Gaussian Process Based Prediction and Control of Hammerstein-Wiener Systems

Mingzhou Yin

Institute of Automatic Control
Leibniz University Hannover

January 21, 2025

Multi-step-ahead data-driven predictor

- Data: $(\mathbf{u}^d, \mathbf{y}^d) := (u_k^d, y_k^d)_{k=1}^N$
- Initial condition horizon: L_0, prediction horizon: L', $L := L_0 + L'$
- Prediction condition: $\mathbf{u}_{\mathsf{p}} := (u_k)_{k=1}^{L_0}$, $\mathbf{y}_{\mathsf{p}} := (y_k)_{k=1}^{L_0}$ (initial condition), $\mathbf{u}_{\mathsf{f}} := (u_k)_{k=L_0+1}^{L}$ (input)
- Output prediction: $\mathbf{y}_{\mathsf{f}} := (y_k)_{k=L_0+1}^{L}$

- **Problem:** Learn the mapping $\mathbf{y}_{\mathsf{f}} = f(\mathbf{u}, \mathbf{y}_{\mathsf{p}})$ from $(\mathbf{u}^d, \mathbf{y}^d)$, specifically

$$H_u = \begin{bmatrix} u_1^d & u_2^d & \cdots & u_{N-L+1}^d \\ u_2^d & u_3^d & \cdots & u_{N-L+2}^d \\ \vdots & \vdots & \ddots & \vdots \\ u_L^d & u_{L+1}^d & \cdots & u_N^d \end{bmatrix}, \quad H_y = \begin{bmatrix} y_1^d & y_2^d & \cdots & y_{N-L+1}^d \\ y_2^d & y_3^d & \cdots & y_{N-L+2}^d \\ \vdots & \vdots & \ddots & \vdots \\ y_L^d & y_{L+1}^d & \cdots & y_N^d \end{bmatrix}, \quad \mathbf{u} := \begin{bmatrix} \mathbf{u}_{\mathsf{p}} \\ \mathbf{u}_{\mathsf{f}} \end{bmatrix}$$

Two ends of the spectrum

- Pure black box: e.g., Gaussian process

$$\eta := \begin{bmatrix} \mathbf{u} \\ \mathbf{y}_{\mathsf{p}} \end{bmatrix}, \quad f(\eta) \sim \mathcal{GP}\left(\mathbf{m}(\cdot), \mathbf{k}(\cdot,\cdot)\right)$$

$$\text{trained by inputs: } \eta_k^d = \mathrm{col}\left(u_k^d, \ldots, u_{k+L-1}^d, y_k^d, \ldots, y_{k+L_0-1}^d\right)$$

$$\text{outputs: } \chi_k^d = \mathrm{col}\left(y_{k+L_0}^d, \ldots, y_{k+L-1}^d\right)$$

- Linear systems + noise-free data: e.g., Willems' lemma

$$\mathbf{y}_{\mathsf{f}} \text{ satisfies } \begin{bmatrix} H_u \\ H_{y\mathsf{p}} \\ H_{y\mathsf{f}} \end{bmatrix} g = \begin{bmatrix} \mathbf{u} \\ \mathbf{y}_{\mathsf{p}} \\ \mathbf{y}_{\mathsf{f}} \end{bmatrix} \text{ for some } g, \quad \text{i.e., } \mathbf{y}_{\mathsf{f}} = [\Gamma_1 \ \Gamma_2] \begin{bmatrix} \mathbf{u} \\ \mathbf{y}_{\mathsf{p}} \end{bmatrix}, \ [\Gamma_1 \ \Gamma_2] := H_{y\mathsf{f}} \begin{bmatrix} H_u \\ H_{y\mathsf{p}} \end{bmatrix}^{\dagger}$$

Somewhere in the middle...

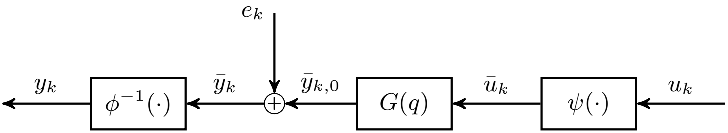


Figure: Hammerstein-Wiener systems

- Direct extension of Willems' lemma?
- For Hammerstein systems, basis function decomposition: $\psi(u) = \sum_{i=1}^{r} a_i \psi_i(u) = \left(\mathbf{a}^\top \otimes \mathbb{I}\right) \boldsymbol{\psi}(u)$
- Linear prediction by Willems' lemma using augmented inputs $(\psi_i(u_k))_{i=1}^{r}$
- Not possible for Wiener part since $\phi(y) = \sum_{i=1}^{q} b_i \phi_i(y) = \left(\mathbf{b}^\top \otimes \mathbb{I}\right) \boldsymbol{\phi}(y)$

