stability of the designed LPV controllers operated systems have not been available and the possibility of malfunction has existed. After 20 years, this was resolved by the introduction of interpolation-based methods that aimed to provide various forms of global stability guarantees, see *e.g.*, (Stilwell and Rugh, 2000; Wang et al., 2024). In the mean time it was also realised that many NL systems can be directly converted into an LPV form without any linearisation via the concept of *LPV embedding* (Lee, 1997; Tóth, 2010). Besides of a growing number of NL model conversion methods, data-driven approaches have also appeared that provided direct LPV models for control without the laborious process of NL system modelling (for an overview, see Tóth (2010)). LPV control has gained momentum during the 1990s, when the first results on the extension of $\mathcal{H}_\infty$ and $\mathcal{H}_2$ optimal control through *linear matrix inequalities* (LMIs) based optimization appeared (see Scherer (1996); Apkarian and Gahinet (1995); Packard (1994)) together with μ-synthesis approaches, originating from robust LTI control (Zhou et al., 1995). Contrary to the former gain-scheduling methods, these approaches guarantee stability, optimal performance, and robustness over the entire operating regime of LPV models. Since then, the LPV field has evolved rapidly in the last 35 years and has become a powerful framework for modern industrial control with a wide range of applications in commercial and military aviation (Marcos and Balas, 2004; Papageorgiou et al., 2000; Verdult et al., 2004), satellites and re-entry vehicles (Veenman et al., 2009; Preda et al., 2018; de Lange et al., 2022), engine and active suspension control in the automotive industry (Baslamisli et al., 2009; Do et al., 2010), wind turbines (Bianchi et al., 2007; Lescher et al., 2006), high-precision mechatronics (Wassink et al., 2004; Steinbuch et al., 2003; Broens et al., 2022), but also for internet web servers (Qin and Wang, 2007; Tanelli et al., 2009), and in environmental modelling (Belforte et al., 2005).

2 Modelling with LPV systems

The concept of LPV systems has been introduced to provide a surrogate representation of NL and TV dynamic behaviours in terms of a linear dynamic model, facilitating the extensions of powerful *linear time-invariant* (LTI) control approaches such as $\mathcal{H}_2/\mathcal{H}_\infty$ optimal control and *model predictive control* (MPC), see *e.g.*, Scherer (1996); Mohammadpour and Scherer (2011); Besselmann et al. (2012); Cisneros (2021). Signal relations between the inputs and outputs in an LPV representation are assumed to be linear, but, at the same time, dependent on the *scheduling variable* p (n_p-dimensional signal). The scheduling variable is assumed to be a measurable and free (external) signal in the modelled system and is taking values from a so-called *scheduling region* $\mathbb{P} \subseteq \mathbb{R}^{n_\mathrm{p}}$, often restricted to be a compact set. In this way, variation of p represents time-variance, changing operating conditions, etc., and aims at the embedding of the original NL/TV behaviour into the solution set of an LPV system representation (Tóth, 2010; Rugh and Shamma, 2000). While the former objective is pursued by the so-called *global* LPV modelling approaches, alternatively, one can aim at the approximation of the NL/TV behaviour by the interpolation of various linearisations of the system around operating points or signal trajectories, often referred to as *local* modelling, see, *e.g.*, (Bachnas et al., 2014; Petersson and Löfberg, 2012; Rugh and Shamma, 2000). The two approaches of global and local modelling lead to two different philosophies of capturing the original system behaviour in terms of LPV models, which inherently affects the properties and provided guarantees of the subsequent LPV analysis and control design based on them.

2.1 Global modelling concept

First the global modelling approach is discussed, together with the resulting theoretical concept of LPV systems. Consider a nonlinear system G given in terms of the *state-space* (SS) representation:

$$\mathrm{q}x(t) = f(x(t), u(t)), \tag{1a}$$

$$y(t) = h(x(t), u(t)), \tag{1b}$$

where q denotes the differential operator $\mathrm{q}x(t) = \frac{\mathrm{d}}{\mathrm{d}t}x(t) = \dot{x}(t)$ if the considered system is defined in *continuous-time* (CT), or the time-shift operator $\mathrm{q}x(t) = x(t+1)$ for *discrete-time* (DT) systems. Furthermore, $x(t)$ is the state, $u(t)$ and $y(t)$ being the input and output signals of the system, and $t \in \mathbb{T}$ is the time with $\mathbb{T} = \mathbb{R}$ in CT and $\mathbb{T} = \mathbb{Z}$ in DT, while f and h are assumed to be possibly nonlinear, but bounded functions on the sets

$$x(t) \in \mathbb{X} \subseteq \mathbb{R}^{n_\mathrm{x}} \quad \text{and} \quad u(t) \in \mathbb{U} \subseteq \mathbb{R}^{n_\mathrm{u}},$$

and $y(t) \in \mathbb{Y} \subseteq \mathbb{R}^{n_\mathrm{y}}$. The sets $\mathbb{X}$, $\mathbb{U}$ and $\mathbb{Y}$ represent the regions of interest where analysis or control of the NL system is intended. Hence, these sets are often considered to be closed and bounded.

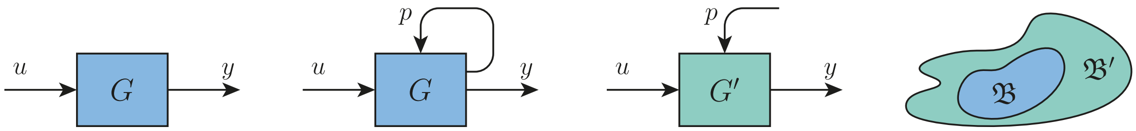
As an example, consider an unbalanced disc system, *i.e.*, a disc with an off-centered mass driven by a DC motor. By neglecting the fast electrical dynamics of the motor, the motion of this system in CT can be described as

$$\dot{x}_1(t) = x_2(t), \tag{2a}$$

$$\dot{x}_2(t) = \frac{mgl}{J}\sin(x_1(t)) - \frac{1}{\tau}x_2(t) + \frac{\kappa_\mathrm{m}}{\tau}u(t), \tag{2b}$$

where $x_1(t) \in \mathbb{R}$ is the angular position of the mass, $x_2(t) \in \mathbb{R}$ is its angular velocity and $u(t) \in \mathbb{R}$ is the input voltage to the motor. The physical coefficients $m, g, l, \tau, \kappa_\mathrm{m}$ are given in (Kulcsár et al., 2009) and the regions of interest can be considered as $\mathbb{X} = [-\pi, \pi]$ and $\mathbb{U} = [-10, 10]$. The angular position of the mass x_1 is taken as the output $y(t)$ of the plant with $\mathbb{Y} = [-\pi, \pi]$.

Let $\mathfrak{B} \subseteq (\mathbb{Y} \times \mathbb{U})^\mathbb{T}$ ($\mathbb{W}^\mathbb{T}$ stands for all maps from $\mathbb{T}$ to $\mathbb{W}$) containing all trajectories of (y, u) that are compatible with G, *i.e.*, they are solutions of (1). We call $\mathfrak{B}$ the (manifest) behaviour of the system G. The main idea in LPV modelling is to introduce an auxiliary variable $p = \psi(x, u) : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^{n_\mathrm{p}}$, with range $\mathbb{P}$ satisfying $\psi(\mathbb{X}, \mathbb{U}) \subseteq \mathbb{P} \subseteq \mathbb{R}^{n_\mathrm{p}}$, and reformulate G as shown in Fig. 1.b, where it holds true that if the loop is disconnected and p is assumed to be a known signal as in Fig. 1.c, then the "remaining" relations between y and u are linear,



(a) Original NL system. (b) characterization of p. (c) LPV form by disconnecting p. (d) Relation of the resulting behaviours.

Fig. 1 The concept of global LPV embedding of a nonlinear system.

leading to a representation of the system as

$$G' : \begin{cases} \mathrm{q}x(t) = A(p(t))x(t) + B(p(t))u(t), \\ y(t) = C(p(t))x(t) + D(p(t))u(t), \end{cases} \tag{3}$$

where A, B, C, D are matrix function with appropriate dimensions, satisfying that

$$f(x,u) = A(\psi(x,u))x + B(\psi(x,u))u, \tag{4a}$$
$$h(x,u) = C(\psi(x,u))x + D(\psi(x,u))u, \tag{4b}$$

for all $(x,u) \in \mathbb{X} \times \mathbb{U}$. This can be achieved in our example (2) by taking, as a possible choice, $p = \frac{\sin(x_1)}{x_1} := \mathrm{sinc}(y)$, resulting in:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ \frac{mgl}{J}p(t) & -\frac{1}{\tau} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{\kappa_\mathrm{m}}{\tau} \end{bmatrix} u(t), \tag{5a}$$
$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t). \tag{5b}$$

Applying this reformulation with a disconnected p and assuming that all trajectories of p are allowed, *i.e.*, p is a free variable with $p \in \mathbb{P}^\mathbb{T}$ independent of (x,u), the possible trajectories of this reformulated system G' form a solution set of (3) in terms of (x,u,y) trajectories for which there exists a $p \in \mathbb{P}^\mathbb{T}$ such that (3) is satisfied, denoted as $\mathfrak{B}'$, which contains $\mathfrak{B}$, as visualized in Fig. 1.d. This concept of formulating G' as a linear, but p-dependent description of G, enables the use of simple stability analysis and convex controller synthesis, see *e.g.*, Scherer (1996); Apkarian and Gahinet (1995); Packard (1994); Mohammadpour and Scherer (2011), which can be conservative w.r.t. G, but computationally more attractive and robust than other approaches directly addressing $\mathfrak{B}$. Specifically, control synthesis based on the above mentioned modelling procedure results in the implementation of an LPV controller K visualized in Figure 3. It is obvious that a key assumption is that p must be “observable” from the real system. The observed value of p is required to complete the hidden relation of p to the other variables in (3) in terms of ψ and enable a linear controller to continuously change, *i.e.*, “schedule”, its behaviour according to p to regulate (1). Hence, *scheduling variable* has become a popular name for p, while ψ is often called the *scheduling map*. The overall representation and control concept can be seen as a multi-path feedback linearisation, similar to the well-known approach in NL system theory, see (Isidori, 1995), as the obtained information from the system in terms of p is fed back to arrive to a varying linear relation (3) (in contrast with the NL theory where the resulting behaviour is intended to be LTI).

