System Identification via Frequency-Domain Gaussian Process Regression for Transfer-Function Estimation

Atsushi Yanagisawa   Kentaro Ohki

Kyoto University

SICE ISCS 2026

Outline

1. Introduction
2. Proposed Method
3. Experimental Results
4. Discussion & Conclusion

Motivation

System identification: constructs models from input–output data

Parametric vs. Nonparametric

- Parametric: model structure selection is difficult
- Nonparametric: avoids this problem

Why Gaussian Process Regression?

- No model structure selection required, and provides **uncertainty quantification**

The Problem: $O(n^3)$ Complexity

- Due to high computational cost Gaussian Process Regression cannot be directly applied to raw input–output data

Main Idea — Frequency-Domain Compression

Time Domain

- Accurate FRF estimation requires $n \sim 10^5$–10^7 samples
- Gaussian Process Regression cost: $O(n^3)$ prohibitive

$$u(t), y(t) \xrightarrow{\text{Fourier}} \hat{G}(j\omega_k)$$

Frequency Domain

- Only $N_d \sim 50$–100 FRF points
- Gaussian Process Regression cost: $O(N_d^3)$ tractable
- Dynamics fully preserved

$$O(n^3) \longrightarrow O(N_d^3)$$

Key Insight

Fourier transform compresses data into a compact Frequency Response Function (FRF), making Gaussian Process Regression **practical** for long recordings.

Contributions

Contributions

1. **Frequency-domain Gaussian Process Regression framework** — enables practical Gaussian Process Regression-based system identification even for long-duration observation data by operating on ~ 100 FRF samples instead of 10^5–10^7 time-domain points
2. **Real-hardware validation** — experimentally validated on a Quanser Rotary Flexible Link, demonstrating applicability beyond simulation
3. **Comprehensive kernel comparison** — compared 11 kernel functions plus conventional methods (LS / NLS), showing that Gaussian Process Regression can achieve accuracy comparable to parametric methods

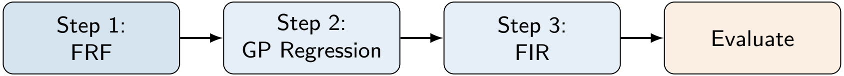
Proposed Method

1. Introduction
2. Proposed Method
3. Experimental Results
4. Discussion & Conclusion

Method Overview — Three Steps

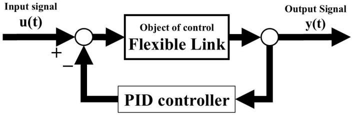
Three-Step Pipeline

1. **Frequency Response Function Estimation** — Estimate discrete $\hat{G}(j\omega_k)$ from input–output data
2. **Gaussian Process Regression Interpolation** — Apply Gaussian Process Regression to interpolate a continuous Frequency Response Function
3. **Finite Impulse Response Reconstruction** — Reconstruct a Finite Impulse Response model via inverse Fourier transform



Experimental Setup

Quanser Rotary Flexible Link



Closed-loop block diagram

Training: Multisine

$$u(t) = \sum_{k=0}^{N_d-1} a_k \sin(2\pi f_k t + \phi_k)$$

f_k: log-spaced in $[0.1, 250]$ Hz, $a_k \sim \mathcal{U}(0, 20/N_d)$, $\phi_k \sim \mathcal{U}(0, 2\pi)$

Validation: Square wave

Random amplitude, period, duty ratio

Step 1 — Frequency Response Function (FRF) Estimation

Log-spaced frequency grid:

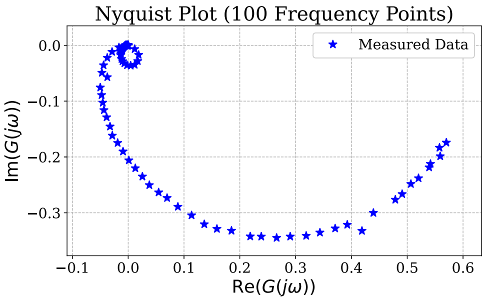
$$f_k = 10^{\log_{10}(f_{\min}) + k\,\Delta_f\,/\,N_d}, \quad k = 0, \ldots, N_d - 1$$