$$\begin{cases} x_{k+1} & = \; Ax_k + \bar{B}\boldsymbol{\psi}(u_k), \\ \left(\mathbf{b}^\top \otimes \mathbb{I}\right) \boldsymbol{\phi}(y_k) & = \; Cx_k + \bar{D}\boldsymbol{\psi}(u_k), \end{cases}$$

Neither $\left(\mathbf{b}^\top \otimes \mathbb{I}\right) \in \mathbb{R}^{n_y \times n_y q}$ nor $\boldsymbol{\phi}(\cdot) : \mathbb{R}^{n_y} \to \mathbb{R}^{q n_y}$ is invertible

An alternative route

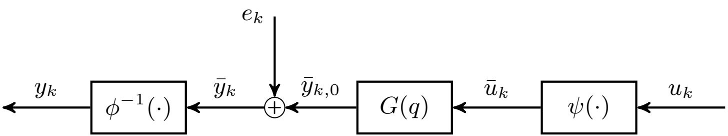


Figure: Hammerstein-Wiener systems

- Extend explicit predictor structure to H-W systems:

$$\mathbf{y}_\mathrm{f} = [\Gamma_1 \;\; \Gamma_2] \begin{bmatrix} \mathbf{u} \\ \mathbf{y}_\mathrm{p} \end{bmatrix} + [-\Gamma_2 \;\; \mathbb{I}]\,\mathbf{e} \quad \Rightarrow \quad \mathbf{0} = [\Gamma_1 \;\; \Gamma_2 \;\; -\mathbb{I}] \begin{bmatrix} \Psi(\mathbf{u}) \\ \Phi(\mathbf{y}_\mathrm{p}) \\ \Phi(\mathbf{y}_\mathrm{f}) \end{bmatrix} + [-\Gamma_2 \;\; \mathbb{I}]\,\mathbf{e}$$

$$\eta := \begin{bmatrix} \mathbf{u} \\ \mathbf{y}_\mathrm{p} \\ \mathbf{y}_\mathrm{f} \end{bmatrix}, \quad f(\eta) = [\Gamma_1 \;\; \Gamma_2 \;\; -\mathbb{I}] \begin{bmatrix} \Psi(\mathbf{u}) \\ \Phi(\mathbf{y}_\mathrm{p}) \\ \Phi(\mathbf{y}_\mathrm{f}) \end{bmatrix}, \quad \begin{cases} \psi(\cdot) \sim \mathcal{GP}\left(m_u(\cdot), k_u(\cdot, \cdot; \theta_u)\right) \\ \phi(\cdot) \sim \mathcal{GP}\left(m_y(\cdot), k_y(\cdot, \cdot; \theta_y)\right) \end{cases}$$

- Train a GP of $f(\eta)$ with inputs: $\eta_k^d = \mathrm{col}\left(u_k^d, \ldots, u_{k+L-1}^d, y_k^d, \ldots, y_{k+L-1}^d\right)$, outputs: $\chi_k^d = \mathbf{0}$
  noise covariance: $\Sigma = \mathrm{cov}\left([-\Gamma_2 \;\; \mathbb{I}]\,\mathbf{e}\right) = \sigma^2\left(\Gamma_2\Gamma_2^\top + \mathbb{I}\right)$, assuming $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbb{I})$

Gaussian process regression

- Learn unknown function

$$f : \mathbb{R}^{n_\eta} \to \mathbb{R}^{n_\chi} \sim \mathcal{GP}\left(m(\cdot), k(\cdot,\cdot;\Theta)\right)$$

from data

$$(\boldsymbol{\eta}^d, \boldsymbol{\chi}^d) = \left(\eta_k^d, \chi_k^d\right)_{k=1}^M, \quad \chi_k^d = f(\eta_k^d) + \epsilon_k, \quad \epsilon_k \sim \mathcal{N}(\mathbf{0}, \Sigma)$$

