2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Robust LPV Modeling of Precision Motion Systems via Edge-Theorem Verification

Yazan M. Al-Rawashdeh Mohammad Al Janaideh

***Abstract*— This work proposes a systematic workflow for constructing grid-based Linear Parameter-Varying (LPV) models from frequency response data. Transfer functions are estimated at multiple scheduling-parameter grid points, fitted with a fixed model order, and transformed into controllable canonical realizations to ensure structural consistency. These vertex models are interpolated into an LPV state-space representation, while robust stability is verified using the Edge Theorem, which reduces the problem to checking edge polynomials of the convex hull. The novelty of the approach lies in integrating frequency-domain identification, canonical-form embedding, and polytope-based robust stability analysis into a unified LPV framework. Unlike conventional methods that rely on time-domain experiments or subspace techniques, the proposed method exploits experimentally accessible frequency-response data and avoids coordinate mismatches during interpolation. Validation on a precision motion system demonstrates both theoretical soundness and practical applicability, confirming the workflow as a reliable pathway from frequency-domain data to robust LPV control design.**

I. Introduction

Accurate models are essential for model-based control, system identification, and learning applications [1]. In industrial contexts such as semiconductor manufacturing, reliable models enable a workload distribution based on tracking error budgets [2]. Precision motion systems such as wafer scanners exhibit position-dependent dynamics [1], [3], nonlinearities [4], and resonance shifts from local linearizations [5], complicating modeling and control. Nonlinear methods [4], [6] capture rich dynamics, but are rarely controller-friendly, while linearized models are tractable, yet only locally valid [5].

Robust control [7] and gain-scheduling [8] mitigate these issues, but face conservatism or stability limitations. Linear parameter-varying (LPV) control provides a systematic alternative, although its success depends on the quality of the underlying LPV model [9]. Identification remains challenging: subspace methods [10] face state alignment issues, nonlinear block models [11] are complex, and multi-model techniques [6] struggle with robust stability.

Y. M. Al-Rawashdeh is with the Computer Engineering Department, Princess Sumaya University for Technology, Al-Jubeiha 11941, Amman, Jordan, and the Electrical and Computer Engineering Department, Guelp University, Ontario, Canada, emails: y.alrawashdeh@psut.edu.jo, yalrawas@uoguelph.ca. M. Al Janaideh is with the Departments of Interdisciplinary Engineering and Electrical and Computer Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada; and the Department of Mechatronics Engineering, University of Jordan, Amman, Jordan, and the Fraunhofer Institute for Applied Optics and Precision Engineering, Albert-Einstein-Straße 7, 07745 Jena, Germany, email: maljanai@uoguelph.ca

This work proposes a frequency-domain workflow for LPV modeling based on experimentally obtained frequency response functions (FRFs) [1], [12]. FRFs are fitted with fixed-order transfer functions, converted into controllable canonical realizations for automatic state alignment [8], [13], and interpolated into a polytopic LPV model. Stability is verified using the Edge Theorem [14], [15], reducing the analysis to a finite set of edges.

Compared to gain-scheduling [8], [13], subspace methods [10], and regression approaches [16], our workflow combines frequency-domain fidelity, consistent state alignment, and robust stability checks in a modular, MATLAB-implementable process. The main contributions are: (i) a systematic FRF-based LPV modeling workflow; (ii) canonical realizations for state consistency; and (iii) Edge Theorem-based stability guaranties.

The paper is structured as follows: the problem statement, system handling, and the outline of the proposed method are provided in Section II, the characterization of the driven motion system using random walk steps is discussed in Section III, and the experimental verification is presented in Section IV. Final remarks are provided in the last section.

II. Mathematical Modeling of Motion System

1) Problem Statement: Adopting the input-output perspective from the desired motion $(p_d(t))$ to the actual motion $(p_a(t))$ yields the system Σ_1 depicted in Fig. 1.

Let the x-y plane spatial dynamics of the motion system shown in Fig. 2 be captured using a linear parameter-varying model structured in the controllable canonical form. Mathematically, this is given as

$$\begin{aligned}
\dot{\boldsymbol{\zeta}}(t) &= \mathbf{A}_x(\boldsymbol{\theta}_A(t))\,\boldsymbol{\zeta}(t) + \mathbf{B}_x\,x_d(t)\\
x_a(t) &= \mathbf{C}_x(\boldsymbol{\theta}_C(t))\,\boldsymbol{\zeta}(t) + \mathbf{D}_x(\boldsymbol{\theta}_D(t))\,x_d(t) + \nu_x(t)\\
\dot{\boldsymbol{\eta}}(t) &= \mathbf{A}_y(\boldsymbol{\psi}_A(t))\,\boldsymbol{\eta}(t) + \mathbf{B}_y\,y_d(t)\\
y_a(t) &= \mathbf{C}_y(\boldsymbol{\psi}_C(t))\,\boldsymbol{\eta}(t) + \mathbf{D}_y(\boldsymbol{\psi}_D(t))\,y_d(t) + \nu_y(t), \quad (1)
\end{aligned}$$

where x_a and y_a denote the approximate positions along the x and y axes, and $\boldsymbol{\zeta} \in \mathbb{R}^{n_x\times 1}$ denotes the states vector associated with the x-axis motion, additionally with $\mathbf{A}_x \in \mathbb{R}^{n_x\times n_x}$ as the system matrix, $\mathbf{B}_x \in \mathbb{R}^{n_x\times 1}$ as the input matrix, $\mathbf{C}_x \in \mathbb{R}^{1\times n_x}$ as the sensors matrix, $\mathbf{D}_x \in \mathbb{R}$ as the feedforward matrix, $x_d(t)$ as the desired position along the x-axis, and $\nu_x(t)$ as a random process with Gaussian distribution capturing the x-axis modeling errors. The varying parameters affecting the spatiotemporal variation of the matrices $\mathbf{A}_x, \mathbf{C}_x$ and $\mathbf{D}_x$ are $\boldsymbol{\theta}_A(t) :=$

$\boldsymbol{\theta}_A(x_d(t), y_d(t)) = [\theta_{A,n_x-1}(t), \cdots, \theta_{A,0}(t)]^{\mathrm{T}} \in \mathbb{R}^{n_x \times 1}$, $\boldsymbol{\theta}_C(t) \coloneqq \boldsymbol{\theta}_C(x_d(t), y_d(t)) = [\theta_{C,n_x-1}(t), \cdots, \theta_{C,0}(t)]^{\mathrm{T}} \in \mathbb{R}^{n_x \times 1}$, and $\boldsymbol{\theta}_D(t) \coloneqq \boldsymbol{\theta}_D(x_d(t), y_d(t)) \in \mathbb{R}$, with $y_d(t) \in \mathbb{R}$ denoting the desired position along the y-axis. Similarly, the associated variables related to the motion along the y-axis, as given in (1) follow where n_y is used instead of n_x to specify the model order in the y-direction. Note that n_x and n_y may be different, in general. Moreover, the pair $\rho(t) \coloneqq (x_d(t), y_d(t))$ is used as the scheduling parameter.