Following the above procedure, the scheduling variable p itself can appear in many different relations w.r.t. the original variables (x,u,y). If p is a free variable w.r.t. G, *e.g.*, wind speed for a wind turbine, appearing as part of the input signals: $p = \psi(u_\mathrm{d})$ with $u = [\ u_\mathrm{d}^\top \ \ u_\mathrm{c}^\top\ ]$ where u_c are signals available for actuation and u_d are external effects, then we can speak about a *true parameter-varying system* without conservativeness. This also includes the special case of pure time variation, *i.e.*, $p(t) = t$, when (3) becomes a *linear time-varying* (LTV) representation, showing that LTV systems are a subclass of LPV systems. However, in many practical applications, like in our example, p depends on other signals, like outputs, states or general inputs of the modelled system (*e.g.*, operating conditions). Such situations are sometimes warningly labelled to be *quasi*-LPV (q-LPV) in the literature, which is a certain misconception of why LPV systems were introduced in the first place. What happens in this common case is that we willingly disregard the connection of p in terms of ψ to (x,u) and we deliberately assume that any variation of p can take place in $\mathbb{P}$, independently from these signals. This assumed freedom of p gives the advantage of linearity of the resulting varying-relation, enabling the utilization of a whole arsenal of powerful analysis and control synthesis techniques, while its price is the introduction of conservativeness in the embedding of the nonlinear behaviour, *i.e.*, there are solution trajectories in $\mathfrak{B}'$ that were not part of the original behaviour $\mathfrak{B}$ (see Fig. 1.d), and which the controller still need to address to provide stability and the expected performance during design. Hence, one important objective of LPV modelling, besides achieving complete embedding, is to *minimize* such *conservativeness*.

LPV model conservativeness, which is a measure of the difference of admissible solution trajectories of the original NL model and the embedded LPV model, is directly connected to the choice of the scheduling map $p = \psi(x,u) : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^{n_\mathrm{p}}$ and the scheduling range $\psi(\mathbb{X},\mathbb{U}) \subseteq \mathbb{P}$. It can be quantified in terms of gap-metric-based and admissible-set-based formulations. Its main cause is that choosing n_p high is often advantageous to reduce complexity of the LPV model (see it later). However, in the resulting high-dimensional space, the typical trajectories of (x,u) that occur during operation of the system can map to a highly narrow region in $\mathbb{P}$, seriously increasing the gap between $\mathfrak{B}$ and $\mathfrak{B}'$ in Figure 1.d. Furthermore, convex LPV analysis and control design approaches often rely on the assumption that $\mathbb{P}$ is convex, satisfying $\psi(\mathbb{X},\mathbb{U}) \subseteq \mathbb{P}$, which, in practice, leads to the choice of simple polytopic sets, such as hyper-rectangle-based descriptions of $\mathbb{P}$, based on min-max values of $\psi(\mathbb{X},\mathbb{U})$. While this results in a simple modelling process, it can introduce extreme conservatism in case of high n_p dimensions. In practice, reliable methods are available to deal with (i) reduction of the scheduling dimension and (ii) optimization of the number of vertices and volume of $\mathbb{P}$ in the modelling process, see, *e.g.*, (Sadeghzadeh et al., 2020). Beyond characterising the range of p in which it can realistically vary through $\psi(x,u)$, variation of $p(t)$ is also limited w.r.t. t as $x(t)$ can not vary arbitrary fast in terms of (1a). Hence, it is often assumed that variation of $p(t)$ is also restricted to a compact set, *i.e.*, $\mathrm{q}p(t) \in \mathbb{V} \subset \mathbb{R}^{n_\mathrm{p}}$. Based on these consideration, the *admissible set* of trajectories of p can be expressed in the form of

$$p \in \mathscr{P} := \left\{ p \in (\mathbb{R}^{n_\mathrm{p}})^{\mathbb{T}} \mid p(t) \in \mathbb{P},\ \mathrm{q}p(t) \in \mathbb{V} \right\} \tag{6}$$

where $\mathbb{P}$ and $\mathbb{V}$ are commonly chosen to be polytopic sets.

During the modelling process, there is also a degree of freedom in the choice of ψ and the resulting dependence of A, B, C, D in terms of (4). In practice, ψ are chosen such that the matrix functions belong to a certain function class: affine, polynomial, rational or point-wise defined maps and look-up table based forms, determining the *complexity* of the resulting LPV model. The commonly chosen affine dependency gives the simplest complexity, greatly simplifying the subsequent analysis and control design, but it results in a complicated ψ with large n_p. Introducing more complexity in the dependence of A, B, C, D on p in terms of polynomial or rational functions results in simplification of ψ and often reduction of n_p, while making connected analysis and control design problem increasingly more difficult to solve. By choosing ψ as just a selection of elements of $[\ x^\top \quad u^\top\ ]$, results in the simplest scheduling map and the least possible source for conservatism, but the resulting A, B, C, D are often only described point-wise or with look-up-tables, requiring the use of gridded LPV analysis and control design (see it later).

Furthermore, the ambiguity of the choice of p in the LPV embedding process, also makes it possible to end up with LPV models that have reduced controllability/stabilizability or observability/detectability. For example, if $\dot{x}_1 = x_1 x_2$ and $\dot{x}_2 = x_1 + u$, then taking $\dot{x}_1 = p x_1$ with $p = x_2$ makes $\dot{x}_1$ uncontrollable. In extreme cases, the resulting LPV form cannot even be defined for desired (x, u) trajectories of the original system. The latter is called, *well-posedness* of the LPV representation and can occur in cases such as conversion of $\dot{x} = x^2 + 1 + u$ to $\dot{x} = px + u$ with $p = x + 1/x$, making the representation of $x(t) = 0$ ill-posed. Hence, checking the possible loss of characteristic properties of the system and well-posedness is an important step of the LPV embedding process (Hoffmann and Werner, 2015).

2.2 Local modelling concept

Alternatively, LPV models of NL / TV systems can be also obtained by following a local modelling approach that results in a different representation concept. Consider again the nonlinear system given in terms of (1). Linearisation of (1) at an arbitrary $(x_*, u_*) \in \mathbb{X} \times \mathbb{U}$ gives:

$$\begin{aligned} \mathrm{q}x(t) &\approx f(x_*, u_*) + a(x_*, u_*)(x(t) - x_*) + b(x_*, u_*)(u(t) - u_*), \\ y(t) &\approx h(x_*, u_*) + c(x_*, u_*)(x(t) - x_*) + d(x_*, u_*)(u(t) - u_*), \end{aligned}$$

where

$$a = \frac{\partial f}{\partial x}, \quad b = \frac{\partial f}{\partial u}, \quad c = \frac{\partial h}{\partial x}, \quad d = \frac{\partial h}{\partial u}.$$

We can consider a family of such linearisations at the entire $\mathbb{X} \times \mathbb{U}$ or a set of points $\mathcal{D}_* = \{(x_*^{(k)}, u_*^{(k)})\}_{k=1}^N \subset \mathbb{X} \times \mathbb{U}$. Then, by rewriting the linearised system in terms of *deviation variables* we get

$$\dot{\tilde{x}}(t) = a(x_*, u_*)\tilde{x}(t) + b(x_*, u_*)\tilde{u}(t) + \tilde{w}, \tag{7a}$$

$$\tilde{y}(t) = c(x_*, u_*)\tilde{x}(t) + d(x_*, u_*)\tilde{u}(t), \tag{7b}$$

where $\tilde{x} = x - x_*$, $\tilde{u} = u - u_*$, $\tilde{y} = y - h(x_*, u_*)$ and the corresponding correction terms are called *trimming values*, while $\tilde{w} = f(x_*, u_*)$ is an *affine term* resulting when (x_*, u_*) is not an equilibrium point, *i.e.*, $f(x_*, u_*) \neq 0$. Then, the concept of local LPV modelling is to introduce the scheduling signal $p = \psi(x, u) : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^{n_\mathrm{p}}$ with ψ being the scheduling map such that

$$A(p) = a(x, u), \quad B(p) = b(x, u), \quad C(p) = c(x, u), \quad D(p) = d(x, u)$$

where the resulting A, B, C, D matrix functions have a chosen level of complexity, *i.e.*, affine, polynomial, rational or gridded/point-wise defined in terms of a look-up table. In this way, p characterises the variation of the local linearisations in view of the operating point and the choice of ψ determines how much of the complexity of this variation is still represented in A, B, C, D, often at the expense of increased scheduling dimension n_p, representing a complexity/conservativeness trade-off, as discussed earlier. If linearisation of (1) is only known at the finite number of points in $\mathcal{D}_*$, then the matrix functions A, B, C, D are obtained by various forms of interpolation or approximative fit-

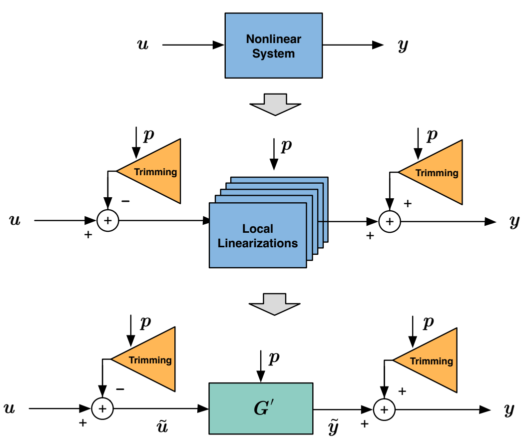


Fig. 2 The local LPV modelling concept.