- **Joint distribution:**

$$\begin{bmatrix} f(\eta) \\ \boldsymbol{\chi}^d \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\eta) \\ \mathbf{m}(\boldsymbol{\eta}^d) \end{bmatrix}, \begin{bmatrix} k(\eta,\eta) & \mathbf{k}^\top(\boldsymbol{\eta}^d, \eta) \\ \mathbf{k}(\boldsymbol{\eta}^d, \eta) & K(\boldsymbol{\eta}^d, \boldsymbol{\eta}^d) + \bar{\Sigma} \end{bmatrix} \right), \quad \bar{\Sigma} := \mathbb{I} \otimes \Sigma$$

- **Posterior distribution:**

$$f(\eta)|\boldsymbol{\chi}^d \sim \mathcal{N}\Big(\underbrace{m_{\mathrm{p}}(\eta)}_{\text{nominal pred.}}, \underbrace{k_{\mathrm{p}}(\eta)}_{\text{pred. error}} \Big)$$

$$m_{\mathrm{p}}(\eta) := m(\eta) + \mathbf{k}^\top(\boldsymbol{\eta}^d, \eta)\left(K + \bar{\Sigma}\right)^{-1}\left(\boldsymbol{\chi}^d - \mathbf{m}(\boldsymbol{\eta}^d)\right)$$

$$k_{\mathrm{p}}(\eta) := k(\eta,\eta) - \mathbf{k}^\top(\boldsymbol{\eta}^d, \eta)\left(K + \bar{\Sigma}\right)^{-1}\mathbf{k}(\boldsymbol{\eta}^d, \eta)$$

The formulas

$$m(\eta) = \mathbb{E}\left[f(\eta) = [\Gamma_1 \ \Gamma_2 \ \ -\mathbb{I}] \begin{bmatrix} \Psi(\mathbf{u}) \\ \Phi(\mathbf{y}_\mathsf{p}) \\ \Phi(\mathbf{y}_\mathsf{f}) \end{bmatrix}\right] = [\Gamma_1 \ \Gamma_2 \ \ -\mathbb{I}] \begin{bmatrix} \mathbf{m}_u(\mathbf{u}) \\ \mathbf{m}_y(\mathbf{y}) \end{bmatrix}$$

$$k(\eta, \eta) = \operatorname{cov}\left(f(\eta) = [\Gamma_1 \ \Gamma_2 \ \ -\mathbb{I}] \begin{bmatrix} \Psi(\mathbf{u}) \\ \Phi(\mathbf{y}_\mathsf{p}) \\ \Phi(\mathbf{y}_\mathsf{f}) \end{bmatrix}\right) = [\Gamma_1 \ \Gamma_2 \ \ -\mathbb{I}] \begin{bmatrix} \mathbf{k}_u(\mathbf{u}, \mathbf{u}) & \mathbf{0} \\ \mathbf{0} & \mathbf{k}_y(\mathbf{y}, \mathbf{y}) \end{bmatrix} \begin{bmatrix} \Gamma_1^\top \\ \Gamma_2^\top \\ -\mathbb{I} \end{bmatrix}$$

$$\mathbf{m}(\boldsymbol{\eta}^d) = \operatorname{vec}\left([\Gamma_1 \ \Gamma_2 \ \ -\mathbb{I}] \begin{bmatrix} \mathbf{m}_u(H_u) \\ \mathbf{m}_y(H_y) \end{bmatrix}\right)$$

$$\mathbf{k}(\boldsymbol{\eta}^d, \eta) = (\mathbb{I} \otimes [\Gamma_1 \ \Gamma_2 \ \ -\mathbb{I}]) \begin{bmatrix} \mathbf{k}_u(\operatorname{vec}(H_u), \mathbf{u}) & \mathbf{0} \\ \mathbf{0} & \mathbf{k}_y(\operatorname{vec}(H_y), \mathbf{y}) \end{bmatrix} \begin{bmatrix} \Gamma_1^\top \\ \Gamma_2^\top \\ -\mathbb{I} \end{bmatrix}$$

$$K(\boldsymbol{\eta}^d, \boldsymbol{\eta}^d) = (\mathbb{I} \otimes [\Gamma_1 \ \Gamma_2 \ \ -\mathbb{I}]) \begin{bmatrix} \mathbf{k}_u(\operatorname{vec}(H_u), \operatorname{vec}(H_u)) & \mathbf{0} \\ \mathbf{0} & \mathbf{k}_y(\operatorname{vec}(H_y), \operatorname{vec}(H_y)) \end{bmatrix} \left(\mathbb{I} \otimes \begin{bmatrix} \Gamma_1^\top \\ \Gamma_2^\top \\ -\mathbb{I} \end{bmatrix}\right)$$