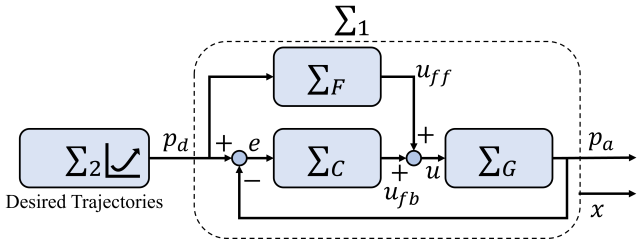

Fig. 1. The closed-loop representation of the driven motion system according to the combined feedforward-feedback control schema.

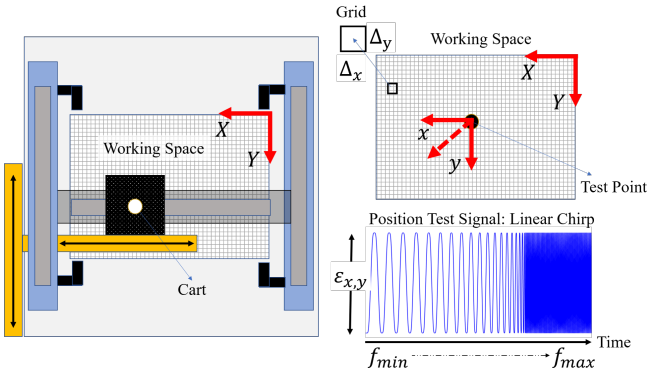

Fig. 2. The working space of the driven motion system showing a square grid of dimensions $\Delta_y \times \Delta_x$ and the linear chirp excitation signal used.

Remark II.1. *Utilizing the controllable canonical structure in* (1) *results in reducing the number of parameters used. For example, instead of having* $\boldsymbol{\theta}_A(t) \in \mathbb{R}^{n_x^2 \times 1}$, $\boldsymbol{\theta}_A(t) \in \mathbb{R}^{n_x \times 1}$ *is used.*

Recalling **Remark** II.1, the number of parameters needed in (1) to model the x-axis motion is equal to $2\,n_x + 1$ considering $\boldsymbol{\theta}_A(t), \boldsymbol{\theta}_C(t)$ and $\boldsymbol{\theta}_D(t)$. Similarly, the number of parameters needed in (1) to model the y-axis motion is equal to $2\,n_y + 1$ considering $\boldsymbol{\psi}_A(t), \boldsymbol{\psi}_C(t)$ and $\boldsymbol{\psi}_D(t)$. In total, $2\,(n_x + n_y) + 2$ parameters are needed in (1). The structures of the matrices $\mathbf{A}_x, \mathbf{B}_x, \mathbf{C}_x$, and $\mathbf{D}_x$ are given as

$$\mathbf{A}_x(t) = \begin{bmatrix} \theta_{A,n_x-1}(t) \cdots & \theta_{A,0}(t) \\ \mathbf{I}_{n_x-1 \times n_x-1} & \mathbf{0}_{n_x-1 \times 1} \end{bmatrix}, \mathbf{B}_x = [1, 0, \cdots, 0]^{\mathrm{T}}$$
$$\mathbf{C}_x(t) = [\theta_{C,n_x-1}(t), \cdots, \theta_{C,0}(t)],\ \mathbf{D}_x(t) = \theta_{D,0}(t), \quad (2)$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{0}$ is the zero vector. Similarly, the matrices $\mathbf{A}_y, \mathbf{B}_y, \mathbf{C}_y$, and $\mathbf{D}_y$ can be obtained using n_y and $\boldsymbol{\psi}_A(t), \boldsymbol{\psi}_C(t)$ and $\boldsymbol{\psi}_D(t)$ parameters associated with the motion along the y-axis. Interestingly, a similar structure to (2) - with $\mathbf{D}_x(t) = 0$ - was used in [5]. Having $\mathbf{D}_x(t) \neq 0$ will be useful in error compensation, as will be discussed in this study. More details about (1) will be provided in the coming subsections.

2) Working space and test points: Consider the motion system shown in Fig. 2, where the working space is partitioned into an $N_x \times N_y$ grid with cell dimensions $(\Delta x, \Delta y)$. A finer grid improves model accuracy at the expense of computational effort. Each corner of the grid defines a test point, yielding $N_x N_y$ points in total. At each test point, the system—kept active and fully controlled—is excited locally by a small reciprocating motion driven by a chirp signal of amplitude ε_x or ε_y and frequency range $f \in [f_{min}, f_{max}]$. The resulting frequency response functions (FRFs) are recorded. A similar concept was used in [5], though restricted to three points along a single axis. The grid resolution is constrained by the experiment economy [17].

These local FRFs are essential for fitting stable transfer functions of order n_x or n_y. Various methods can be applied, such as quadrature-based vector fitting [18] and subspace identification [19]; in this work, MATLAB's `tfest` is employed. The fitting quality depends strongly on both the chosen model order and the bandwidth (BW) of interest, which also influences the parameter count in (1). For instance, Fig. 3 shows the y-axis FRF at $(50, 50)$ fitted with order $n_y = 3$, where a BW of 100 Hz yields a stable and satisfactory model with 96.22% fit percentage. This example highlights the interplay between model order $\{n_x, n_y\}$, BW, fit percentage, and parameter count in (1), underscoring the need for careful trade-offs in constructing accurate and stable local models.