ting, as will be discussed later. The resulting LPV model is:

$$\begin{bmatrix} \dot{\tilde{x}}(t) \\ \tilde{y}(t) \end{bmatrix} = \begin{bmatrix} A(p(t)) & B(p(t)) \\ C(p(t)) & D(p(t)) \end{bmatrix} \begin{bmatrix} \tilde{x}(t) \\ \tilde{u}(t) \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} \tilde{w}(t) \tag{8}$$

with $\tilde{w}(t)$ representing the variation of the affine terms and $p(t) \in \mathbb{P}$, where $\mathbb{P}$ is often chosen to be a convex set such that

$$\psi(\mathbb{X}, \mathbb{U}) \subseteq \mathbb{P} \quad \text{or} \quad \{\psi(x_*^{(k)}, u_*^{(k)})\}_{k=1}^N \subseteq \mathbb{P}.$$

By seeing p as an external variable that can be calculated or measured based on the current operation of the system, (8) captures the variation of the local dynamics in a continuous sense over $\mathbb{X} \times \mathbb{U}$, providing an approximative, but linear representation of (1). The obtained linear representation can then be used to design controllers, *e.g.*, in the form of $\tilde{u} = K(p)\tilde{x}$, where the gains/state-matrices of the controller vary with p, representing change of the local control law as the system progresses over its operating domain. To be able to represent the original input-output relation of the NL system (1), the trimming terms are interpolated and added back to the deviation inputs and outputs of the LPV model (8) in terms of $u = \tilde{u} + v_{\mathrm{u}}(p)$ and $y = \tilde{y} + v_{\mathrm{y}}(p)$ with $v_{\mathrm{u}}(\psi(x_*, u_*)) = u_*$ and $v_{\mathrm{y}}(\psi(x_*, u_*)) = h(x_*, u_*)$. The resulting overall modelling concept is visualised in Figure 2.

Additionally, to ensure linearity of (8), $\tilde{w}(t)$ is often treated as a disturbance[1] to-be-rejected by control design or absorbed by further trimming in terms of $\tilde{u}(t) - \check{u}_*$ such that $b(x_*, u_*)(\check{u}_* - u_*) = f(x_*, u_*)$, although the latter is often only possible in an approximative sense. Due to the fact that if (x_*, u_*) is an equilibrium point, then $\tilde{w} = f(x_*, u_*) = 0$, in the past, often only an equilibrium family $\mathcal{D}_* = \{(x_*^{(k)}, u_*^{(k)})\}_{k=1}^N$ or manifold was used in connection with local LPV modelling.

By applying the local modelling concept on the unbalanced disc system (2) with the equilibrium manifold

$$\mathcal{D}_* = (x_*, u_*) = (x_{1,*}0, \tfrac{mgl\tau}{J\kappa_{\mathrm{m}}} \sin(x_{1,*})),$$

and choosing affine dependency for $A, \dots, D$, the resulting scheduling is $p(t) = \psi(x(t), u(t)) = \cos(x_1(t))$ and the local LPV model is:

$$\dot{\tilde{x}}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{mgl}{J} p(t) & -\frac{1}{\tau} \end{bmatrix} \tilde{x}(t) + \begin{bmatrix} 0 \\ \frac{\kappa_{\mathrm{m}}}{\tau} \end{bmatrix} \tilde{u}(t), \tag{9a}$$

$$\tilde{y}(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \tilde{x}(t), \tag{9b}$$

with $p(t) \in \mathbb{P} = [-1, 1]$. The associated trimming is computed as $y_* = \mathrm{acos}(p)$ and $u_* = \frac{mgl\tau}{J\kappa_{\mathrm{m}}} \sqrt{1 - p^2}$.

Compared to LPV models obtained via the global embedding approach, the local methodology aims at only approximating the solution set $\mathfrak{B}$ of (1). On the other hand, the involved issues related to conservativeness and complexity of the resulting models are shared and can be handled with similar approaches. Well-posedness of the obtained models is often less of an issue, and difficulties mainly arise in case of order drop or loss of controllability/observability of the system at local linearisation points. A significant advantage of the local approach is that linearisations or local LTI identification of the NL system are commonly available in practice, and, by interpolating them, it is relatively easy to assemble LPV models. Although, there are many pitfalls involved in the interpolation process, and the resulting LPV models are only capable to represent the system trajectories for sufficiently slow variation of p (Shamma and Athans, 1992). Nevertheless, local LPV modelling has been successfully used in many industrial segments and it is supported by extensive software tooling, *e.g.*, in Matlab.

2.3 Forms of LPV representations

Beyond the discussed LPV state-space representations (3), LPV systems can be also represented in terms of various equivalent representation forms, just like in the LTI case. Such an important alternative form of the state-space representation is the so-called *linear fractional representation* (LFR), defined as

$$\mathrm{q}x(t) = Ax(t) + B_{\mathrm{d}} w_\Delta(t) + B_{\mathrm{u}} u(t), \tag{10a}$$

$$z_\Delta(t) = C_{\mathrm{d}} x(t) + D_{\mathrm{dd}} w_\Delta(t) + D_{\mathrm{du}} u(t), \tag{10b}$$

$$y(t) = C_{\mathrm{y}} x(t) + D_{\mathrm{yd}} w_\Delta(t) + D_{\mathrm{yu}} u(t), \tag{10c}$$

$$w_\Delta(t) = \Delta(p(t))\, z_\Delta, \tag{10d}$$

where $w_\Delta(t) \in \mathbb{R}^{n_{\mathrm{w}}}$ and $z_\Delta(t) \in \mathbb{R}^{n_{\mathrm{z}}}$ are latent variables, $\Delta : \mathbb{P} \to \mathbb{R}^{n_{\mathrm{w}} \times n_{\mathrm{z}}}$ is a matrix depending linearly on p (often containing only repeated elements of p in its diagonal) and $A, \dots, D_{\mathrm{yu}}$ are real matrices with appropriate dimensions. The main advantage of LFRs is that they separate the representation of the LPV form to a LTI dynamical system defined by (10a)-(10c) interconnected with a scheduling-dependent varying gain (10d). Just like in robust-control, this reformulation is highly advantageous for analysis and control design, while it can express a wide range of dependencies of (3) from affine to rational, under the well-posedness condition that $I - D_{\mathrm{dd}}\Delta(\mathrm{p})$ is nonsingular for all $\mathrm{p} \in \mathbb{P}$ (Zhou et al., 1995). LPV systems can be also represented in terms of *input-output* (IO) representations:

$$y(t) + \sum_{i=1}^{n_{\mathrm{a}}} a(p(t))\mathrm{q}^i y(t) = \sum_{j=0}^{n_{\mathrm{b}}} a(p(t))\mathrm{q}^j u(t), \tag{11}$$

[1] To ensure better rejection of $\tilde{w}(t)$, the corresponding variation of the affine term can be introduced in the system description in terms of $\tilde{w}(t) = B_{\mathrm{w}}(p(t))w(t)$, where $B_{\mathrm{w}}(p) = f(x, u)$ under a chosen ψ scheduling map and $w(t) \in \mathbb{R}$ with $\|w(t)\| \leq 1$ is a bounded disturbance.

with $n_\mathrm{a} \geq n_\mathrm{b}$, corresponding to an n_a-th order *differential equation* in CT, and an n_a-th order *difference equation* in DT. Especially in discrete-time, IO representations written in q^{-1}, *i.e.*, $\mathrm{q}^{-1}x(t) = x(t-1)$ corresponding to the backward time-shift operator, have been used extensively, as the resulting *filter* forms give rise to similar model structures used in LTI *prediction-error-minimization* (PEM) methods. As a further extension of IO-representations, *infinite impulse-response* representations of LPV systems is also available in DT and CT.

An important caveat of using various form of LPV representations that the involved transformations between them is highly complicated and often requires the introduction of *dynamic scheduling dependency*, which means that equivalent state-space and IO forms might have their coefficients not only dependent on the instantaneous value of $p(t)$, but also on its derivatives $\dot{p}, \ddot{p}, \ldots$ in CT or its time-shifted values $\ldots, p(t+1), p(t), p(t-1), \ldots$ (Tóth, 2010). This is due to fact that, for example in CT, $\frac{\mathrm{d}}{\mathrm{d}t}a(p(t)) \neq a(p(t))\frac{\mathrm{d}}{\mathrm{d}t}$, but $\frac{\partial a}{\partial p}(p(t))\dot{p}(t) + a(p(t))\frac{\mathrm{d}}{\mathrm{d}t}$. For the exact details, and how to avoid the undesired occurrence of dynamic dependency in practice see Tóth et al. (2012a), while for transformation between CT and DT representations see Tóth et al. (2010).