Compare to the pure black box

Pure black box

$$\eta = \begin{bmatrix} \mathbf{u} \\ \mathbf{y}_{\mathsf{p}} \end{bmatrix}, \quad \chi = \mathbf{y}_{\mathsf{f}}$$

$$f(\eta) \sim \mathcal{GP}\left(\mathbf{m}(\cdot), \mathbf{k}(\cdot, \cdot)\right)$$

Structured grey box

$$\eta = \begin{bmatrix} \mathbf{u} \\ \mathbf{y}_{\mathsf{p}} \\ \mathbf{y}_{\mathsf{f}} \end{bmatrix}, \quad \chi = \mathbf{0}$$

$$f(\eta) = \left[\Gamma_1 \;\; \Gamma_2 \;\; -\mathbb{I}\right] \begin{bmatrix} \Psi(\mathbf{u}) \\ \Phi(\mathbf{y}_{\mathsf{p}}) \\ \Phi(\mathbf{y}_{\mathsf{f}}) \end{bmatrix}$$

$$\begin{cases} \psi(\cdot) \sim \mathcal{GP}\left(m_u(\cdot), k_u(\cdot, \cdot; \theta_u)\right) \\ \phi(\cdot) \sim \mathcal{GP}\left(m_y(\cdot), k_y(\cdot, \cdot; \theta_y)\right) \end{cases}$$

- Pure black box learns everything

  $$f : \mathbb{R}^{n_u L + n_y L_0} \to \mathbb{R}^{n_y L'}$$

  as general nonlinear functions modeled by GP
- Structured grey box only learns static nonlinearities

  $$\psi : \mathbb{R}^{n_u} \to \mathbb{R}^{n_u}, \quad \phi : \mathbb{R}^{n_y} \to \mathbb{R}^{n_y}$$

  as general nonlinear functions modeled by GP
- Linearity is maintained by linear operator $\left[\Gamma_1 \;\; \Gamma_2 \;\; -\mathbb{I}\right]$
- Structured grey box learns an implicit function $\Rightarrow$ no explicit prediction

Interpretation of the implicit GP

- From posterior distribution:

$$[\Gamma_1 \;\; \Gamma_2] \begin{bmatrix} \Psi(\mathbf{u}) \\ \Phi(\mathbf{y}_\mathsf{p}) \end{bmatrix} - \Phi(\mathbf{y}_\mathsf{f}) \sim \mathcal{N}\left(m_\mathsf{p}(\eta), k_\mathsf{p}(\eta)\right) \tag{1}$$

- From model equation:

$$[\Gamma_1 \;\; \Gamma_2]\, \mathrm{col}\left(\Psi(\mathbf{u}), \Phi(\mathbf{y}_\mathsf{p})\right) - \Phi(\mathbf{y}_{\mathsf{f},0}) = [\Gamma_2 \;\; \mathbf{0}]\, \mathbf{e} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \Gamma_2 \Gamma_2^\top\right) \tag{2}$$

$\mathbf{y}_{\mathsf{f},0}$: noise-free output

- Any prediction admits projected prediction error (1)–(2)

$$\Phi(\mathbf{y}_{f,0}) - \Phi(\mathbf{y}_\mathsf{f}) \sim \mathcal{N}\left(m_\mathsf{p}(\eta), \hat{k}_\mathsf{p}(\eta)\right), \quad \hat{k}_\mathsf{p}(\eta) = k_\mathsf{p}(\eta) + \sigma^2 \Gamma_2 \Gamma_2^\top$$

- **Optimal prediction:**

$$\mathbf{y}_\mathsf{f} = \underset{\mathbf{y}_\mathsf{f}}{\mathrm{argmin}}\; \mathrm{tr}\left(\hat{k}_\mathsf{p}(\eta)\right) + m_\mathsf{p}^\top(\eta) m_\mathsf{p}(\eta) \tag{Min-MSE}$$