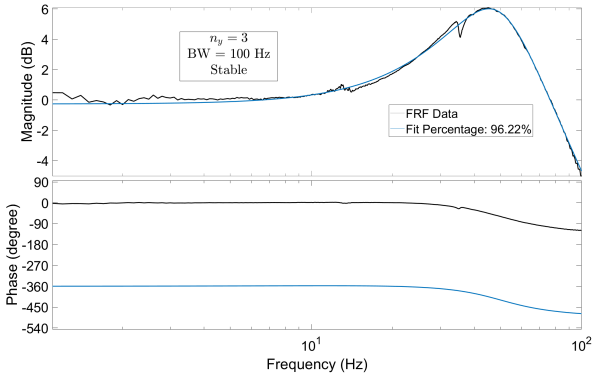

Fig. 3. The fitting results of the y-axis FRF at the test point $(50, 50)$ obtained using the MATLAB function `tfest` with order $n_y = 3$ resulting in a stable transfer function for BW 100 Hz.

To preserve the structure of the matrices used in (1) for a specific BW, both n_x and n_y must be constant throughout the entire working spaces, i.e., at each test point, where the resulting single-input single-output (SISO) linear time-invariant (LTI) transfer functions $G_i(s) = \mathbf{C}_i(s\mathbf{I} - \mathbf{A}_i)^{-1}\mathbf{B}_i + \mathbf{D}_i, i = 1, 2, \cdots, N_x N_y$ - $s \in \mathbb{C}$ being the complex frequency variable - are stable with satisfactory fitting percentages (preferably $\geq 95\%$). This will result

in a uniform mesh. Moreover, it is worth noting that the MATLAB function `tfest` produces strictly proper transfer functions ($\mathbf{D}_i = 0$); therefore, the steady-state errors for these transfer functions may not be zero.

To eliminate the tracking error $e_x(x_i, y_i, t)$ steady-state value $e_{x_{ss}}(x_i, y_i)$ at the i-th test point in the x direction, for example, mainly due to a step input $U(s) = \alpha/s, \alpha \in \mathbb{R}$ is its amplitude, an error compensation mechanism must be involved such that $G_i(0) = -\mathbf{C}_i\mathbf{A}_i^{-1}\mathbf{B}_i + \mathbf{D}_i = 1$. To that end, a feedforward path is realized with $\mathbf{D}_i \neq 0$ whose value is given using the final-value theorem as

$$\begin{aligned}\lim_{t\to\infty} e_x(x_i, y_i, t) &\coloneqq e_{x_{ss}}(x_i, y_i) = \alpha - \lim_{s\to 0} sG_i(s)U(s) = 0\\ &= \alpha - \lim_{s\to 0} sG_i(s)\frac{\alpha}{s} = 1 - G_i(0) = 0\\ &= 1 + \mathbf{C}_i\mathbf{A}_i^{-1}\mathbf{B}_i - \mathbf{D}_i = 0\\ \Rightarrow \mathbf{D}_i &= 1 + \mathbf{C}_i\mathbf{A}_i^{-1}\mathbf{B}_i, \end{aligned} \tag{3}$$

and is referred to as the *static spatial error compensation*. Interestingly, if $\mathbf{A}_i$ is ill-conditioned, then $\mathbf{D}_i$ in (3) may not be calculated correctly. As a remedy, the order n_x, or n_y, should be adjusted- typically reduced. The local SISO and LTI system that approximates the dynamics of the motion system at the vicinity of the i-th test point $(\Sigma_i(xi, yi) \coloneqq (\mathbf{A}_i, \mathbf{B}, \mathbf{C}_i, \mathbf{D}_i)), i = 1, 2, \cdots, N_xN_y$ is given along the x-axis as

$$\Sigma_{x_i}(x_i, y_i) \coloneqq \begin{cases} \dot{\boldsymbol{\zeta}}(t) & = \mathbf{A}_i\,\boldsymbol{\zeta}(t) + \mathbf{B}\,x_d(t)\\ x_a(t) & = \mathbf{C}_i\,\boldsymbol{\zeta}(t) + \mathbf{D}_i\,x_d(t), \end{cases} \tag{4}$$

where $\mathbf{D}_i \neq 0$ is the contribution of the proposed error compensation mechanism and not directly obtained from the `tfest` function. Similarly, $\Sigma_{y_i}(x_i, y_i)$ can be defined. Now, consider the convex hull comprised of the four vertices $\{v_1, v_2, v_3, v_4\}$ corresponding to four adjacent test points. Creating a convex hull of these vertices yields the dynamics along the x-axis given as

$$\begin{aligned}\Sigma_x(x_a, y_a) &\coloneqq \sum_{i=1}^{4} \alpha_i(t)\,\Sigma_{x_i}(x_i, y_i)\\ &= \begin{cases} \dot{\boldsymbol{\zeta}}(t) & = \mathbf{A}_x(\boldsymbol{\theta}_A(t))\,\boldsymbol{\zeta}(t) + \mathbf{B}_x\,x_d(t)\\ x_a(t) & = \hat{\mathbf{C}}_x(\boldsymbol{\theta}_C(t))\,\boldsymbol{\zeta}(t) + \hat{\mathbf{D}}_x(\boldsymbol{\theta}_D(t))\,x_d(t) \end{cases}\\ \mathbf{A}_x(\boldsymbol{\theta}_A(t)) &= \sum_{i=1}^{4} \alpha_i(t)\,\mathbf{A}_i, \quad \hat{\mathbf{C}}_x(\boldsymbol{\theta}_C(t)) = \sum_{i=1}^{4} \alpha_i(t)\,\mathbf{C}_i\\ \hat{\mathbf{D}}_x(\boldsymbol{\theta}_D(t)) &= \sum_{i=1}^{4} \alpha_i(t)\,\mathbf{D}_i, \quad \sum_{i=1}^{4} \alpha_i(t) = 1, \quad \alpha_i(t) \geq 0, \end{aligned} \tag{5}$$

with $\alpha_i(t) \coloneqq \alpha_i(\rho(t)) = \alpha_i(x_d(t), y_d(t)), i = 1, 2, 3, 4$ as the linear interpolating parameters. The proposed error compensation mechanism along the x-axis, for example, can be refined by introducing an additional term $(q_x(t) \neq 0)$ in the feedforward path when $|e_x(t)| \equiv |e_x(x, y, t)| = |x_d(t) - x_a(t)| > \delta_x$ with $\delta_x(\omega) > 0 \in \mathbb{R}$ denoting the motion system precision at the operating frequency ω (rad/s), which is usually known by observing the tracking error, for example. Mathematically, it is given as

