Prediction horizon and cable-length generalisation in machine-learning models of tower crane payload sway

Kutyrev Ratmir

Anti-sway control of tower cranes requires knowing where the payload will be based on the input data. This paper measures how far ahead payload sway can be predicted from its own recent history, using an open experimental dataset recorded on a laboratory Inteco tower crane. Five methods, from a zero forecast to an ensemble of feedforward networks, are compared across prediction horizons from 0.01 s to 1.0 s and across three families of crane motion, with generalisation to unseen cable lengths using leave-one-scenario-out validation. The results are as follows. One-step prediction is degenerate: linear regression on three lags reaches 0.0796°, below the 0.0879° encoder resolution, so no comparison of methods is useful at that horizon. A linear model remains adequate to roughly 0.07 s and becomes worse than predicting zero sway near 0.85 s. The dominant error at long horizons is a phase error: oscillation amplitude is reproduced to between 79% and 93% while the predicted waveform lags the true one by 10% to 17% of a period. Only the network ensemble stops degrading, having a normalised error of 0.416 between 0.5 s and 1.0 s, against 0.612 for the trees and 0.883 for linear regression. Cable lengths outside the training range worsen the predictions in every method, and the loss grows with horizon: at 1.0 s the linear model is worse than the zero forecast on extrapolation folds (1.304) and the tree ensemble loses 65%, because past its last split threshold a tree simply returns the value of the longest cable it was trained on. How far an unseen cable lies from the training range is better measured in natural frequency than in metres. The two ends of the trolley range are separated by similar distances in metres, but their natural frequencies differ by a factor of 4.8. In both families tested, predictions are more difficult at the short-cable end. The accuracy is 1.4 to 2.8 times worse while the crane is under command than during free decay, which is the regime in which a controller actually has to operate. The free-oscillation period produces an effective pendulum length $L_{\text{eff}} = 0.924\,L + 0.166$, reproduced across five measurements, including three excitation types and two axes.

I. INTRODUCTION

Construction cranes have become an integral part of the modern world. Their ability to move quickly and lift heavy loads makes them indispensable in construction and transportation. When using a crane, suspended loads inevitably sway. Unexpected sway slows down the positioning process, complicates automation, and poses a safety hazard on construction and industrial sites. Since the loads are suspended by a flexible cable and are not directly driven, their sway can only be controlled indirectly, through the movement of the trolley and the jib [1, 2].

Most work in this field is control-side. Input shaping filters the reference command so that the excitation it produces cancels itself, and has been demonstrated on rotary and tower crane rigs in several forms [3–5]. More recent schemes combine an adaptive shaper with feedback: a neural-network zero-vibration-derivative shaper coupled to fuzzy-logic feedback control reduces mean absolute sway on this rig from 9.28° to 2.14° on the x axis and from 8.02° to 1.12° on the y axis [6], and a composite super-twisting controller achieves comparable residual sway under parameter uncertainty [7].

These approaches all rely on knowing the system's natural frequency, which depends on the cable length. The feedforward controller then needs to predict how the payload will move. The prediction step is typically provided by a crane dynamics model, and its accuracy is not reported separately from the accuracy of the controller built on it. How well payload fluctuations can be predicted based on measurements alone, and over what time horizon, is therefore not established, even on a rig, which has been studied for years.

An open dataset of tower crane trajectories and payload oscillations was recently published for exactly this purpose [8], covering 33 operating scenarios with varied cable lengths, payload masses and shapes, and combinations of trolley motion, jib rotation, and hoisting. Its metadata record lists no related research article, and to our knowledge no machine-learning analysis of it has been published.

This paper studies how far ahead payload oscillations can be predicted from recent angular history and recorded rope length. We also test whether the models can generalise to rope lengths that were not present during training.

We compare five methods across forecast horizons from 0.01 s to 1.0 s and three families of crane motions, and evaluate their generalisation ability by considering entire scenarios rather than random samples.

This study focuses on three main findings. We first examine how prediction accuracy changes with forecast horizon. We then investigate how cable length and effective pendulum length affect generalisation. Finally, we compare the behaviour of the models across different motion scenarios and examine where their predictions become unreliable.

II. METHODS

A. The dataset

The dataset used in this study [8] was collected on an Inteco tower crane, a laboratory-scaled model of an industrial tower crane [9]. The platform has been used in several previous crane-control studies [3–6], which allows comparison with existing work. The complete installation consists of the crane itself, measuring $1200 \times 1200 \times 1500$ mm, a power interface, a data acquisition unit, and an emergency switch, with the entire setup linked to MATLAB/Simulink for real-time logging (Fig. 1).