Price to pay: linear model estimated as hyperparameters

- In addition to standard hyperparameters θ_u, θ_y, σ^2, linear model parameters Γ_1, Γ_2 also become hyperparameters
- Standard maximum marginal likelihood:

$$\begin{aligned} \Theta &= \operatorname*{argmax}_{\Theta} p\left(\boldsymbol{\chi}^d | \boldsymbol{\eta}^d, \Theta\right) && \Theta := \left\{\theta_u, \theta_y, \sigma^2, \Gamma_1, \Gamma_2\right\} \\ &= \operatorname*{argmin}_{\Theta} \operatorname{logdet}\left(\Lambda(\Theta)\right) + \mathbf{m}^\top(\boldsymbol{\eta}^d)\Lambda^{-1}(\Theta)\mathbf{m}(\boldsymbol{\eta}^d), && \Lambda(\Theta) := K(\boldsymbol{\eta}^d, \boldsymbol{\eta}^d; \Theta) + \bar{\Sigma} \end{aligned}$$

  is prone to overfitting due to the high dimensionality of Θ, $[\Gamma_1 \; \Gamma_2] \in \mathbb{R}^{n_y L' \times (n_u L + n_y L_0)}$

- **Solution:** consider Γ_1, Γ_2 as random variables admitting a hyperprior
  $\Rightarrow$ a joint maximum-a-posteriori/maximum likelihood (JMAP-ML) problem:

$$\begin{aligned} \Theta &= \operatorname*{argmax}_{\Theta} p\left(\boldsymbol{\chi}^d | \boldsymbol{\eta}^d, \boldsymbol{\theta}, \Gamma\right) p(\Gamma|\zeta) && \boldsymbol{\theta} := \operatorname{col}\left(\theta_u, \theta_y, \sigma^2\right), \; \Gamma := [\Gamma_1 \; \Gamma_2] \\ &= \operatorname*{argmin}_{\Theta} \operatorname{logdet}\left(\Lambda(\Theta)\right) + \mathbf{m}^\top(\boldsymbol{\eta}^d)\Lambda^{-1}(\Theta)\mathbf{m}(\boldsymbol{\eta}^d) + \log p(\Gamma|\zeta) \end{aligned}$$

  ζ : parameters on the hyperprior ('hyper-hyperparameters')

Intermezzo: stable kernel design

- Consider a stable linear system given by its impulse response

$$y_t = \int_0^\infty g(\tau)u(t-\tau)\mathrm{d}\tau$$

- To learn impulse response $g(\tau)$ by GP, what would be a good prior kernel?
- Required property: asymptotic stability, i.e., $g(\tau) \le \lambda\alpha^\tau$, $\alpha \in (0,1)$ with high probability $\Rightarrow$ stable kernels
- Tune/correlated (TC) kernel:

$$s^{\mathsf{TC}}(\tau,\tau';\zeta) := \lambda\alpha^{\max(\tau,\tau')}$$

$$\zeta := [\lambda\ \alpha]^\top,\ \lambda > 0,\ \alpha \in (0,1)$$

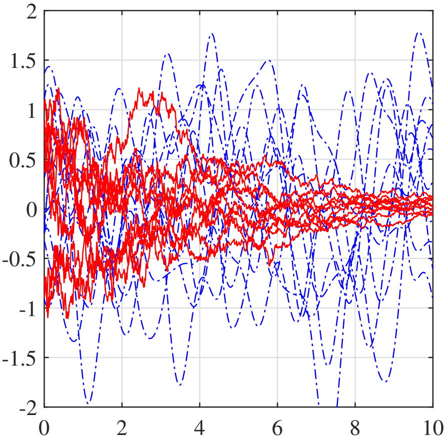


Figure: red: samples of TC kernel, blue: samples of SE kernel

Hyperprior design I

- Consider the noise-free relation of the linear part and further decompose Γ:

$$\bar{\mathbf{y}}_{\mathsf{f},0} = [\Gamma_{11}\ \Gamma_{12}\ \Gamma_{2}] \begin{bmatrix} \bar{\mathbf{u}}_{\mathsf{p}} \\ \bar{\mathbf{u}}_{\mathsf{f}} \\ \bar{\mathbf{y}}_{\mathsf{p},0} \end{bmatrix}$$

- Following a similar idea, each row of Γ_{11} and Γ_{2} is modeled with a TC kernel indexed by the input-output lag

$$\operatorname{cov}\left((\Gamma_{11})_{i,j}\,,(\Gamma_{11})_{i',j'}\right) = \begin{cases} 0, & i \neq i' \\ s^{\mathsf{TC}}_{L_0+i-j,L_0+i'-j'}, & \text{otherwise} \end{cases}, \quad s^{\mathsf{TC}}_{i,j} := \lambda\alpha^{\max(i,j)}$$