Complex Fourier coefficient:

$$C_x(\omega_k) = \frac{2}{T} \int_{t_0+\tau_{\text{drop}}}^{t_1} \big(x(t) - \bar{x}\big)\, e^{-j\omega_k t}\, dt$$

Discrete FRF estimate $(\omega_k = 2\pi f_k)$:

$$\hat{G}(j\omega_k) = \frac{C_y(\omega_k)}{C_u(\omega_k)}$$



Nyquist plot of $\hat{G}(j\omega_k)$, $N_d = 100$

Notation:

$\bar{x}$: time-domain mean

T: effective interval

$\tau_{\text{drop}} = 0.02\,\text{s}$: transient drop

Step 2 — Gaussian Process Regression for Frequency Response

Assumption: Model real and imaginary parts as **independent** Gaussian processes:

$$x_i = \log_{10}(\omega_i), \qquad y_i \in \{\mathrm{Re}\{\hat{G}\},\ \mathrm{Im}\{\hat{G}\}\}$$

Gaussian Process Regression Predictive Mean

$$\hat{m}(x_*) = \boldsymbol{k}(x_*)^\top \left(\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}\right)^{-1} \boldsymbol{y}$$

x_*: test input (log-frequency), $\boldsymbol{k}(x_*)$: kernel vector between x_* and training inputs, $\boldsymbol{K}$: kernel matrix, σ_n^2: noise variance, $\boldsymbol{y}$: training outputs

- Hyperparameters $\boldsymbol{\theta} = \{\sigma_f^2, \ell, \sigma_n^2, \dots\}$ optimized via Grid Search

Kernel Functions (11 Tested)

Kernel	Expression	Character
RBF	$k(x,x') = \sigma_f^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$	Infinitely smooth
Matérn-5/2	$k = \sigma_f^2\left(1 + \frac{\sqrt{5}\,d}{\ell} + \frac{5d^2}{3\ell^2}\right)\exp\left(-\frac{\sqrt{5}\,d}{\ell}\right)$	Finite smoothness (C^2)
Matérn-3/2	$k = \sigma_f^2\left(1 + \frac{\sqrt{3}\,d}{\ell}\right)\exp\left(-\frac{\sqrt{3}\,d}{\ell}\right)$	Once differentiable (C^1)
DI	$k = \beta\,\alpha^x$ if $x = x'$; 0 otherwise	Sparse-data specialist
SS1	$k = \sigma_f^2 \exp\left(-\beta \cdot \min(x,x')\right)$	Stability prior

Notation

σ_f^2: signal variance

ℓ: lengthscale

$d = |x - x'|$: input distance

α, β: kernel parameters

C^n: n-times diff.

Smoothness & Convergence

RBF (C^∞): logarithmic convergence unless target is analytic

Matérn-5/2 (C^2): finite smoothness

Additional Kernels

Matérn-1/2, Exponential, DC, SS2, SSHF, Stable Spline also tested (11 total).

FIR Model Construction

From Gaussian Process Regression to Impulse Response

GP prediction → Uniform grid → Hermitian ext. → IDFT

Step A: Impulse response via Inverse Discrete Fourier Transform (IDFT)

$$h_k = \mathrm{Re}\left\{ \frac{1}{M} \sum_{n=0}^{M-1} X_n e^{\frac{j2\pi kn}{M}} \right\}$$

Hermitian: $X_{M-n} = X_n^*,\ \ M = 2N_d - 1$

FIR order: $N = \min(M, 1024)$

Step B: Finite Impulse Response (FIR) prediction (convolution)

$$\hat{y}(t) = \sum_{k=0}^{N-1} h_k\, u(t - k\Delta t)$$

$\Delta t = 0.002\,\mathrm{s}$ (500 Hz sampling)

Baselines: Rational function $G(j\omega) = \frac{\sum_{i=0}^{n_b} \beta_i (j\omega)^i}{\sum_{j=0}^{n_a} \alpha_j (j\omega)^j},\ \ n_b = 2,\ \ n_a = 4$