2.4 Model conversion methods

One important question in *global* LPV modelling of NL dynamical systems is that under which condition (1) can be represented in the form of (3) and how existing NL models can be automatically converted to such forms. A rather general answer to this problem is based on the observation that if given a continuously differentiable function $g : \mathbb{R}^n \to \mathbb{R}^m$, then based on the *Fundamental Theorem of Calculus* (FTC):

$$g(\eta) - g(0) = \left(\int_0^1 \frac{\partial g}{\partial \eta}(\lambda\eta)\,\mathrm{d}\lambda \right) \eta, \tag{12}$$

where $\frac{\partial g}{\partial \eta}(\lambda\eta) \in \mathbb{R}^{m\times n}$ is the Jacobian of g evaluated in $\lambda\eta$. Provided that the functions f and h in (1) are differentiable and $f(0,0) = 0$ and $h(0,0) = 0$, which can be achieved by an appropriate coordinate transformation, choosing $\eta = [\ x^\top \ \ u^\top\ ]^\top$ gives via the FTC that

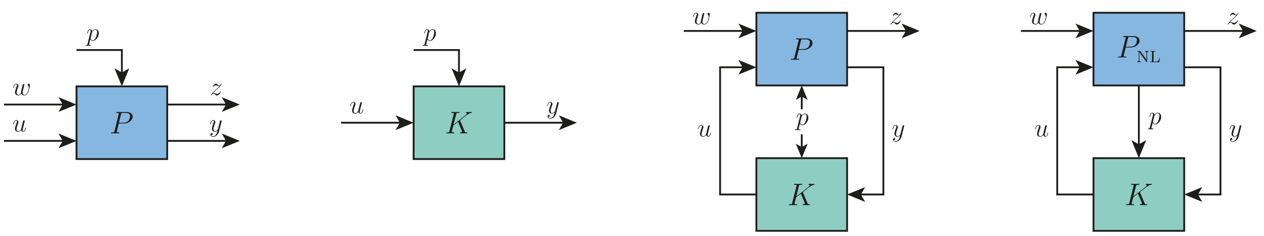
$$f(x,u) = \underbrace{\left(\int_0^1 \frac{\partial f}{\partial x}(\lambda x, \lambda u)\,\mathrm{d}\lambda \right)}_{A(p)} x + \underbrace{\left(\int_0^1 \frac{\partial f}{\partial u}(\lambda x, \lambda u)\,\mathrm{d}\lambda \right)}_{B(p)} u, \tag{13a}$$

$$h(x,u) = \underbrace{\left(\int_0^1 \frac{\partial h}{\partial x}(\lambda x, \lambda u)\,\mathrm{d}\lambda \right)}_{C(p)} x + \underbrace{\left(\int_0^1 \frac{\partial h}{\partial u}(\lambda x, \lambda u)\,\mathrm{d}\lambda \right)}_{D(p)} u. \tag{13b}$$

With $p = [\ x^\top \ \ u^\top\ ]^\top$ and $\psi(x,u)$ the identity function, this gives a direct conversion of (1) to an LPV form. Naturally, by restricting $A, \ldots, D$ to various complexity levels, a spectrum of LPV forms of (1) can also be obtained via (13). While the FTC provides an automated process of conversion, its disadvantage is that potentially p can depend on the entire x and often the integrals in (13) are too complex for an analytic solution. Hence, often numerical integration is required, giving only room for a gridded representation of the dependency.

Existing alternative approaches for *global* LPV modelling of NL dynamical systems can be classified into three main categories: *substitution based transformation* (SBT) methods (Papageorgiou et al., 2000; Rugh and Shamma, 2000; Leith and Leithhead, 1998; Marcos and Balas, 2004), *automated factorization procedures* (Donida et al., 2009; Kwiatkowski et al., 2006; Tóth, 2010; Hoffmann and Werner, 2015), and *data-driven methods* (Sadeghzadeh et al., 2020; Kwiatkowski and Werner, 2008; Koelewijn and Tóth, 2020). SBT methods are often based on various application-specific substitution of nonlinear terms with scheduling variables, or using state-transformations that can factorize nonlinear terms to a scheduling dependent linear signal relation. These methods often do not pay serious attention to several issues regarding the resulting LPV models, namely: how the scheduling variable and its bounds are chosen, what is the relation between these choices and the behaviour of the system, and the usefulness of the resulting LPV form for control synthesis or as a source of model structure information for identification. On the other hand, automated factorization methods are based on decision tree algorithms or optimization of the scheduling relation by taking the above mentioned concepts into account, resulting in computationally heavy methods. The data-driven methods focus on approximate conversion by aiming at the minimization of the dimension of p and the number of vertices and volume of $\mathbb{P}$.

Regarding *local* LPV modelling of NL dynamical systems, approaches are mainly distinguished in terms of the way how the local linearisations of the system are interpolated, and the type of linearisation that is used to obtain the local aspects. Regarding the latter, early methods used only linearisation around a set of equilibrium points, which was later extended to off-equilibrium and trajectory-based linearisations of NL systems. While conceptually the resulting representation of the NL behaviour is different, these approaches only differ in the need of handling affine terms in the state equation (7). In terms of interpolation, we distinguish approaches that only blend together the outputs of the local linear methods in terms of scheduling dependent weights, which is called *output interpolation*, similarly there exist *input interpolation* methods, and the most common form is *coefficient interpolation*, *i.e.*, interpolation of the local values of $A, \ldots, D$ by using piece-wise affine, polynomial, rational, etc. functions or polytopic embeddings, see Bachnas et al. (2014) for an overview. While the latter methods correspond to a rather intuitive approach, they are ill-posed for state-space realisation when the local state-space matrices are not defined on the same global state-basis. This is often the case when the local descriptions of the system are identified from data, as each locally estimated model is obtained for a random state-basis, and it is not possible from local information to obtain the global state-basis on which the interpolation can be meaningfully accomplished (Zhang and Ljung, 2017). Due to this reason, the concept of *behavioural interpolation* methods have been introduced, where a global parametrization of (8) is taken, *e.g.*, in terms of affine dependence on the operating condition, and the parameters are chosen such that the $\mathcal{H}_\infty$ or $\mathcal{H}_2$ error of the frozen transfer function of the LPV model for each observed local model of the NL system is minimized (Petersson and Löfberg, 2009; Vizer et al., 2014). At the expense of computational complexity, the latter methods provide the most reliable way of obtaining LPV models in practice via the local approach.



(a) LPV plant. (b) LPV controller. (c) LPV closed-loop interconnection. (d) Closed-loop operation with the NL plant.

Fig. 3 The concept of LPV control interconnection and implementation.

2.5 Data-driven modelling and system identification

Obtaining global LPV models of NL systems from data sets with varying p has been studied extensively, resulting in the development of a wide range of methods: *prediction-error-minimization* (PEM) based estimation of LPV-IO model structures such as ARX, ARMAX, OE and BJ (Tóth et al., 2012b) together with LPV-SS models under innovation noise, extensive set of *subspace* methods (van Wingerden and Verhaegen, 2009; Cox and Tóth, 2021), *instrumental variable* methods (Laurain et al., 2010), etc., together with various approaches of models structure selection and machine-learning-based estimation of the scheduling dependencies. Current approaches aim at co-estimating the scheduling map ψ with the LPV relations, discovering which elements of (x,u) are essentially required for ψ to obtain accurate LPV models of the NL system (Verhoek et al., 2022a). Data-driven approaches of estimating LPV models through the local paradigm does not differ from the model conversion methods, as after estimation of the local models by LTI approaches, the only remaining step is their interpolation to a global model. Recent methods also try to combine global data-sets with local information, resulting in so-called *glocal* methods. A wide range of the above mentioned methods are implemented in the software toolbox LPVcore (den Boef et al., 2021).

3 Control of LPV systems

Due to the extensive work on the extension of LTI methods to the parameter-varying case, control design for the obtained LPV models is available in many different forms: observer design, state-feedback, dynamic output-feedback, model-predictive control, etc. under various objective functions and using model-based and lately even model-free paradigms. In the remaining part of this chapter, we will mainly focus on the model-based output-feedback LPV control design problem as both observer and state-feedback controller design are essentially special cases of dynamic output-feedback control and because early forms of these approaches were responsible for the success and wide-spreading of the LPV framework.

3.1 The LPV feedback control design problem

The two main objectives of control design is guaranteeing stability and performance of the controlled closed-loop system. In optimal-gain control, performance is measured in terms of a signal norm relationship between general disturbances $w(t)$ such as reference signals, input loads, noise, etc., and performance signals $z(t)$ such as tracking error, actuator inputs, etc., of the closed-loop behaviour, while the controller operates the system via the control inputs $u(t)$ and obtains information about it via the measurement signals $y(t)$. An example of such an interconnection is given in Figure 4. A wide range of various LPV control and observer design problems can be expressed in the form of *generalized plants*, corresponding to the well-known concept of the LTI robust control framework (Zhou et al., 1995) that allows for systematic handling of general design problems and efficient performance shaping.

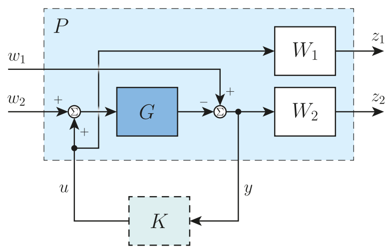


Fig. 4 Example of control interconnection corresponding to a generalized plant.

Consider the LPV model G of the system and the corresponding control interconnection represented as a generalized plant P in the form of

$$P: \left\{ \begin{array}{rcl} \mathrm{q}x(t) &=& A(p(t))x(t) + B_{\mathrm{w}}(p(t))w(t) + B_{\mathrm{u}}(p(t))u(t), \\ z(t) &=& C_{\mathrm{z}}(p(t))x(t) + D_{\mathrm{zw}}(p(t))w(t) + D_{\mathrm{zu}}(p(t))u(t), \\ y(t) &=& C_{\mathrm{y}}(p(t))x(t) + D_{\mathrm{yw}}(p(t))w(t) + D_{\mathrm{yu}}(p(t))u(t), \end{array} \right. \tag{14}$$

where $A,\ldots,D_{\mathrm{yu}}$ are matrix functions of appropriate dimensions and $p \in \mathscr{P}$ with $\mathscr{P}$ being an admissible scheduling set defined as in (6). The controller that we aim to design for the plant P is considered with a similar structure:

$$K: \left\{ \begin{array}{rcl} \mathrm{q}x_{\mathrm{K}}(t) &=& A_{\mathrm{K}}(p(t))x_{\mathrm{K}}(t) + B_{\mathrm{K}}(p(t))y(t), \\ u(t) &=& C_{\mathrm{K}}(p(t))x_{\mathrm{K}}(t) + D_{\mathrm{K}}(p(t))y(t), \end{array} \right. \tag{15}$$

where $x_{\mathrm{K}}(t) \in \mathbb{R}^{n_{x_{\mathrm{K}}}}$ and $A_{\mathrm{K}},\ldots,D_{\mathrm{K}}$ are p-dependent matrix functions with a given class of scheduling dependency, *e.g.*, affine, polynomial, etc. Then, P is considered to be a generalized plant, if there exists a K such that all signal relations in the resulting closed-loop, illustrated by Figure 3.c, are stable (internal stability). This property is equivalent with stabilizability and detectability of P.