Hyperprior design II

- Note that Γ_{12} is a Toeplitz matrix of Markov parameters

$$\Gamma_{12} = \begin{bmatrix} D & & & \\ CB & D & & \\ \vdots & \vdots & \ddots & \\ CA^{L'-2}B & CA^{L'-3}B & \dots & D \end{bmatrix}$$

  the last row γ_{12} contains all independent elements

$$\text{cov}\left(\left(\gamma_{12}\right)_j, \left(\gamma_{12}\right)_{j'}\right) = s^{\mathsf{TC}}_{L'-j, L'-j'}$$

- Vectorizing all the independent elements of Γ, define $\boldsymbol{\gamma} := \text{col}\left(\text{vec}\left(\Gamma_{11}^\top\right), \text{vec}\left(\Gamma_2^\top\right), \gamma_{12}^\top\right)$
- The hyperprior of Γ can be expressed as $\boldsymbol{\gamma}|\zeta \sim \mathcal{N}(\mathbf{0}, S_{\boldsymbol{\gamma}})$
- The JMAP-ML problem becomes:

$$\bar{\Theta} = \underset{\bar{\Theta}}{\text{argmin}} \; \text{logdet}\left(\Lambda(\bar{\Theta})\right) + \mathbf{m}^\top(\boldsymbol{\eta}^d)\Lambda^{-1}(\bar{\Theta})\mathbf{m}(\boldsymbol{\eta}^d) + \boldsymbol{\gamma}^\top S_{\boldsymbol{\gamma}}^{-1}\boldsymbol{\gamma}, \quad \bar{\Theta} := \{\boldsymbol{\theta}, \boldsymbol{\gamma}\}$$

- Nonlinearities $\Psi(\mathbf{u})$, $\Phi(\mathbf{y})$ can also be learned using very similar GP regression procedures

- **Joint distribution:**

$$\begin{bmatrix} \phi(\mathbf{y}) \\ \boldsymbol{\chi}^d \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m}_y(\mathbf{y}) \\ \mathbf{m}(\boldsymbol{\eta}^d) \end{bmatrix}, \begin{bmatrix} \mathbf{k}_y(\mathbf{y},\mathbf{y}) & \boldsymbol{\kappa}_y^\top(\boldsymbol{\eta}^d,\mathbf{y}) \\ \boldsymbol{\kappa}_y(\boldsymbol{\eta}^d,\mathbf{y}) & K(\boldsymbol{\eta}^d,\boldsymbol{\eta}^d) + \bar{\Sigma} \end{bmatrix} \right)$$

$$\boldsymbol{\kappa}_y(\boldsymbol{\eta}^d,\mathbf{y}) := (\mathbb{I} \otimes [\Gamma_2 \;\; -\mathbb{I}])\, \mathbf{k}_y(\mathrm{vec}\,(H_y)\,, \mathbf{y})$$

- **Posterior distribution:** $\Phi(\mathbf{y})|\boldsymbol{\chi}^d \sim \mathcal{N}\left(\mathbf{m}_{y\mathsf{p}}(\mathbf{y}), \mathbf{k}_{y\mathsf{p}}(\mathbf{y})\right)$

$$\mathbf{m}_{y\mathsf{p}}(\mathbf{y}) := \mathbf{m}_y(\mathbf{y}) - \boldsymbol{\kappa}_y^\top(\boldsymbol{\eta}^d,\mathbf{y}) \left(K + \bar{\Sigma}\right)^{-1} \mathbf{m}(\boldsymbol{\eta}^d)$$

$$\mathbf{k}_{y\mathsf{p}}(\mathbf{y}) := \mathbf{k}_y(\mathbf{y},\mathbf{y}) - \boldsymbol{\kappa}_y^\top(\boldsymbol{\eta}^d,\mathbf{y}) \left(K + \bar{\Sigma}\right)^{-1} \boldsymbol{\kappa}_y(\boldsymbol{\eta}^d,\mathbf{y})$$

From prediction to predictive control

- Receding horizon control with expected control cost:

$$J_{\text{ctr}} := \left\|\mathbf{u}^k\right\|_R^2 + \mathbb{E}\left[\left\|\Phi(\mathbf{y}_0^k) - \Phi(\mathbf{r}^k)\right\|_Q^2\right], \ \mathbf{r}^k := \begin{bmatrix} r_k \\ \vdots \\ r_{k+L'-1} \end{bmatrix}, \ \mathbf{u}^k := \begin{bmatrix} u_k \\ \vdots \\ u_{k+L'-1} \end{bmatrix}, \ \mathbf{y}_0^k := \begin{bmatrix} y_{k,0} \\ \vdots \\ y_{k+L'-1,0} \end{bmatrix}$$