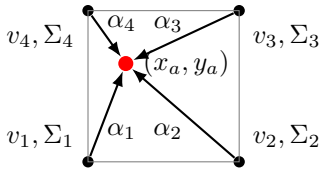

Fig. 4. The convex hull of a grid comprising four vertices and containing the point (x_a, y_a).

$$\begin{aligned}&|e_x(x, y, t)| = |x_d(t) - \{x_a(t) + q_x(t)\,x_d\}| \leq \delta_x(\omega)\\ &\frac{e_x(t) - \delta_x(\omega)}{x_d(t)} \leq q_x(t) \leq \frac{e_x(t) + \delta_x(\omega)}{x_d(t)}, x_d(t) \neq 0\\ \Rightarrow &q_x(t) = \frac{e_x(t) - \delta_x(\omega)}{x_d(t)} + \cdots\\ &\left\{\frac{e_x(t) + \delta_x(\omega)}{x_d(t)} - \frac{e_x(t) - \delta_x(\omega)}{x_d(t)}\right\} r_x\\ &\Rightarrow q_x(t) = \frac{1}{x_d(t)}\{e_x(t) + (2\,r_x - 1)\delta_x(\omega)\},\ x_d(t) \neq 0\\ &r_x \sim \text{Unif}(0, 1) \end{aligned} \tag{6}$$

where r_x is a random number drawn from a uniform distribution (Unif), with a mean equal to 0.5. Using (6) in (5) yields the LPV system dynamics along the x-axis, including the full error compensation mechanism, which is given as

$$\begin{aligned}\dot{\boldsymbol{\zeta}}(t) &= \mathbf{A}_x(\boldsymbol{\theta}_A(t))\,\boldsymbol{\zeta}(t) + \mathbf{B}_x\,x_d(t)\\ x_a(t) &= \hat{\mathbf{C}}_x(\boldsymbol{\theta}_C(t))\,\boldsymbol{\zeta}(t) + \{\hat{\mathbf{D}}_x(\boldsymbol{\theta}_D(t)) + q_x(t)\}\,x_d(t)\\ &= \left\{\frac{1}{2}\hat{\mathbf{C}}_x(\boldsymbol{\theta}_C(t))\right\}\boldsymbol{\zeta}(t) + \cdots\\ &\quad \left\{\frac{1}{2}\hat{\mathbf{D}}_x(\boldsymbol{\theta}_D(t)) + \frac{1}{2}\right\}x_d(t) + \left(r_x - \frac{1}{2}\right)\delta_x(\omega)\\ &= \mathbf{C}_x(\boldsymbol{\theta}_C(t))\,\boldsymbol{\zeta}(t) + \mathbf{D}_x(\boldsymbol{\theta}_D(t))\,x_d(t) + \nu_x(t), \end{aligned} \tag{7}$$

which is the motion system along the x-axis given in (1). Similarly, the LPV dynamics along the y-axis can be derived.

3) Steps of the proposed workflow: According to the proposed workflow, it is required to do the following:

- To model the motion system as an LPV model
  1) Determine the needed grid resolution, i.e., $(N_x \times N_y)$ vertices.
  2) Obtain the FRF response at each vertex.
  3) Determine the orders n_x and n_y.
  4) Fit monic and, preferably, strictly proper *stable* transfer functions at each vertex, e.g., use `tfest` built-in Matlab function. The fitting percent should be acceptable according to the study objectives.

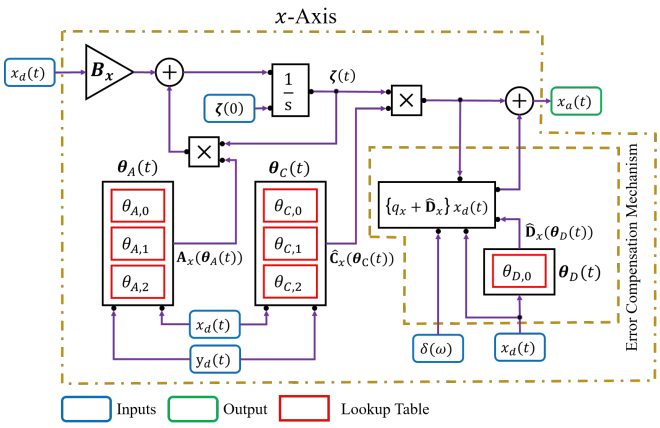

Fig. 5. An illustrative simulation outline based on the proposed workflow showing the x-axis only with $n_x = 3$. The y-axis outline follows seamlessly.

5) Convert the obtained transfer function into SISO state-space canonical representation, e.g., controllable.
6) Store the resulting matrices $\{\mathbf{A}_i, \mathbf{C}_i\}$, and the calculated static error compensation value $\mathbf{D}_i$. Note that $\mathbf{B}$ is fixed and known for all the vertices, assuming that a fixed-order controllable canonical form is used along the same axis of motion.
7) Check global stability. If passed, then proceed; otherwise, go to step 3.
8) *Linearly* interpolate the matrices, or equivalently their parameters, convexly via the scheduling parameter $\rho = (x_d, y_d)$ evaluated at each vertex in the grid.
9) Store the interpolated parameter surfaces into lookup tables to be used in the simulation environment.
10) Build the simulation model similar to the outline depicted in Fig.5.

- To check the obtained LPV (open-loop) stability
  1) Local stability: Make sure that *all* the fitted transfer functions or $\mathbf{A}_i, i = 1, 2, \cdots, N_x N_y$ matrices are stable (Hurwitz) using the same orders (n_x, n_y). You may use `pole, eig` built-in Matlab functions.
  2) Global stability: The well-known *Edge Theorem* [15] can be used.