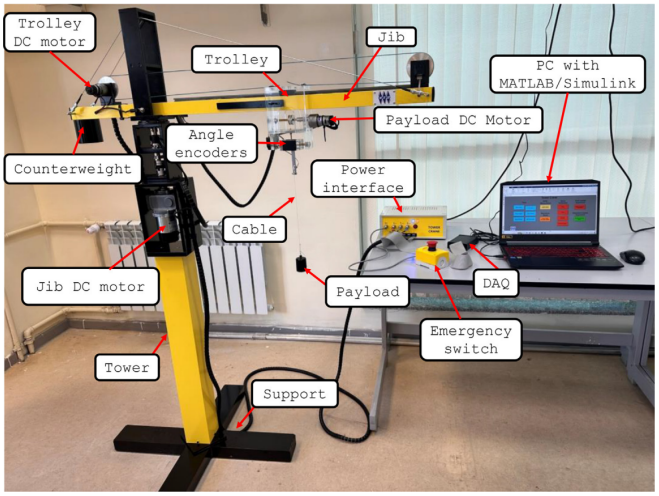


FIG. 1. The tower crane experiment setup.

1. Degrees of freedom

The crane has five degrees of freedom. Three are actuated: the radial displacement of the trolley along the jib, denoted x; the rotation of the jib about the vertical axis of the tower, denoted J; and the length of the suspension cable, denoted L, varied by the hoisting mechanism. Two are unactuated: the oscillation of the payload in the x axis, denoted θ, and in the y axis, denoted β.

Because the number of independent inputs is smaller than the number of degrees of freedom, the crane is underactuated: payload sway cannot be commanded directly and can only be shaped indirectly through trolley and jib motion. Payload sway therefore has to be predicted rather than measured and cancelled directly.

In every scenario used in this study the hoisting mechanism is inactive and cable length is constant within a recording, so the system reduces to four degrees of freedom (x, J, θ, β) under two actuated inputs. In the trolley-only family the jib is additionally held fixed, reducing the system further to two degrees of freedom (x, θ) under a single input. This is the simplest configuration the platform allows, and the natural starting point for the future analysis.

2. Sensing

The platform carries five encoders. Three are mounted on the drive motors and record trolley displacement, jib rotation, and cable length. The remaining two, referred to as angle encoders, are attached to the payload-hanging mechanism, and record the oscillations in the x and y axes.

The encoder resolution is 4096 counts per revolution, corresponding to a smallest observable change of $0.0879°$ ($0.001\,533$ rad).

3. Payload and motion types

Four payloads of differing mass and shape are available in the full dataset: a 300 g solid cylinder, used unless otherwise stated; a 75 g hollow ball; a 170 g hollow cylinder; and a 500 g distributed rectangular beam, which is suspended from two lines rather than one and so does not behave as a simple pendulum.

The dataset covers three kinds of excitation, which can be combined. Trolley motion displaces the payload along a single axis, so only θ oscillates. Jib rotation displaces it along both axes, so both θ and β oscillate. Cable-length change is never the sole excitation, so when it occurs the payload oscillates along whichever axis the accompanying motion excites.

Independently of motion type, every recording contains two regimes: an active-input segment, in which the crane is driven by a non-zero command, and a free-decay segment, in which the command is held at baseline and the oscillation dies out unforced.

The variables recorded depend on which axes are actuated, as summarised in Table I.

TABLE I. Variables recorded in each family of scenarios. A variable is recorded only when the corresponding axis is energised. After Ref. [8].

Variable	Symbol	Trolley	Jib	Simult.
Reference trolley command	x_r	✓	—	✓
Actual trolley displacement	x	✓	—	✓
Reference jib command	J_r	—	✓	✓
Actual jib rotation	J	—	✓	✓
Cable length	L	✓	✓	✓
Sway, x axis	θ	✓	✓	✓
Sway, y axis	β	—	✓	✓

4. Scenarios used

This study uses 23 of the 33 scenarios in the data set (69.7%). The scenarios are grouped into three families that hold cable length fixed at the platform's default payload while varying crane motion: trolley motion alone (scenarios 1–8), jib rotation alone (scenarios 14–20), and simultaneous trolley motion and jib rotation (scenarios 24–31). The remaining scenarios, which involve hoisting or payload other than the standard 300 g default, are excluded because they vary more than one factor at a time and would confound cable length with mass, shape or a time-varying pendulum length. They are noted as a direction for future work.