- The distribution of $\Phi(\mathbf{y}_0^k)$ is given by the prediction error quantification

$$\Phi(\mathbf{y}_0^k) \sim \mathcal{N}\left(\Phi(\mathbf{y}^k) + m_{\text{p}}(\eta^k), \hat{k}_{\text{p}}(\eta^k)\right),$$

- The expected control cost becomes:

$$J_{\text{ctr}} = \left\|\mathbf{u}^k\right\|_R^2 + \left\|\Phi(\mathbf{y}^k) + m_{\text{p}}(\eta^k) - \Phi(\mathbf{r}^k)\right\|_Q^2 + \operatorname{tr}\left(Q\hat{k}_{\text{p}}(\eta^k)\right)$$

- Unknown nonlinear function values $\Phi(\mathbf{y}^k)$, $\Phi(\mathbf{r}^k)$ are replaced by their posterior estimates $\mathbf{m}_{y\text{p}}(\mathbf{y}^k)$, $\mathbf{m}_{y\text{p}}(\mathbf{r}^k)$

Institute of Automatic Control
Leibniz University Hannover

Chance constraint satisfaction I

- Output chance constraints $\Pr\left(H\mathbf{y}_0^k \leq q\right) \geq p$ can also be enforced

Lemma: Chance constraint satisfaction

Suppose the output nonlinearity $\phi^{-1}(\cdot)$ is Lipschitz continuous with

$$\left\|\phi^{-1}(\bar{y}_1) - \phi^{-1}(\bar{y}_2)\right\|_2 \leq M \left\|\bar{y}_1 - \bar{y}_2\right\|_2 .$$

The chance constraint $\Pr\left(H\mathbf{y}_0^k \leq q\right) \geq p$ is guaranteed by

$$H\mathbf{y}^k \leq q - c_p\sqrt{\operatorname{diag}\left(HH^\top\right)},$$

where $c_p := M\left(\sqrt{\mu(p)\sigma_{\mathrm{p}}(\eta^k)} + \left\|m_{\mathrm{p}}(\eta^k)\right\|_2\right)$, $\sigma_{\mathrm{p}}(\eta^k)$ is the largest eigenvalue of $\hat{k}_{\mathrm{p}}(\eta^k)$ and $\mu(\cdot)$ is the quantile function of the χ^2-distribution with $n_y L'$ degrees of freedom.

Chance constraint satisfaction II

Sketch of proof: Since $\Phi(\mathbf{y}_0^k) \sim \mathcal{N}\left(\Phi(\mathbf{y}^k) + m_\mathrm{p}(\eta^k), \hat{k}_\mathrm{p}(\eta^k)\right)$,

$$\Pr\left(\left\|\Phi(\mathbf{y}_0^k) - \Phi(\mathbf{y}^k) - m_\mathrm{p}(\eta^k)\right\|^2_{\hat{k}_\mathrm{p}^{-1}(\eta^k)} \leq \mu(p)\right) = p$$

Lower-bound $\left\|\Phi(\mathbf{y}_0^k) - \Phi(\mathbf{y}^k) - m_\mathrm{p}(\eta^k)\right\|^2_{\hat{k}_\mathrm{p}^{-1}(\eta^k)}$:

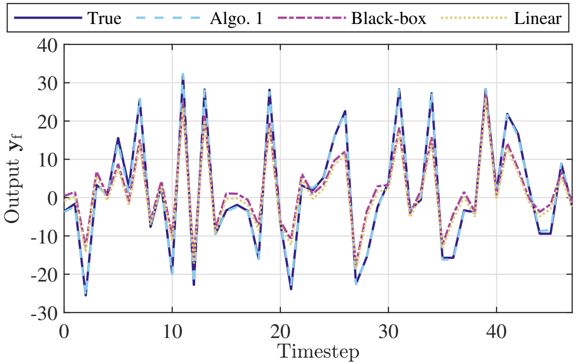
$$\begin{aligned}
&\left\|\Phi(\mathbf{y}_0^k) - \Phi(\mathbf{y}^k) - m_\mathrm{p}(\eta^k)\right\|_{\hat{k}_\mathrm{p}^{-1}(\eta^k)} \\
\geq\ & \frac{1}{\sqrt{\sigma_\mathrm{p}(\eta^k)}} \left\|\Phi(\mathbf{y}_0^k) - \Phi(\mathbf{y}^k) - m_\mathrm{p}(\eta^k)\right\|_2 \\
\geq\ & \frac{1}{\sqrt{\sigma_\mathrm{p}(\eta^k)}} \left(\left\|\Phi(\mathbf{y}_0^k) - \Phi(\mathbf{y}^k)\right\|_2 - \left\|m_\mathrm{p}(\eta^k)\right\|_2\right) \\
\geq\ & \frac{1}{M\sqrt{\sigma_\mathrm{p}(\eta^k)}} \left\|\mathbf{y}_0^k - \mathbf{y}^k\right\|_2^2 - \frac{1}{\sqrt{\sigma_\mathrm{p}(\eta^k)}} \left\|m_\mathrm{p}(\eta^k)\right\|_2 .
\end{aligned}$$