- LS — Least Squares
- NLS — Nonlinear Least Squares

Experimental Results

1 Introduction

2 Proposed Method

3 Experimental Results

4 Discussion & Conclusion

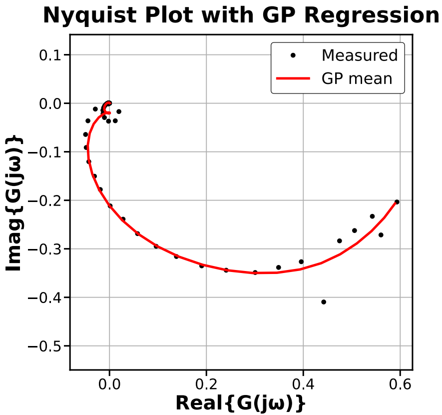
Baseline Results ($N_d = 50$, $T = 1$ hr)

Kernel / Method	Multisine	Square
Matérn-5/2	**2.90**	**5.89**
Matérn-3/2	2.94	6.04
RBF	3.05	5.96
SS1 (Stable Spline 1)	3.01	5.97
DI (Diagonal)	6.92	15.1
LS (Least Squares)	9.79	26.9
NLS (Nonlinear LS)	**2.75**	**5.77**

RMSE $\times 10^{-2}$ [rad]

Key Result

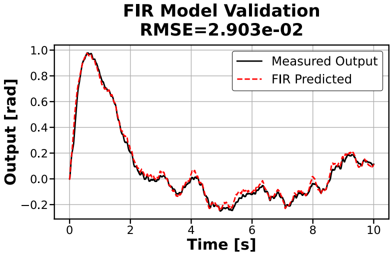
Matérn-5/2 achieves best Gaussian Process Regression accuracy (RMSE = 0.0290), close to NLS (0.0275) — **without** specifying model structure.



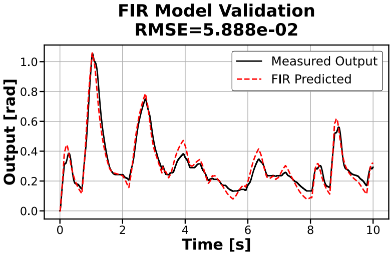
Gaussian Process Regression interpolation on Nyquist plot

(Matérn-5/2, $N_d = 50$)

Time-Series Predictions (Matérn-5/2)



Multisine input — predicted vs. actual



Square-wave input — predicted vs. actual

Observation

The FIR model reconstructed from Gaussian Process Regression accurately tracks both the training signal (multisine) and a completely different validation signal (square wave).

Effect of Frequency Points N_d ($T = 60$ min)

Method	$N_d = 10$	$N_d = 30$	$N_d = 50$	$N_d = 100$
DI	**7.93**	**9.92**	6.92	6.75
Matérn-5/2	9.79	18.0	**2.90**	2.46
RBF	9.16	18.0	3.05	2.47
SS1	9.13	22.4	3.01	**2.45**
NLS	9.40	14.5	**2.75**	**2.35**

RMSE $\times 10^{-2}$ [rad] (multisine)

Sparse Data ($N_d \leq 30$)

DI kernel excels — diagonal structure avoids over-extrapolation.

Dense Data ($N_d \geq 50$)

Matérn-5/2 best among Gaussian Process Regression kernels.

Discussion & Conclusion

1. Introduction
2. Proposed Method
3. Experimental Results
4. Discussion & Conclusion

Discussion

Method	$N_d = 10$	$N_d = 30$	$N_d = 50$
DI	**7.93**	**9.92**	6.92
Matérn-5/2	9.79	18.0	**2.90**
RBF	9.16	18.0	3.05
NLS	9.40	14.5	**2.75**

RMSE $\times 10^{-2}$ [rad] (multisine, $T = 60\,\text{min}$)

- Matérn-5/2's C^2 smoothness matches physical systems better than the infinitely smooth RBF
- Gaussian Process Regression achieves accuracy comparable to NLS **without** requiring a parametric model structure

Limitations & Future Work

Current Limitations

- Simple Single-Input Single-Output apparatus — Multi-Input Multi-Output and nonlinear systems not yet validated
- Independent Re/Im modeling prevents stability/causality constraints
- Posterior uncertainty not yet exploited in controller design