The payload mass is therefore constant across every scenario used here, and every result below describes the crane for the default payload only.

B. Effective oscillation length

The angle encoders are located on the suspension mechanism, not on the payload, so the recorded cable length L is not the length of the physical pendulum that determines the oscillation period: the hook, suspension equipment, and the payload's own dimensions add a geometric offset that the cable encoder does not consider.

The effective length was therefore estimated independently from the oscillation period,

$$T = 2\pi\sqrt{\frac{L_{\text{eff}}}{g}} \quad \Longrightarrow \quad L_{\text{eff}} = \frac{T^2}{4\pi^2}\, g, \tag{1}$$

with the period measured from the free-decay segment of each recording. This was repeated for every scenario in a family, giving one (L, L_{eff}) pair per scenario, and a straight line was fitted through the resulting points. L_{eff} is used only to describe the system and is never given to the model as a feature. It can therefore be fitted using all scenarios, including those held out during model evaluation.

C. Prediction task

The model predicts the payload sway angle at a future time using its recent measured history. The target is the sway angle $\hat{\theta}(t+h)$ at a horizon h, and the feature vector is

$$\mathbf{x}(t) = \big[\theta(t-\Delta),\ \theta(t-2\Delta),\ \theta(t-3\Delta),\ L(t)\big], \quad \Delta = 0.01\,\text{s}. \tag{2}$$

The pendulum is a second-order system, so two lags are enough in principle to describe unforced motion; the third absorbs damping and quantisation noise. Wider lag windows are tested further.

This justification, however, only establishes that three lags suffice to predict the *current* state of an unforced oscillator. It does not by itself justify predicting h steps ahead under an active command. During the forced segment of a recording, $\theta(t+h)$ depends on the state at time t and on whatever force is subsequently applied between t and $t+h$. That second quantity is absent from $\mathbf{x}(t)$: neither x_r nor J_r appears as a feature. The prediction problem is therefore well-defined during free decay, since the future motion can be inferred from the measured history. During the forced part, however, the future motion also depends on the crane commands, which are not available to the model.

Every horizon is fitted as a separate model with its own target. This avoids accumulating errors from repeated one-step predictions and allows each horizon to be evaluated independently.

Because the two axes have different natural frequencies (Sec. III B), a single model conditioned on L would predict the same frequency for both. Families with two angles therefore use separate models for θ and β, each built from that axis's own three lags and L.

D. Data separation

Because no further data could be collected from the physical installation, the scenarios were split into training and evaluation sets by whole scenario. At 100 Hz, neighbouring rows are very similar, so a random split would put nearly identical samples in both sets. The resulting error would then reflect interpolation between nearby samples instead of generalisation to new conditions.

Model performance is evaluated using a leave-one-scenario-out (LOSO) scheme within each family. Each scenario is used once as the test set, while the remaining scenarios are used for training. From these remaining scenarios, the one with the cable length closest to their median is used for validation. The validation set is used only for early stopping of the neural network. The other scenarios are used for training.

This evaluation scheme is used for all results below, except for the nine-point horizon sweep in Sec. III D. For this experiment, scenarios 1–6 are used for training and scenarios 7–8 are used for testing.

The training and validation sets are subsampled by a factor of 4 to reduce the size of the fitting problem. The test sets always use every recorded sample. Therefore, all reported test errors are calculated from the full test data.

1. Extrapolation distance.

Two distances characterise how far a held-out scenario lies from the training range:

$$d_L = \max\bigl(0,\ L_{\text{test}} - \max L_{\text{train}},\ \min L_{\text{train}} - L_{\text{test}}\bigr), \quad (3)$$

and d_ω, defined analogously in natural-frequency space through the mapping of Sec. II B. The second definition exists because the length-frequency relationship is nonlinear, so equal intervals in L do not correspond to equal intervals in frequency.

2. Forced and free-decay regimes.

No column in the dataset labels individual samples by regime. The two are defined operationally by a fixed threshold at 15 s: command pulses in every family used here are confined to the first 13 s, and the threshold is set slightly beyond this to let the transient following the final pulse complete. A sample with target time $t + h < 15$ s is scored as forced; all later samples as free decay.

E. Models

Five methods are compared, starting with two simple baselines, followed by linear regression, gradient-boosted trees, and a neural network. We first compare the ML models with two simple baselines that do not require training. This shows whether the learned models provide a meaningful improvement over predictions based only on the current or previous state.