By linearity of LPV systems, the closed-loop interconnection is linear in the signals w and z and the state $x_{\mathrm{cl}} := [\, x^\top \; x_{\mathrm{K}}^\top \,]^\top$. Hence, under the well-posedness condition that

$$\begin{bmatrix} I & -D_{\mathrm{K}}(\mathtt{p}) \\ -D_{\mathrm{yu}}(\mathtt{p}) & I \end{bmatrix}^{-1} \text{ exists } \forall \mathtt{p} \in \mathbb{P},$$

where satisfaction of the condition for all $\mathtt{p}$ constant vectors in $\mathbb{P}$ is equivalent with satisfying it along all possible $p \in \mathscr{P}$ trajectories, the closed-loop can be represented by the state-space realisation

$$P \star K : \begin{cases} \mathrm{q}x_{\mathrm{cl}}(t) &= \mathcal{A}(p(t))x_{\mathrm{cl}}(t) + \mathcal{B}(p(t))w(t), \\ z(t) &= \mathcal{C}(p(t))x_{\mathrm{cl}}(t) + \mathcal{D}(p(t))w(t). \end{cases} \tag{16}$$

Stability of the resulting closed-loop behaviour can be considered in various forms, but most commonly it is characterised in terms of the existence of a scalar quadratic Lyapunov function of the form[2]

$$V(x_{\mathrm{cl}}, p) = x_{\mathrm{cl}}^\top X(p) x_{\mathrm{cl}}, \qquad \exists \alpha > 0 \text{ such that } 0 \prec X(p) \preccurlyeq \alpha I, \;\; \forall p \in \mathscr{P},$$

that satisfies that $\frac{\mathrm{d}}{\mathrm{d}t} V(x_{\mathrm{cl}}, p)$ in CT and $\mathrm{q}V(x_{\mathrm{cl}}, p) - V(x_{\mathrm{cl}}, p)$ in DT is a negative definite function in x_{cl} along all $p \in \mathscr{P}$. Existence of such a $X : \mathbb{P} \to \mathbb{S}^{n_{\mathrm{x}}+n_{\mathrm{x_k}}}$, where $\mathbb{S}^n$ stands for symmetric real matrices of dimension n, implies asymptotic stability of all signal relations in Fig. 3.c. To search for $X(p)$, often parametrization of its dependency over p is considered in an affine form or it is simplified to be just a constant matrix.

Beyond stability, in control design it is also important to characterise the desired closed-loop behaviour and quantify the performance of the controller in achieving it. This quantification of performance in LPV control is often considered via the concept of dissipativity with quadratic supply functions, *i.e.*, performance is evaluated through the existence of a quadratic storage function $V(x_{\mathrm{cl}}, p)$ such that

$$\text{CT:} \qquad V(x_{\mathrm{cl}}(t_1), p(t_1)) - V(x_{\mathrm{cl}}(t_0), p(t_0)) \leq \int_{t_0}^{t_1} s(w(t), z(t))\mathrm{d}t, \tag{17a}$$

$$\text{DT:} \qquad V(x_{\mathrm{cl}}(t_1+1), p(t_1+1)) - V(x_{\mathrm{cl}}(t_0), p(t_0)) \leq \sum_{t=t_0}^{t_1} s(w(t), z(t)), \tag{17b}$$

where $s(w(t), z(t)) = \begin{bmatrix} w(t) \\ z(t) \end{bmatrix}^\top \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} w(t) \\ z(t) \end{bmatrix}$ is the supply function with $Q \in \mathbb{S}^{n_{\mathrm{w}}}$, $S \in \mathbb{R}^{n_{\mathrm{w}} \times n_{\mathrm{z}}}$ and $R \in \mathbb{S}^{n_{\mathrm{z}}}$. Just like in classical robust control, through $s(\cdot, \cdot)$ we can characterise various well-known performance notions[3], such as $\mathcal{L}_2$, passivity, $\mathcal{L}_\infty$ and generalized $\mathcal{H}_2$ (Scherer and Weiland, 2021). By the introduction of frequency-domain weighting filters in the control interconnection, which constitutes to P, characteristics of the expected closed-loop behaviour can be specified in the relation between w and z. Then, via (17), it can be evaluated that how well a given LPV controller K achieves these specifications. This allows to use popular techniques of *mixed-sensitivity*, *loop-shaping*, *signal-based* and *model reference* techniques for analyzing or shaping the closed-loop behaviour (Skogestad and Postlethwaite, 1996). Furthermore, dissipativity for supply functions with $R \prec 0$ also implies that $V(x_{\mathrm{cl}}, p)$ qualifies as a Lyapunov function, implying asymptotic stability of the closed-loop system behaviour. Under these consideration the LPV control *design* problem at a general level can be formulated as follows:

Problem. Given P, design K such that for all $p \in \mathscr{P}$, the map $P \star K : (\mathbb{R}^{n_{\mathrm{w}}})^{\mathbb{T}} \to (\mathbb{R}^{n_{\mathrm{z}}})^{\mathbb{T}}$ is internally asymptotically stable and dissipative w.r.t. a user-defined performance criterion in terms of the supply function $s(\cdot, \cdot)$. □

Similarly, in case of a given LPV controller K, *analyzing* its stability and achieved performance corresponds to testing its dissipativity under a given s. Note that, many times, the achieved performance level, *e.g.*, the $\mathcal{L}_2$ gain, can be minimized in an optimization-based design process for K, which is called optimal-gain *synthesis*. Furthermore, for a global LPV embedding of a nonlinear system based on which P is constructed, the guarantees on stability and performance of the LPV embedding carry over to the nonlinear system when the designed controller is attached to it, which makes LPV control a highly attractive control design approach. For further details on the exact form of guarantees, conditions and how possible problems can be avoided, see Koelewijn (2023).

3.2 Solving the LPV control problem

There is a plethora of research results and a wide range of methods available in the literature on the solution of the LPV control problem. Here, we will give a brief overview of the main results for three specific scheduling dependency structures of the plant P:

- Affine/linear scheduling dependence, resulting in a polytopic representation of P,
- Rational scheduling dependence, resulting in a LFR form of P,

---

[2] The standard notation $\prec$ and $\succ$ ($\preccurlyeq$ and $\succcurlyeq$) stands for negative and positive (semi-)definiteness of matrices, respectively.

[3] The $\mathcal{H}_\infty$-norm and the $\mathcal{H}_2$-norm are only defined for transfer functions of LTI systems. Their generalization to LPV systems is achieved through their induced norm, *i.e.*, gain, interpretation (the $\mathcal{L}_2$-gain and the generalized $\mathcal{H}_2$-norm).

- Gridded scheduling dependence, resulting in a point-wise representation of P.

3.2.1 General form of the solution

The LPV controller synthesis problem for CT systems can be expressed as a feasibility condition of an (infinite) set of matrix inequalities, which can be found in multiple forms throughout the literature, *e.g.*, (Apkarian and Adams, 2000; Scherer and Weiland, 2021; De Caigny et al., 2012; Briat, 2016). Conditions for the DT case follow a similar formulation. For the sake of simplicity, we take the assumption that $D_{\mathrm{yu}} = 0$ in (14) to avoid problems of well-posedness during synthesis.

Theorem 1. *There exists a controller K given by* (15) *such that the closed-loop interconnection of a generalized plant P given by* (14) *and K is (Q, S, R) dissipative, if there exist a positive-define matrix $M_{\mathrm{x}} \in \mathbb{S}^{n_{\mathrm{x}}}$, positive-definite matrix function $M_{\mathrm{y}} : \mathbb{P} \to \mathbb{S}^{n_{\mathrm{x}}}$ with $M_{\mathrm{y}} \in C_1$, and (transformed controller) matrix functions $\mathcal{A}_{\mathrm{K}} : \mathbb{P} \to \mathbb{R}^{n_{\mathrm{x}}\times n_{\mathrm{x}}}$, $\mathcal{B}_{\mathrm{K}} : \mathbb{P} \to \mathbb{R}^{n_{\mathrm{x}}\times n_{\mathrm{y}}}$, $C_{\mathrm{K}} : \mathbb{P} \to \mathbb{R}^{n_{\mathrm{u}}\times n_{\mathrm{x}}}$, and $\mathcal{D}_{\mathrm{K}} : \mathbb{P} \to \mathbb{R}^{n_{\mathrm{u}}\times n_{\mathrm{y}}}$, such that for all $(\mathsf{p}, \mathsf{v}) \in \mathbb{P} \times \mathbb{V}$, it holds that*