4) Satbility of LPV: According to the proposed method, the FRF at each vertex v_i at the grid position (x_i, y_i) along the x-axis - for example - is fitted using a *fixed-order* n_x strictly proper transfer function with *monic* denominator and is given as

$$G_{x_i}(s) = \frac{b^{(i)}_{n_x-1,x}\, s^{n_x-1} + \cdots + b^{(i)}_{1,x}\, s + b^{(i)}_{0,x}}{s^{n_x} + a^{(i)}_{n_x-1,x}\, s^{n_x-1} + \cdots + a^{(i)}_{1,x}\, s + a^{(i)}_{0,x}} + d^{(i)} = \frac{l_{x_i}(s)}{p_{x_i}(s)} \tag{8}$$

where the superscript (i) denotes the vertex index. As stated earlier, the built-in MATLAB `tfest` fits a strictly proper transfer function of order n_x; and therefore, $d^{(i)} = 0$ is obtained in (8).

Considering the design of the proposed workflow, the (open-loop) stability of the resulting LPV model can be preferably checked using the Edge Theorem [15]. This theorem fits neatly in the proposed workflow because of how the LPV system is built. Let $\mathcal{P}$ be a polytope of real-coefficient N polynomials of *fixed degree* n given by

$$\mathcal{P} \triangleq \operatorname{conv}\{\, p_1(s), \ldots, p_N(s)\,\},$$
$$p_i(s) = s^n + a_{i,n-1}s^{n-1} + \cdots + a_{i,0}, \quad i = 1, 2, \cdots, N.$$

Thus any $p(s) \in \mathcal{P}$ can be written as

$$p(s) = \sum_{i=1}^{N} \lambda_i p_i(s), \qquad \lambda_i \geq 0, \quad \sum_{i=1}^{N} \lambda_i = 1,$$

and the degree of $p(s)$ is n *for all convex combinations* (i.e., no degree drop on the boundary). Denote by $\mathcal{E}$ the set of *exposed edges* of $\mathcal{P}$, i.e.,

$$\mathcal{E} \triangleq \Big\{ \operatorname{conv}\{p_i, p_j\} : \overline{p_i p_j} \Big\}.$$

where $\overline{p_i p_j}$ is an edge of the convex hull in coefficient space. For a polynomial p, write $p \in \mathcal{H}$ if it is *Hurwitz* (all roots in the open left half-plane).

Definition II.1 (Robust Hurwitz stability on a family)**.** *A set of polynomials $\mathcal{S}$ is* robustly Hurwitz *if every $p \in \mathcal{S}$ is Hurwitz.*

Theorem II.1 (Edge Theorem for polynomial polytopes [15])**.** *Let $\mathcal{P}$ be a polytope of real polynomials of fixed degree n as above. Then $\mathcal{P}$ is robustly Hurwitz if and only if* every polynomial on every exposed edge *is Hurwitz; i.e.,*

$$\mathcal{P} \subset \mathcal{H} \iff \forall\, \operatorname{conv}\{p_i, p_j\} \in \mathcal{E},$$
$$\{\lambda p_i + (1-\lambda) p_j : \lambda \in [0,1]\} \subset \mathcal{H}.$$

Equivalently, it suffices to check all one-dimensional exposed edges.

Remark II.2. *The Edge Theorem is applicable for any order n_x, n_y, and this is another reason why it fits neatly in the proposed workflow.*

III. Point-to-Point Positioning Characterization

In this section, the driven motion system characterization is discussed, c.f. [3], [20], under mainly the frequency-aware B-spline (F-B-spline) motion profiles that are introduced in [21]. Also in [21], the point-to-point tracking errors during the random walk motion with no dwell are harvested at each vertex in the grid that covers the working space or a portion of it.

1) Tracking error computation: At each i-th vertex, the averaged tracking error ($||e||_i$) is obtained by averaging the associated tracking errors of each test run. The averaged tracking error reduces the contribution of the measurement noise and any path-dependent behavior of the motion system and is given as [21]

$$||e||_i = \frac{1}{N_{m,i}} \sum_{r=1}^{N_{m,i}} \left(\frac{1}{N_{i,r}} \sum_{k=1}^{N_{i,r}} \sqrt{e_{x_{i,k,r}}^2 + e_{y_{i,k,r}}^2} \right) \quad (9)$$

where $N_{m,i}$ is the runs counted during which the i^{th} control point is visited, $N_{i,r}$ is the repetition number of the i-th vertex v_i in each run.

2) Random Walk: A random walk process is a discrete-time stochastic process that comprises sums of independent and identically distributed random variables or random vectors in Euclidean space [22]. This motion is used to assess the LPV model quality of the studied motion system.

IV. Experimental Work and Model Validation

The LPV model of the motion system in Fig. 7 is validated using F-B-spline profiles [21]. The platform is a $450\times450\,\mathrm{mm}^2$ PIglide HS XY scanner with $1\,\mathrm{nm}$ resolution, driven by an ACS controller and H-configuration linear motors [23]. Its workspace is meshed into $50\times50\,\mathrm{mm}^2$ grids (81 vertices).

1) Selecting Error Bounds: Based on the tracking error measurements [21] as depicted in Fig. 6, the value of $\delta_x(w), \delta_y(\omega)$ given in (7) can be determined as functions of the operating frequency ω. Since $||e||$ is available, it is assumed that $\delta_x(w) = \delta_y(\omega) \sim \mu_{||e||}(\omega)$, where $\mu_{||e||}(\omega)$ denotes the mean of $||e||$ as a function of the operating frequency ω. These values are listed in Table I and approximated as normal distributions as depicted in Fig. 6. Mathematically, from (6), having $q_x(t) = q_y(t) = 0$ requires $e_x(t) \sim (1-2r_x)\delta_x$ and $e_y(t) \sim (1-2r_y)\delta_y$; therefore, δ_x can be given as

$$\begin{aligned} &||e|| = \sqrt{e_x^2 + e_y^2} \sim \delta_x \sqrt{(2\,r_x - 1)^2 + (2\,r_y - 1)^2} \\ &\text{where } r_x, r_y \sim \mathrm{Unif}(0,1). \text{ Let } U = 2\,r_x - 1, V = 2\,r_y - 1 \\ &\Rightarrow U, V \sim \mathrm{Unif}(-1,1). \text{ Then,} \\ &\mathbb{E}[||e||] = \mu_{||e||} = \delta_x\, \mathbb{E}\left[\sqrt{U^2 + V^2}\right] = \\ &\frac{1}{4}\delta_x \int_{-1}^{1}\int_{-1}^{1} \sqrt{u^2+v^2}\, du\, dv = \delta_x \frac{\sqrt{2} + \ln(1+\sqrt{2})}{3}, \\ &\Rightarrow \delta_x = \delta_y = \frac{3\,\mu_{||e||}}{\sqrt{2} + \ln(1+\sqrt{2})} \approx 1.3068552\,\mu_{||e||} \end{aligned} \quad (10)$$