Future Directions

- Complex-valued Gaussian Process Regression — joint Re/Im modeling with stability/causality constraints
- Multi-Input Multi-Output extension — multi-input multi-output systems
- Uncertainty-aware robust controller synthesis

Summary

1. **Problem:** Gaussian Process Regression's $O(n^3)$ complexity prevents direct use in system identification with large time-domain datasets
2. **Approach:** Frequency-domain compression reduces data from $\sim 10^5$ time samples to ~ 100 FRF points, making Gaussian Process Regression tractable
3. **Results:** Matérn-5/2 achieves best Gaussian Process Regression accuracy (RMSE = 0.0290), closely matching NLS (0.0275) — *without* specifying a parametric model

Take-Home Message

A simple Fourier transform unlocks Gaussian Process Regression for system identification — achieving parametric-level accuracy **without** model structure selection.

Thank You — Questions & Discussion

Introduction

3 Motivation — Gaussian Process Regression's Challenge

4 Frequency-Domain Compression

5 Contributions

Proposed Method

7 Method Overview (3 Steps)

8 Experimental Setup

9 FRF Estimation

10 Gaussian Process Regression for Frequency Response

11 Kernel Functions

12 FIR Model Construction

Experimental Results

14 Baseline Results ($N_d = 50$, 1 hr)

15 Time-Series Predictions

16 Effect of Frequency Points

Discussion & Conclusion

19 Kernel Insights

20 Limitations & Future Work

21 Summary

Backup: Experimental Configuration

Plant & Controller

Param.	Value
Plant	Flexible Link
Controller	P-ctrl, $K_p = 1.65$
Δt	0.002 s (500 Hz)
τ_{drop}	0.02 s

Data Conditions

Param.	Value
Freq. range	$[0.1,\ 250]$ Hz
N_d	$\{10,\ 30,\ 50,\ 100\}$
T	$\{10,\ 30,\ 60,\ 600\}$ min
Training	Multisine
Validation	Square wave

Backup: Complete Kernel Formulas (All 11)

#	Kernel	Expression
1	RBF	$k = \sigma_f^2 \exp\!\big(-\frac{(x-x')^2}{2\ell^2}\big)$
2	Matérn-1/2	$k = \sigma_f^2 \exp\!\big(-\frac{\lvert x-x'\rvert}{\ell}\big)$
3	Matérn-3/2	$k = \sigma_f^2\big(1 + \frac{\sqrt{3}\,\lvert d\rvert}{\ell}\big) \exp\!\big(-\frac{\sqrt{3}\,\lvert d\rvert}{\ell}\big)$
4	Matérn-5/2	$k = \sigma_f^2\big(1 + \frac{\sqrt{5}\,\lvert d\rvert}{\ell} + \frac{5d^2}{3\ell^2}\big) \exp\!\big(-\frac{\sqrt{5}\,\lvert d\rvert}{\ell}\big)$
5	Exponential	$k = \sigma_f^2 H(x)H(x') \exp\!\big(-\omega(x+x')\big)$
6	DC	$k = \beta\,\alpha^{(x+x')/2}\,\rho^{\lvert x-x'\rvert}$
7	DI	$k = \beta\,\alpha^{x}$ if $x = x'$; 0 otherwise
8	SS1	$k = \sigma_f^2 \exp\!\big(-\beta \cdot \min(x,x')\big)$
9	SS2	$k = \sigma_f^2\big[\frac{1}{2}e^{-\beta(x+x'+\max)} - \frac{1}{6}e^{-3\beta\cdot\max}\big]$
10	SSHF	$k = \sigma_f^2(-1)^{x+x'} \max\!\big(e^{-\beta x}, e^{-\beta x'}\big)$
11	Stable Spline	$k = \sigma_f^2 \cdot \frac{1}{2}r^2\big(R - \frac{r}{3}\big),\;\; r=\min,\; R=\max$

Notes

Kernels 5–11 encode system-theoretic priors (stability, sparsity, causality).
Kernels 1–4 are general-purpose stationary kernels with varying smoothness.