$$\begin{bmatrix} \mathcal{A}_{\mathrm{cl}}(\mathsf{p},\mathsf{v}) + (\mathcal{A}_{\mathrm{cl}}(\mathsf{p},\mathsf{v}))^\top & \mathcal{B}_{\mathrm{cl}}(\mathsf{p}) \\ (\mathcal{B}_{\mathrm{cl}}(\mathsf{p}))^\top & 0 \end{bmatrix} - \begin{bmatrix} 0 & I \\ \mathcal{C}_{\mathrm{cl}}(\mathsf{p}) & \mathcal{D}_{\mathrm{cl}}(\mathsf{p}) \end{bmatrix}^\top \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} 0 & I \\ \mathcal{C}_{\mathrm{cl}}(\mathsf{p}) & \mathcal{D}_{\mathrm{cl}}(\mathsf{p}) \end{bmatrix} \preceq 0, \qquad \begin{bmatrix} M_{\mathrm{y}}(\mathsf{p}) & I \\ I & M_{\mathrm{x}} \end{bmatrix} \succ 0, \tag{18}$$

where the closed-loop related matrices are

$$\mathcal{A}_{\mathrm{cl}}(\mathsf{p},\mathsf{v}) = \begin{bmatrix} A(\mathsf{p})M_{\mathrm{y}}(\mathsf{p}) + B_{\mathrm{u}}(\mathsf{p})\mathcal{C}_{\mathrm{K}}(\mathsf{p}) & A(\mathsf{p}) + B_{\mathrm{u}}(\mathsf{p})\mathcal{D}_{\mathrm{K}}(\mathsf{p})C_{\mathrm{y}}(\mathsf{p}) \\ \mathcal{A}_{\mathrm{K}}(\mathsf{p}) & M_{\mathrm{x}}A(\mathsf{p}) + \mathcal{B}_{\mathrm{K}}(\mathsf{p})C_{\mathrm{y}}(\mathsf{p}) \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}\partial M_{\mathrm{y}}(\mathsf{p},\mathsf{v}) & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathcal{B}_{\mathrm{cl}}(\mathsf{p}) = \begin{bmatrix} B_{\mathrm{w}}(\mathsf{p}) + B_{\mathrm{u}}(\mathsf{p})D_{\mathrm{K}}(\mathsf{p})D_{\mathrm{yw}}(\mathsf{p}) \\ M_{\mathrm{x}}B_{\mathrm{w}}(\mathsf{p}) + \mathcal{B}_{\mathrm{K}}(\mathsf{p})B_{\mathrm{yw}}(\mathsf{p}) \end{bmatrix},$$

$$\mathcal{C}_{\mathrm{cl}}(\mathsf{p}) = \begin{bmatrix} C_{\mathrm{z}}(\mathsf{p})M_{\mathrm{y}}(\mathsf{p}) + D_{\mathrm{zu}}(\mathsf{p})\mathcal{C}_{\mathrm{K}}(\mathsf{p}) & C_{\mathrm{z}}(\mathsf{p}) + D_{\mathrm{zu}}(\mathsf{p})\mathcal{D}_{\mathrm{K}}(\mathsf{p})C_{\mathrm{y}}(\mathsf{p}) \end{bmatrix}, \qquad \mathcal{D}_{\mathrm{cl}}(\mathsf{p}) = D_{\mathrm{zw}}(\mathsf{p}) + D_{\mathrm{zu}}(\mathsf{p})\mathcal{D}_{\mathrm{K}}(\mathsf{p})D_{\mathrm{yw}}(\mathsf{p}),$$

where $\partial M_{\mathrm{y}}(\mathsf{p},\mathsf{v}) = \sum_{i=1}^{n_{\mathrm{p}}} \frac{\partial M_{\mathrm{y}}(\mathsf{p})}{\partial \mathsf{p}_i}\mathsf{v}_i$. The matrix functions of the controller K then are given by

$$\begin{bmatrix} A_{\mathrm{K}}(\mathsf{p}) & B_{\mathrm{K}}(\mathsf{p}) \\ C_{\mathrm{K}}(\mathsf{p}) & D_{\mathrm{K}}(\mathsf{p}) \end{bmatrix} = \begin{bmatrix} U & M_{\mathrm{x}}B_{\mathrm{u}}(\mathsf{p}) \\ 0 & I \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathcal{A}_{\mathrm{K}}(\mathsf{p}) - M_{\mathrm{x}}A(\mathsf{p})M_{\mathrm{y}}(\mathsf{p}) & \mathcal{B}_{\mathrm{K}}(\mathsf{p}) \\ \mathcal{C}_{\mathrm{K}}(\mathsf{p}) & \mathcal{D}_{\mathrm{K}}(\mathsf{p}) \end{bmatrix} \right) \begin{bmatrix} V(\mathsf{p})^\top & 0 \\ C_{\mathrm{y}}(\mathsf{p})M_{\mathrm{y}}(\mathsf{p}) & I \end{bmatrix}^{-1}, \tag{19}$$

where U and V are arbitrary solutions to $M_{\mathrm{x}}M_{\mathrm{y}}(\mathsf{p}) + UV(\mathsf{p})^\top = I$.

For a proof of Theorem 1 and its simplification to state-feedback design, see e.g. (Koelewijn, 2023). To solve the controller design problem based on (18) and (19), we need to satisfy (18) for all values p and v in the value sets that define $\mathscr{P}$, *i.e.*, $\mathbb{P}$ and $\mathbb{V}$. This corresponds to a feasibility problem with infinitely many constraints. Furthermore, parametrizations of $\mathcal{A}_{\mathrm{K}}, \ldots \mathcal{D}_{\mathrm{K}}$ are required to be introduced to convert (18) to *linear matrix inequality* (LMI) constraints. In order to make this problem tractable, we will consider specific dependency structures of our plant and controller.

3.2.2 Polytopic controller synthesis

The LPV controller synthesis problem becomes tractable when the plant P can be represented as a *polytopic system*, which is the case when

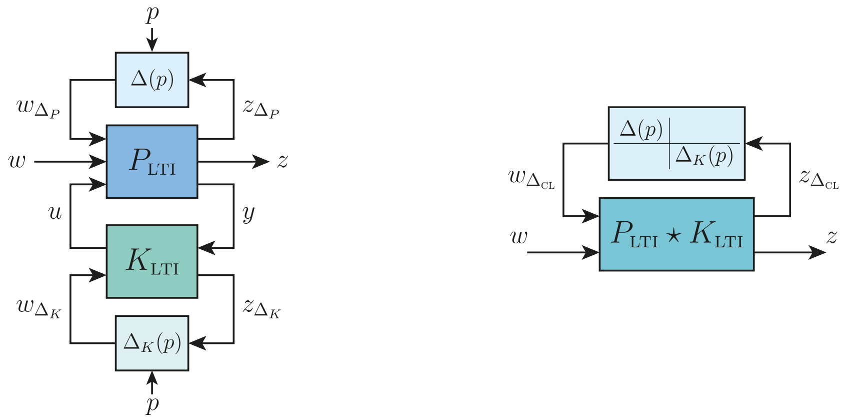
- $A, \ldots, D_{\mathrm{yw}}$ in (14) have only affine dependence on p, *i.e.*, $A(p(t)) = \sum_{i=0}^{n_{\mathrm{p}}} p_i(t)A_i$, with $p_0(t) = 1$ for all t, and $A_i \in \mathbb{R}^{n_{\mathrm{x}}\times n_{\mathrm{x}}}$.
- $\mathbb{P} \subset \mathbb{R}^{n_{\mathrm{p}}}$ and $\mathbb{V} \subset \mathbb{R}^{n_{\mathrm{p}}}$ are convex polytopes with $n_{\mathbb{P}}$ and $n_{\mathbb{V}}$ vertices, respectively, *i.e.*, $\mathbb{P} := \mathrm{cohull}\left(\{\mathsf{p}^{\mathrm{v}_i}\}_{i=1}^{n_{\mathbb{P}}}\right)$, and $\mathbb{V} := \mathrm{cohull}\left(\{\mathsf{v}^{\mathrm{v}_i}\}_{i=1}^{n_{\mathbb{V}}}\right)$.

Under these properties, the plant dynamics for any $p(t) \in \mathbb{P}$ can be represented as a convex combination of the dynamics corresponding to the individual vertices of $\mathbb{P}$. Moreover, by choosing the scheduling dependence of the controller to affine and assuming that $B_{\mathrm{u}}, D_{\mathrm{zu}}, C_{\mathrm{y}}$ and D_{yw} are not dependent on $p(t)$, which can be guaranteed via simple filters, the closed-loop system representation $P \star K$ is also affine in $p(t)$. Taking M_{y} to be a constant matrix, makes the synthesis problem convex in p, independent of v, and linear in the decision variables. Hence, using convex-hull relaxation, (18) becomes an LMI and solving it on the vertices of $\mathbb{P}$ is equivalent with satisfying (18) for all $\mathsf{p} \in \mathbb{P}$. Alternatively, one can also consider scheduling dependent M_{y} or alternative control synthesis solutions with p-dependent storage functions to end up with a set of LMIs depending on both p and v, which are required to be satisfied only at the vertices of $\mathbb{P} \times \mathbb{V}$ to provide a solution to LPV control design problem. The resulting feasibility problems can be formulated and solved as a *semidefinite program* (SDP), for which many tools exist, *e.g.*, YALMIP or CVX. In fact, several Matlab toolboxes are available in which synthesis algorithms for multiple performance objectives are implemented in this fashion, *e.g.*, LPVcore (den Boef et al., 2021), LPVTools (Hjartarson et al., 2015), or IQCLab (Veenman et al., 2021).