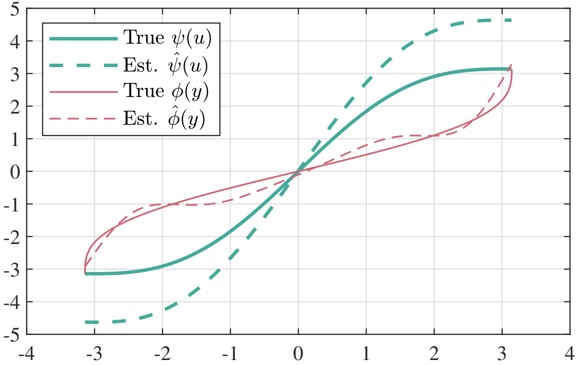
So $\left\|\mathbf{y}_0^k - \mathbf{y}^k\right\|_2 \leq c_p$ is satisfied with at least probability p. Then, the usual robust control argument to derive disturbance invariant sets follows.

Numerical examples

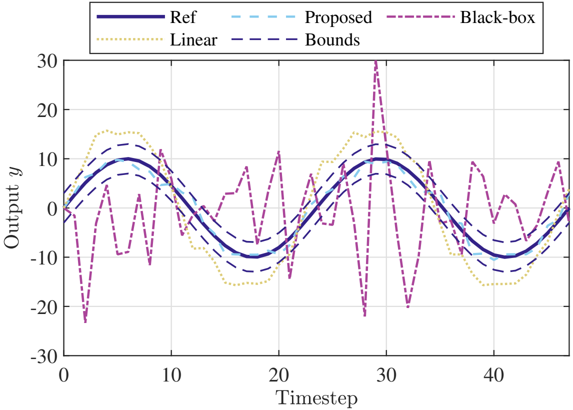
- Random SISO second-order linear systems for $G(q)$
- $\psi(u) = u + \sin(u)$, $\phi^{-1}(\bar{y}) = \bar{y} + \sin(\bar{y})$ (Lipschitz with $M = 2$)
- $L_0 = 2$, $N = 100$, $\sigma = 0.01$, $p = 0.7$, $Q = R = \mathbb{I}$
- Hyper-hyperparameters $\zeta = [\lambda \ \alpha]^\top$ selected by cross-validation with grid $\lambda \in \{0.25, 0.5, 1, 2, 4\}$, $\alpha \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$
- Squared exponential kernel for $k_u(\cdot,\cdot)$ and $k_y(\cdot,\cdot)$
- Identity prior mean functions, i.e., $m_u(u) = u$, $m_y(y) = y$
- **Benchmark:** black-box GP model, linear predictor ignoring nonlinearities

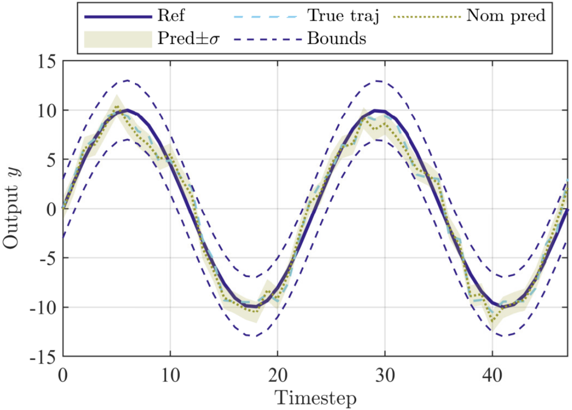
Prediction results





Control results





Institute of Automatic Control
Leibniz University Hannover