In this study, $\delta_x = \delta_y = \sqrt{2}\,\mu_{||e||} \approx 1.4142\,\mu_{||e||}$ is used instead of the value given by (10). Note that in (10) the norm of the tracking error is assumed to be drawn from a normal distribution as depicted in Fig. 6 and U, V are drawn from Unif distribution. Consequently, in (10), matching only the mean values of the two distributions is achieved. To match the variances instead, $\delta_x \approx 3.5112\,\sigma_{||e||}$ with $\sigma_{||e||}$ being the tracking error standard deviation. To match the means and variances simultaneously using a single δ_x value, the condition $\sigma_{||e||}/\mu_{||e||} \approx 0.372$ must be satisfied; otherwise, it is impossible. More details are beyond the scope of this study.

At each vertex, $\mathrm{FRF}_x, \mathrm{FRF}_y$ are collected and stable transfer functions are obtained with $n_x, n_y = 9$ and BW of 600 Hz. These transfer functions are then converted into state-space representation with fit percentage listed in Table II along the x-axis. The y-axis systems have similar results. Stability is ensured by **Theorem** II.1. The interpolated parameters of $\mathbf{A}_i, \mathbf{C}_i, , \mathbf{D}_i$ matrices are generated via MATLAB `curveFitter` [24].

TABLE I
THE VALUES OF $\mu_{||e||}$ FOR THE LPV MODEL BASED ON THE TRACKING ERRORS MEASUREMENTS [21].

ω (rad/s)	**40**	**50**	**60**	**70**	**80**	**90**
$\mu_{\|\|e\|\|}$ (μm)	0.123	0.39	0.83	0.95	1.11	2.29

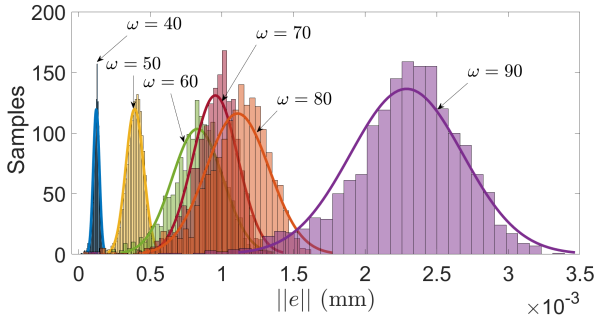

Fig. 6. The statistical analysis of the collected measurements [21] showing almost normal distribution of the working space sampled tracking error over a grid of size $10\,\mathrm{mm}$ using $\omega \in \{40, 50, 60, 70, 80, 90\}\,\mathrm{rad/s}$.

Using the random-walk map from [25], the results show that $n_x = n_y = 9$ yields an accurate LPV digital twin of the studied motion system, thereby validating the workflow presented in this study.

2) Summary, Advantages, and Limitations: The proposed workflow presents several attractive features. First, it bridges frequency-domain and LPV modeling by enabling the direct use of FRF data, which is widely available in many engineering applications. Second, the use of a controllable canonical form across all vertices ensures consistent interpolation and well-defined LPV embedding. Third, stability is efficiently guaranteed using the Edge Theorem, which transforms the infinite stability-checking problem into a finite number of verifications. Finally, the workflow is straightforward to implement, modular, and mathematically transparent. Its experimental validation in a precision motion system highlights its robustness and credibility for real-world deployment. Nevertheless, there are several limitations. The fixed-order requirement for transfer function fitting may lead to either high model complexity or insufficient accuracy if not carefully chosen.

The Edge Theorem applies primarily to convex polytopes, which limits generalizability. Additionally, parameter-derivative effects are not explicitly modeled (embedded in the chosen LPV representation), scalability suffers with decreasing grid dimension, and convex interpolation may not perfectly capture nonlinear parameter dependencies. Although these limitations are inherent to the workflow, the demonstrated agreement between simulation and experimental results in a precision motion system suggests that the method is resilient in practice.

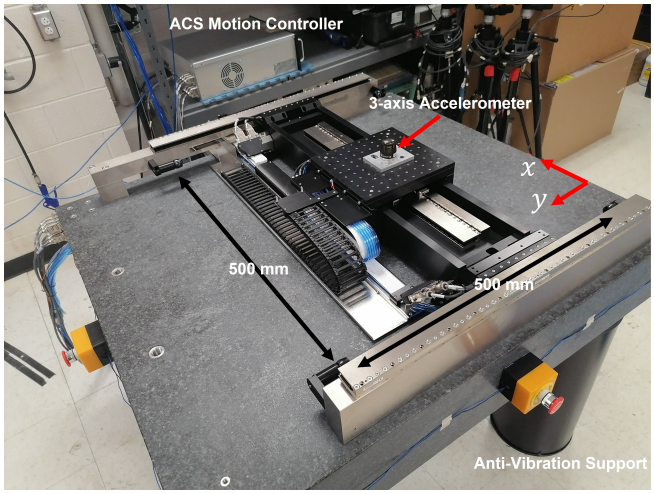

Fig. 7. The motion system used in this study.