The synthesis problem returns a set of LTI controllers corresponding to the generators of $\mathbb{P}$. To use the corresponding LPV controller in real-time, one will need the values of the scheduling-dependent matrix functions of K for a measured $p(t)$ to compute the control input $u(t)$. This problem can be handled in an online or offline setting. The online approach computes $A_{\mathrm{K}}(p(t)), \ldots, D_{\mathrm{K}}(p(t))$ for a measured $p(t)$ every time-step by solving a linear program that finds a $\lambda(t) \in \mathbb{R}^{n_{\mathbb{P}}}$ subject to

$$p(t) = \sum_{i=1}^{n_{\mathbb{P}}} \lambda_i(t)\mathsf{p}^{\mathrm{v}_i}, \quad \lambda_i(t) \geq 0, \quad \sum_{i=1}^{n_{\mathbb{P}}} \lambda_i(t) = 1, \qquad \text{such that:} \quad \begin{bmatrix} A_{\mathrm{K}}(p(t)) & B_{\mathrm{K}}(p(t)) \\ C_{\mathrm{K}}(p(t)) & D_{\mathrm{K}}(p(t)) \end{bmatrix} = \sum_{i=1}^{n_{\mathbb{P}}} \lambda_i(t) \begin{bmatrix} A_{\mathrm{K}}(\mathsf{p}^{\mathrm{v}_i}) & B_{\mathrm{K}}(\mathsf{p}^{\mathrm{v}_i}) \\ C_{\mathrm{K}}(\mathsf{p}^{\mathrm{v}_i}) & D_{\mathrm{K}}(\mathsf{p}^{\mathrm{v}_i}) \end{bmatrix},$$

where uniqueness can be enforced by regularization, *e.g.*, minimizing $\|\lambda(t)\|_2^2$. Alternatively, the offline approach converts the polytopic realisation of K to an affine one. The coefficient matrices $A_{\mathrm{K},0}, \ldots, D_{\mathrm{K},n_{\mathrm{p}}}$ can be obtained by solving the above linear program for $\mathsf{p} = \{0, [\,1\ 0\ \ldots\ 0\,]^\top, \ldots, [\,0\ \ldots\ 0\ 1\,]^\top\}$.



(a) Closed-loop interconnection in terms of the LFRs of P and K.

(b) Closed-loop interconnection as a combined LFR.

Fig. 5 LFR form of the closed-loop interconnection of the plant P and controller K.

3.2.3 LFR-based controller synthesis

Compared to the polytopic form, a more general case is when the plant and the controller are represented by LFRs, as depicted in Figure 5. In this case, the representations of the plant and controller are separated to LTI dynamical systems interconnected with blocks Δ and Δ_{K} depending linearly on p. Often, Δ and Δ_{K} only contain repeated elements of p in their diagonal. LFRs allow to represent affine, polynomial and rational dependency structures of their equivalent LPV state-space form, covering a wide range of possible LPV embeddings.

To derive a tractable solution to (18) in case of LFR forms, it is an important observation that the closed-loop system can be written as a single LFR, where the 'uncertainty'-block is defined as $\Delta_{\mathrm{CL}}(p) := \mathrm{diag}(\Delta(p), \Delta_{\mathrm{K}}(p))$. This formulation allows to use the well-established robust control synthesis techniques to solve the LPV synthesis problem. Again, the first step is to assume that $\Delta_{\mathrm{CL}}(\mathbb{P}) \subseteq \mathrm{cohull}\left(\{\delta_i\}_{i=1}^{n_\Delta}\right)$, which is often the case if $\mathbb{P}$ is a convex polytope. Secondly, through the full-block S-procedure, a multiplier-based approach is employed to transform the problem in Theorem 1 to an LMI-based feasibility problem. An interesting side-result of the multiplier approach is that the structure of $\Delta_{\mathrm{K}}(p)$ can be freely determined during the synthesis. This extra freedom yields the synthesis inequalities convex, which is not the case in *robust* control synthesis. After solving the LPV synthesis problem, $\Delta_{\mathrm{K}}(p)$ can be explicitly computed from $\Delta(p)$ and the solution to the LMI problem. The resulting LPV controller is represented by an LFR and can be easily implemented. In the literature, several extensions and alternative formulations of this approach exist via the use of *integral quadratic constraints* (IQCs). More details on these approaches can be found in (Scherer and Weiland, 2021; Scherer, 2001).

3.2.4 Gridded controller synthesis

In case the plant dynamics are only defined point-wise over the scheduling set $\mathbb{P}$ or they have highly complex dependency over p, then under a global parametrization of the controller (15), *e.g.*, affine, polynomial, etc., the LPV synthesis problem of Theorem 1 can be solved on a dense grid over $\mathbb{P} \times \mathbb{V}$. The main disadvantage of this method is that the guarantees provided by Theorem 1 are *only* valid on the grid points, and thus not for all $p \in \mathscr{P}$. On the other hand, the main advantages are that *(i)* it is fairly simple to realise, *(ii)* it can handle arbitrary scheduling dependence of $A, \ldots, D_{\mathrm{yw}}$, and *(iii)* it can work with plants that are *defined* on a grid, *e.g.*, a set of local linearizations of a nonlinear system. The corresponding synthesis methods are implemented in LPVcore (den Boef et al., 2021) and LPVTools (Hjartarson et al., 2015).

3.2.5 Further extensions and advanced concepts

While we discussed the most common LPV control synthesis approaches for the dynamic output feedback scenario, there are many extensions, variations and advanced concepts developed throughout the years, *e.g.*, extensions for structured synthesis (Apkarian et al., 2014), extensions for different various types of p-dependent Lyapunov functions (Briat, 2016) and even reformulation of LPV synthesis to provide more general equilibrium-free and incremental stability and performance guarantees for the resulting NL closed-loop system (Koelewijn, 2023). Lately, extensions towards model-free synthesis have been developed in terms of data-driven LPV approaches (Formentin et al., 2016; Verhoek et al., 2022b).

3.3 Gain-scheduling

An early alternative of the considered control synthesis approach of Section 3.2 is gain-scheduling, where using a local LPV model of the nonlinear system, one can design LTI controllers for each of the observed local aspects of the nonlinear system. This corresponds to solving an LTI synthesis problem for frozen, *i.e.*, constant, values of p. Then, the resulting set of designed local controllers are interpolated over the scheduling range $\mathbb{P}$ by means of the same interpolation schemes (input, output, coefficient, behavioural) and interpolation methods (linear, polynomial, spline, look-up-table, etc.) as discussed in Section 2.2 for local modelling.

Although formal guarantees for stability an performance of the closed-loop behaviour cannot be obtained due to the gridded nature of the synthesis and the presence of hidden coupling effects (Rugh and Shamma, 2000), this is the most standard use of LPV control in the industry. There are many different types of gain-scheduling approaches (classical, advanced, velocity gain-scheduling), all with an extensive set of supporting tools in Matlab. Recently, it has been shown that formal stability and performance guarantees can be also obtained via gain-scheduling together with the elimination of the hidden coupling effects using appropriate LPV realisation of the local control laws (Wang et al., 2024).

3.4 Other control design methods

Beyond the above considered control synthesis approaches, a wide-range of extension of many powerful LTI design methods has been developed together with various formulation of LPV MPC methods, ranging from robust predictive control methods (Besselmann et al., 2012) to iterated predictive solutions (Cisneros, 2021), where the latter corresponds to an efficient form of NL MPC. Due to the vast number of methods available, a comprehensive overview of them is out of the scope of this work.

4 Discussion and outlooks

As we have seen, *linear parameter-varying* (LPV) modelling and system identification provide attractive approaches to describe complex nonlinear and/or time-varying systems in terms of simple linear surrogate models that can easily be used for convex stability and performance analysis or synthesis of controllers. Using either global methods, based on the embedding of the dynamic behaviour of the nonlinear systems into the solution set of a linear but varying representation, or the local methods that aim at capturing the system dynamics by the interpolation of local linearisations, the LPV framework provides reliable methods to establish such models in practice, even optimizing their complexity, conservativeness and well-posedness for the task at hand. For the obtained LPV surrogate models, a wide range of analysis and control design methods are available that provide stability and performance guarantees for the surrogate model described system, they are computed via powerful convex optimization methods, and they enable performance shaping and intuitive understanding of the design process, making them a popular choice in aerospace engineering, automotive technologies, and the mechatronic industry.
Currently the most important directions of further development are

- Development of objective-oriented LPV modelling and reduction methods, where the utilization objectives, *e.g.*, control specifications, are taken into account in the construction/identification of the LPV models and optimization of their accuracy, complexity, and conservativeness.
- Further development of LPV control methods for synthesis of structured controllers for large scale systems with possible uncertainties, resulting in global (*e.g.*, shifted, incremental) stability and performance guarantees.
- Merging LPV identification and control synthesis in terms of data-driven control methods to provide an end-to-end automated control design solution that goes beyond the efficiency and guarantees of the current separate modelling and design methods.