Backup: Full Results Table — All 11 Kernels + Baselines

Method	**Multisine**	**Square**
DC	7.54	15.8
DI	6.92	15.1
Exponential	16.7	36.2
Matérn-1/2	3.01	5.97
Matérn-3/2	2.94	6.04
Matérn-5/2	**2.90**	**5.89**
RBF	3.05	5.96
SS1	3.01	5.97
SS2	5.59	8.22
SSHF	3.44	6.31
Stable Spline	6.05	9.93
LS	9.79	26.9
NLS	**2.75**	**5.77**

RMSE $\times 10^{-2}$ [rad]

Complete Ranking

NLS $>$ Matérn-5/2 $>$ Matérn-3/2 $>$ SS1 $\approx$ Matérn-1/2 $>$ RBF $>$ SSHF $>$ SS2 $>$ Stable Spline $>$ DI $>$ DC $>$ LS $\gg$ Exponential

Backup: Full Frequency Points Table

Method	$N_d = 10$	$N_d = 30$	$N_d = 50$	$N_d = 100$
DI	**7.93**	**9.92**	6.92	6.75
Matérn-3/2	8.57	18.1	2.94	4.29
Matérn-5/2	9.79	18.0	**2.90**	2.46
RBF	9.16	18.0	3.05	2.47
SS1	9.13	22.4	3.01	**2.45**
SSHF	11.1	24.3	3.44	2.73
NLS	9.40	14.5	**2.75**	**2.35**

RMSE $\times 10^{-2}$ [rad] (multisine)

Sparse Regime

$N_d \leq 30$: **DI** best — diagonal structure avoids over-extrapolation between distant frequency points.

Dense Regime

$N_d \geq 50$: Matérn-5/2 best Gaussian Process Regression kernel; NLS best overall with model structure knowledge.

Backup: Hyperparameter Optimization Details

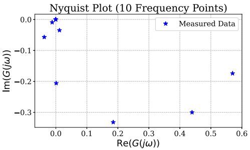
Grid Search Configuration

- **Search grid:** 30 logarithmically-spaced points in $[10^{-3}, 10^{3}]$ per hyperparameter
- **Validation data:** $N_{\text{val}} = 150$ frequency points from an independent measurement session
- **Objective:** Minimize RMSE on validation FRF
- **Real / Imaginary:** Optimized independently for each part
- **Combinatorial limit:** When total combinations > 5000, use random sampling (fixed seed for reproducibility)

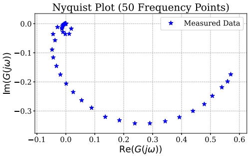
Grid Search → Validate on N_{val} → Select $\boldsymbol{\theta}^*$

Backup: FRF Estimation — Sparse & Short Data

Effect of Frequency Points



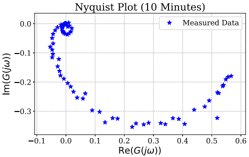
(a) $N_d = 10$



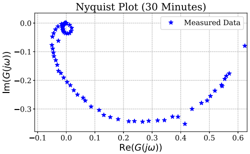
(b) $N_d = 50$

Nyquist plots: sparse vs. moderate sampling

Effect of Observation Duration



(c) $T = 10\,\text{min}$



(d) $T = 30\,\text{min}$

Nyquist plots: short observation durations

Observation

Sparse sampling ($N_d = 10$) and short duration ($T = 10\,\text{min}$) both increase noise contamination in the FRF estimate, explaining the performance degradation of smooth kernels under these conditions.

Backup: Complex-Valued Gaussian Process Regression — Future Direction

Current Approach

- Independent GP models for $\mathrm{Re}\{\hat{G}\}$ and $\mathrm{Im}\{\hat{G}\}$
- Simple but cannot encode stability / causality constraints

Limitation

- A stable, causal transfer function lies in Hardy space H_2
- Independent Re/Im modeling allows physically invalid solutions

Future Direction

Complex-valued Gaussian Process Regression using Hardy-space RKHS:

- Joint Re/Im modeling
- Kernel encodes stability via Laplace transform isomorphism
- Causality constraint built into the reproducing kernel