TABLE II

THE FITTING RESULTS (%) OF x-AXIS FRF RESPONSES USING STABLE TRANSFER FUNCTIONS OF ORDER 9.

y	x → 50	100	150	200	250	300	350	400	450
50	89.77	89.97	89.77	89.46	89.20	88.71	88.65	89.62	89.56
100	89.68	89.94	89.77	89.35	88.74	88.28	88.52	89.27	89.48
150	90.51	90.95	90.69	90.50	90.16	89.43	89.48	90.60	90.26
200	90.43	90.99	90.91	90.49	90.15	89.39	89.45	90.30	90.34
250	90.50	90.90	90.69	90.39	90.10	89.20	89.13	90.01	89.90
300	90.52	90.86	90.53	90.17	89.77	89.12	89.11	90.14	89.69
350	90.47	90.70	90.52	90.09	89.60	88.80	89.10	89.86	89.43
400	90.19	90.75	90.68	90.10	89.53	88.87	88.88	89.65	89.06
450	90.34	90.92	90.89	90.38	89.84	89.17	89.12	89.75	89.36

V. CONCLUSION

We proposed a workflow for constructing LPV models directly from frequency-response data, combining canonical realizations and Edge Theorem-based stability checks. Its modularity, simplicity, and reliance on standard tools make it attractive for practical adoption, while its integration of canonical realizations and Edge Theorem stability analysis ensures theoretical soundness. Validation on a precision motion system showed strong experimental–simulation agreement. Future work targets advanced interpolation, scalable stability analysis, and integration with LPV controller synthesis.

ACKNOWLEDGMENT

This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a Discovery Grant, a Research Tools and Instruments grant, the Canada Foundation for Innovation (CFI), and the University of Guelph.

REFERENCES

[1] T. Oomen, "Advanced motion control for precision mechatronics: Control, identification, and learning of complex systems," *IEEJ Journal of Industry Applications*, vol. 7, no. 2, pp. 127–140, 2018.

[2] L. Chen, Y. Qiao, N. Wu, M. Ghahramani, Y. Shao, and S. Zhan, "Workload balancing for photolithography machines in semiconductor manufacturing via estimation of distribution algorithm integrating kmeans clustering," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 55, no. 8, pp. 5565–5580, 2025.

[3] R. Voorhoeve, R. de Rozario, W. Aangenent, and T. Oomen, "Identifying position-dependent mechanical systems: A modal approach applied to a flexible wafer stage," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 1, pp. 194–206, 2020.

[4] A. F. Guc, Z. Yumrukcal, and O. Ozcan, "Nonlinear identification and optimal feedforward friction compensation for a motion platform," *Mechatronics*, vol. 71, p. 102408, 2020.

[5] M. G. Wassink, M. Van De Wal, C. Scherer, and O. Bosgra, "Lpv control for a wafer stage: beyond the theoretical solution," *Control Engineering Practice*, vol. 13, no. 2, pp. 231–245, 2005.

[6] A. A. Adeniran and S. El Ferik, "Modeling and identification of nonlinear systems: A review of the multimodel approach—part 1," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1149–1159, 2016.

[7] P. Apkarian and R. J. Adams, "Robust multiobjective control with dynamic output feedback controllers," *Automatica*, vol. 34, no. 4, pp. 585–601, 1998.

[8] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[9] R. de Rozario, T. Oomen, and M. Steinbuch, "Iterative learning control and feedforward for lpv systems: Applied to a position-dependent motion system," in *American Control Conference (ACC)*, pp. 3518–3523, IEEE, 2017.

[10] J.-W. van Wingerden and M. Verhaegen, "Subspace identification of bilinear and lpv systems for open- and closed-loop data," *Automatica*, vol. 45, no. 2, pp. 372–381, 2009.

[11] J. Ma, B. Huang, and F. Ding, "Iterative identification of hammerstein parameter varying systems with parameter uncertainties based on the variational bayesian approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 3, pp. 1035–1045, 2017.

[12] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach*. John Wiley & Sons, 2012.

[13] D. J. Leith and W. E. Leithead, "Gain-scheduled controller design: An overview," *International Journal of Control*, vol. 73, no. 11, pp. 1001–1025, 2000.

[14] A. Bartlett, C. Hollot, and L. Huang, "Strongly robust stability of polynomials and matrices—a graphical approach," *IEEE Transactions on Automatic Control*, vol. 33, no. 6, pp. 584–593, 1988.

[15] A. Bartlett, C. Hollot, and H. Lin, "The edge theorem: a necessary and sufficient condition for robust stability of a polytope of polynomials," in *Conference on Decision and Control*, pp. 1737–1742, IEEE, 1987.

[16] R. Tóth, *Modeling and Identification of Linear Parameter-Varying Systems*. Springer, 2010.

[17] N. Dirkx and T. Oomen, "Multivariable experiment design with application to a wafer stage: a sequential relaxation approach for dealing with element-wise constraints," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 8565–8570, 2020.

[18] Z. Drmac, S. Gugercin, and C. Beattie, "Quadrature-based vector fitting for discretized h_2 approximation," *SIAM Journal on Scientific Computing*, vol. 37, no. 2, pp. A625–A652, 2015.

[19] G. Van der Veen, J.-W. van Wingerden, M. Bergamasco, M. Lovera, and M. Verhaegen, "Closed-loop subspace identification methods: an overview," *IET Control Theory & Applications*, vol. 7, no. 10, pp. 1339–1358, 2013.

[20] H. K. Shirvani, J. Q. C. Zeng, P. Bevers, T. Oomen, and K. Erkorkmaz, "Linear time-invariant model identification algorithm for mechatronic systems based on mimo frequency response data," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 2, pp. 703–714, 2023.