References

Apkarian P and Adams RJ (2000), Advanced gain-scheduling techniques for uncertain systems, Advances in linear matrix inequality methods in control, SIAM, 209–228.
Apkarian P and Gahinet P (1995). A convex characterization of gain-scheduled $\mathcal{H}_\infty$ controllers. *IEEE Trans. on Automatic Control* 40: 853–864.
Apkarian P, Gahinet P and Buhr C (2014), Multi-model, multi-objective tuning of fixed-structure controllers, Proc. of the European Control Conference, 856–861.
Bachnas AA, Tóth R, Mesbah A and Ludlage J (2014). A review on data-driven linear parameter-varying modeling approaches: A high-purity distillation column case study. *Journal of Process Control* 24: 272–285.
Baslamisli S, Köse IE and Anlaş G (2009). Gain-scheduled integrated active steering and differential control for vehicle handling improvement. *Vehicle System Dynamics* 47 (1): 99–119.
Belforte G, Dabbene F and Gay P (2005). LPV approximation of distributed parameter systems in environmental modeling. *Environmental Modeling & Software* 20 (8): 1063–1070.
Besselmann T, Löfberg J and Morari M (2012). Explicit MPC for LPV systems: Stability and optimality. *IEEE Trans. on Automatic Control* 57 (9): 2322–2332.
Bianchi FD, De Battista H and Mantz RJ (2007). Wind Turbine Control Systems; Principles, modeling and gain scheduling design, Springer-Verlag.
Briat C (2016). Linear Parameter-Varying and Time-Delay Systems, Advances in Delays and Dynamics, Springer Berlin, Heidelberg.
Broens Y, Butler H and Tóth R (2022), LPV sequential loop closing for high-precision motion systems, Proc. of the American Control Conference, Atlanta, GA, USA, 3178–3183.
Cisneros P (2021). Quasi-Linear Model Predictive Control: Stability, Modelling and Implementation. Ph.D. thesis, Technical University Hamburg.
Cox PB and Tóth R (2021). Linear parameter-varying subspace identification: A unified framework. *Automatica* 123: 109296.
De Caigny J, Camino JF, Oliveira RCLF, Peres PLD and Swevers J (2012). Gain-scheduled dynamic output feedback control for discrete-time LPV systems. *International Journal of Robust and Nonlinear Control* 22 (5): 535–558.
de Lange MH, Verhoek C, Preda V and Tóth R (2022), LPV modeling of the atmospheric flight dynamics of a generic parafoil return vehicle, Proc. of the 5th IFAC Workshop on Linear Parameter-Varying Systems, Montreal, Canada, 238–243.
den Boef P, Cox PB and Tóth R (2021), LPVcore: Matlab toolbox for LPV modelling, identification and control of non-linear systems, Proc. of the 19th Symposium on System Identification, 385–390.

Do AL, Sename O and Dugard L (2010), An LPV control approach for semi-active suspension control with actuator constraints, Proc. of the American Control Conference, 4653–4658.

Donida F, Romani C, Casella F and Lovera M (2009), Towards integrated modeling and parameter estimation: an LFT-Modelica approach, Proc. of the 15th IFAC Symposium on System Identification, Saint-Malo, France, 1286–1291.

Formentin S, Piga D, Tóth R and Savaresi SM (2016). Direct learning of LPV controllers from data. *Automatica* 65: 98–110.

Hjartarson A, Seiler PJ and Packard A (2015), LPVTools: A toolbox for modeling, analysis, and synthesis of parameter varying control systems, Proc. of the 1st IFAC Workshop on Linear Parameter Varying Systems, 139–145.

Hoffmann C and Werner H (2015), Compact LFT-LPV modeling with automated parameterization for efficient LPV controller synthesis, Proc. of the American Control Conference, Chicago, IL, USA, 119–124.

Isidori A (1995). Nonlinear Control Systems: An introduction, Lecture Notes in Control and Information Sciences, Springer, Berlin.

Koelewijn PJW (2023). Analysis and Control of Nonlinear Systems with Stability and Performance Guarantees: A Linear Parameter-Varying Approach. Ph.D. thesis, Eindhoven University of Technology.

Koelewijn P and Tóth R (2020), Scheduling dimension reduction of LPV models - a deep neural network approach, American Control Conference, Denver, USA, 1111–1117.

Kulcsár B, Dong J, van Wingerden JW and Verhaegen M (2009), LPV subspace identification of a DC motor with unbalanced disc, Proc. of the 15th IFAC Symposium on System Identification, Saint-Malo, France, 856–861.

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

Kwiatkowski A, Werner H and Boll MT (2006), Automated generation and assessment of affine LPV models, Proc. of the 45th IEEE Conf. on Decision and Control, San Diego, California, USA, 6690–6695.

Laurain V, Gilson M, Tóth R and Garnier H (2010). Refined instrumental variable methods for identification of LPV Box-Jenkins models. *Automatica* 46 (6): 959–967.

Lee LH (1997). Identification and Robust Control of Linear Parameter-Varying Systems. Ph.D. thesis, University of California, Berkeley.

Leith DJ and Leithhead WE (1998). Gain-scheduled controller design: An analytic framework directly incorporating non-equilibrium plant dynamics. *International Journal of Control* 70: 249–269.

Lescher F, Zhao JY and Martinez A (2006), Feb., Multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ control of a pitch regulated wind turbine for mechanical load reduction, Proc. of the European Wind Energy Conf., Athens, Greece.

Marcos A and Balas GJ (2004). Development of linear-parameter-varying models for aircraft. *J. of Guidance, Control and Dynamics* 27: 218–228.

Mohammadpour J and Scherer CW (2011). Control of linear parameter varying systems with applications, Springer-Verlag.

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

Papageorgiou G, Glover K, D'Mello G and Patel Y (2000), Taking robust LPV control into flight on the VAAC Harrier, Proc. of the 39th IEEE Conf. on Decision and Control, Sydney, Australia, 4558–4564.

Petersson D and Löfberg J (2009), Optimization based LPV-approximation of multiple model systems, Proc. of the European Control Conf., Budapest, Hungary, 3172–3177.

Petersson D and Löfberg J (2012), Identification of LPV state-space models using $\mathcal{H}_2$ minimisation, Varga A, Hansson A and Puyou G, (Eds.), Optimization Based Clearance of Flight Control Laws, 111–128.

Preda V, Cieslak J, Henry D, Bennani S and Falcoz A (2018). Robust microvibration mitigation and pointing performance analysis for high stability spacecraft. *International Journal of Robust and Nonlinear Control* 28 (18): 5688–5716.

Qin W and Wang Q (2007). An LPV approximation for admission control of an internet web server: Identification and control. *Control Engineering Practice* 15: 1457–1467.

Rugh WJ (1991). Analytical framework for gain scheduling. *IEEE Control Systems Magazine* 11 (1): 79–84.

Rugh W and Shamma JS (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: 3363–3373.

Scherer CW (1996). Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control for time-varying and linear parametrically-varying systems. *Int. Journal of Robust and Nonlinear Control* 6 (9-10): 929–952.

Scherer CW (2001). LPV control and full block multipliers. *Automatica* 37 (3): 361–375.

Scherer CW and Weiland S (2021). Linear matrix inequalities in control. *Lecture Notes, Dutch Institute for Systems and Control.* .

Shamma JS and Athans M (1992). Gain scheduling: potential hazards and possible remedies. *IEEE Control Systems Magazine* 12 (3): 101–107.

Skogestad S and Postlethwaite I (1996). Multivariable feedback control-analysis and design, John Wiley and Sons.

Steinbuch M, van de Molengraft R and van der Voort A (2003), Experimental modeling and LPV control of a motion system, Proc. of the American Control Conf., Denver, Colorado, USA, 1374–1379.

Stilwell DJ and Rugh WJ (2000). Stability preserving interpolation methods for synthesis of gain scheduled controllers. *Automatica* 36: 665–671.

Tanelli M, Ardagna D and Lovera M (2009), LPV model identification in virtualized service center environments, Proc. of the 15th IFAC Symposium on System Identification, Saint-Malo, France, 862–867.

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

Tóth R, Heuberger PSC and Van den Hof PMJ (2010). Discretisation of linear parameter-varying state-space representations. *IET Control Theory & Applications* 4: 2082–2096.

Tóth R, Abbas H and Werner W (2012a). On the state-space realization of LPV input-output models: Practical approaches. *IEEE Trans. on Control Systems Technology* 20: 139–153.

Tóth R, Heuberger PSC and Van den Hof PMJ (2012b), Prediction error identification of LPV systems: present and beyond, Mohammadpour J and Scherer CW, (Eds.), Control of Linear Parameter Varying Systems with Applications, Springer, Heidelberg, 27–60.

van Wingerden JW and Verhaegen M (2009). Subspace identification of bilinear and LPV systems for open- and closed-loop data. *Automatica* 45 (2): 372–381.

Veenman J, Scherer CW and Köroglu H (2009), IQC-based LPV controller synthesis for the NASA HL20 atmospheric re-entry vehicle, Proc. of the AIAA Guidance, Navigation, and Control Conference, Chicago, Illinois, USA, 1–18.

Veenman J, Scherer CW, Ardura C, Bennani S, Preda V and Girouart B (2021), IQClab: A new IQC based toolbox for robustness analysis and control design, Proc. of the 4st IFAC Workshop on Linear Parameter Varying Systems, 69–74.

Verdult V, Lovera M and Verhaegen M (2004). Identification of linear parameter-varying state space models with application to helicopter rotor dynamics. *Int. Journal of Control* 77 (13): 1149–1159.

Verhoek C, Beintema GI, Haesaert S, Schoukens M and Tóth R (2022a), Deep-learning-based identification of LPV models for nonlinear systems, Proc. of the 61st IEEE Conference on Decision and Control, Cancun, Mexico, 3274–3280.

Verhoek C, Tóth R and Abbas HS (2022b). Direct data-driven state-feedback control of linear parameter-varying systems. *arXiv preprint arXiv:2211.17182* .
Vizer D, Mercère G, Laroche E and Prot O (2014), LPV modeling and identification of a 2-DOF flexible robotic arm from local experiments using an H_∞-based glocal approach, Lovera M, (Ed.), Control-oriented modelling and identification: theory and practice, IET.
Wang R, Tóth R and Manchester IR (2024). Virtual control contraction metrics: Convex nonlinear feedback design via behavioral embedding. *International Journal of Robust and Nonlinear Control* .
Wassink MG, van de Wal M, Scherer CW and Bosgra O (2004). LPV control for a wafer stage: Beyond the theoretical solution. *Control Engineering Practice* 13 (2): 231–245.
Zhang Q and Ljung L (2017). From structurally independent local lti models to lpv model. *Automatica* 84: 232–235.
Zhou K, Doyle JC and Glover K (1995). Robust and Optimal Control, Prentice-Hall.