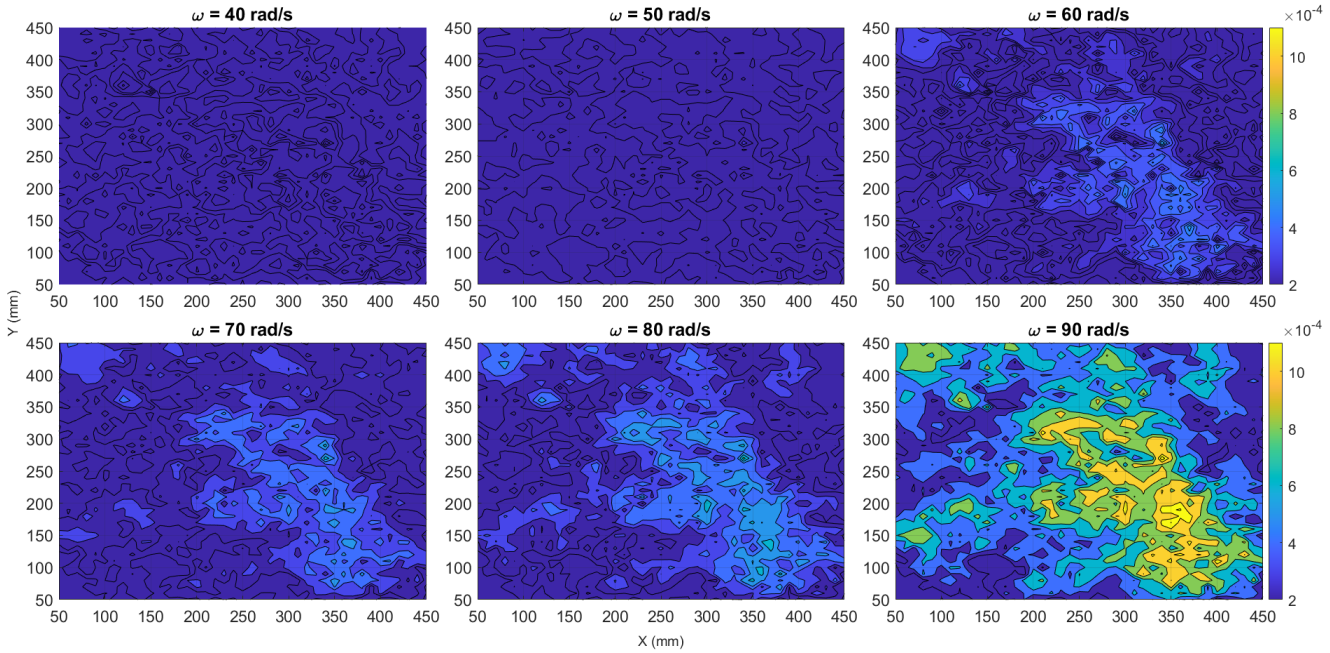

Fig. 8. Spatial point-to-point tracking error contour plots (in mm) based on the *LPV model*, *order 9*, BW 600 Hz, and 10 runs of random walk steps at different frequencies using F-B-spline profiles [21] with 20000 random walk steps per frequency, calculated using (9) over a grid size of 10 mm.

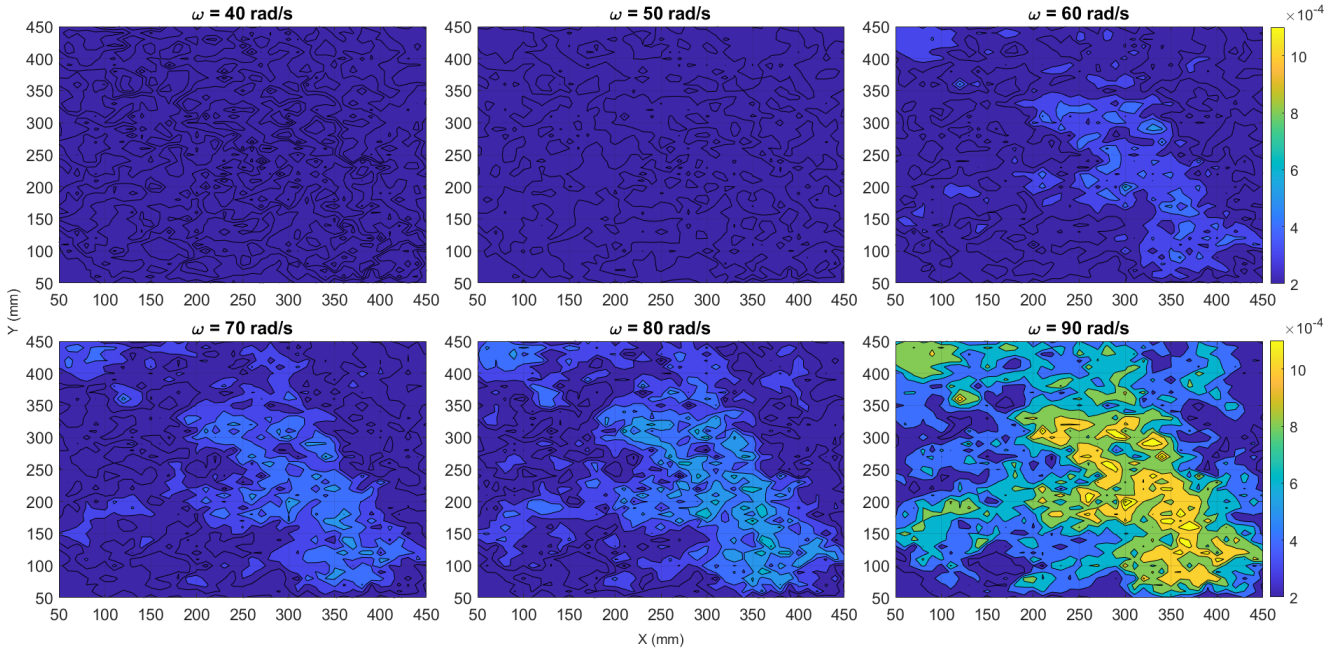

Fig. 9. Spatial point-to-point tracking error contour plots (in mm) based on the *experimental data* and 10 runs of random walk steps at different frequencies using frequency-aware B-spline profiles [21] with 20000 random walk steps per frequency, calculated using (9) over a grid size of 10 mm.

[21] Y. M. Al-Rawashdeh, M. Heertjes, and M. Al Janaideh, "Statistical characterization of position-dependent behavior using frequency-aware b-spline," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8270–8275, 2023.

[22] A. Klenke, *Probability Theory: A Comprehensive Course*. Springer, 2013.

[23] Y. M. Al-Rawashdeh, M. Al Janaideh, and M. Heertjes, "On characterization of a generic lithography machine in a multi-directional space," *Mechanism and Machine Theory*, vol. 170, p. 104638, 2022.

[24] K. R. Green, T. A. Bohn, and R. J. Spiteri, "Direct function evaluation versus lookup tables: When to use which?," *SIAM Journal on Scientific Computing*, vol. 41, no. 3, pp. C194–C218, 2019.

[25] Y. M. Al-Rawashdeh, M. F. Heertjes, and M. Al Janaideh, "Point-to-point reference trajectories generation using frequency-aware b-splines/nurbs," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 21325–21340, 2025.