1. Baselines.

The zero forecast predicts $\hat{\theta}(t+h) = 0$ for every t and h; its MAE therefore equals the mean absolute value of the target signal in that scenario, and this is the threshold of usefulness used throughout Sec. III: any model whose error exceeds it is worse than predicting no sway at all. Persistence predicts $\hat{\theta}(t+h) = \theta(t)$, the simplest forecast that uses the available history.

2. Linear regression.

An ordinary least-squares fit on the features of Eq. (2),

$$\hat{\theta}(t+h) = \boldsymbol{\beta}^{\mathsf{T}}\mathbf{x}(t) + \beta_0. \quad (4)$$

This is a stationary linear autoregressive representation of the dynamics, an ARX model in system-identification terms.

3. Gradient-boosted trees.

A regression tree partitions the feature space into disjoint regions $R_1, \ldots, R_J$, splitting on the feature and threshold that most reduce the within-node sum of squared residuals. Each terminal region is assigned a constant equal to the mean target value of the training points it contains, so a single tree is a piecewise-constant function of its input:

$$h(\mathbf{x}) = \sum_j c_j\, \mathbf{1}[\mathbf{x} \in R_j]. \quad (5)$$

An ensemble of M trees is fitted sequentially by gradient boosting. This is not a random forest, in which trees are trained independently on bootstrap samples and averaged. Here the ensemble is additive: the first tree is fitted to the data directly, and each subsequent tree h_m is fitted to the residuals of the ensemble built so far, then added with a shrinkage weight ν:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu\, h_m(\mathbf{x}), \quad m = 1, \ldots, M. \quad (6)$$

Under squared-error loss this is least-squares boosting. The ensemble is trained with $M = 200$, $\nu = 0.1$, and a maximum depth of 200 splits. MATLAB's default depth of roughly 10 splits underfits the task, with the resulting ensemble losing to linear regression at every horizon, and was increased accordingly.

Two structural properties of Eq. (5) are used later. Because prediction within a leaf is constant, the model's response to a feature value outside the training range equals its response at the nearest training boundary: the model does not extrapolate but plateaus. Since splits represent threshold comparisons, any monotonic transformation of a single feature only changes the threshold labels without changing the partition, so tree-based models are invariant to monotonic transformations of their input data.

4. Neural network ensemble.

A feedforward network with two hidden layers of 20 units each and rectified-linear activations (Fig. 2):

$$\begin{aligned} \mathbf{a}^{(1)} &= \mathrm{ReLU}(W^{(1)}\mathbf{x} + \mathbf{b}^{(1)}), \\ \mathbf{a}^{(2)} &= \mathrm{ReLU}(W^{(2)}\mathbf{a}^{(1)} + \mathbf{b}^{(2)}), \\ \hat{y} &= \mathbf{w}_{\text{out}} \cdot \mathbf{a}^{(2)} + b_{\text{out}}, \end{aligned} \quad (7)$$

where $\mathrm{ReLU}(z) = \max(0, z)$. Unlike Eq. (5) this is a piecewise-linear, continuous function of the input. Features are standardised to zero mean and unit variance using training-set statistics, which is necessary given the differing scales of sway angle (order 0.1 rad) and cable length (order 0.5 m); tree-based models require no such step. Training stops after 20 iterations without validation improvement, or after a cap of 5000 iterations.

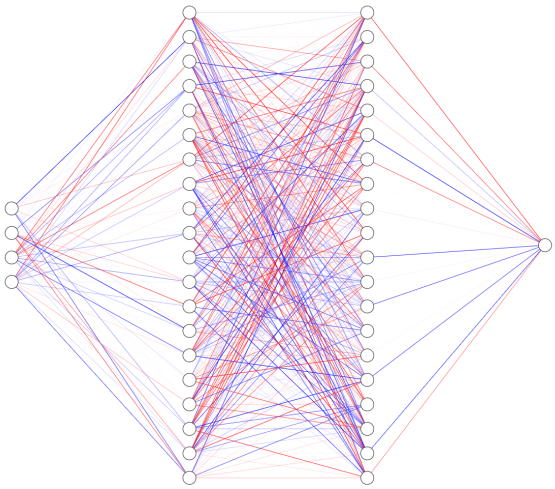


FIG. 2. Architecture of a single network in the ensemble: 4 inputs, three lags of θ and the cable length L, feed two hidden layers of 20 ReLU units, producing $\hat{y}$. Five such networks, differing only in random initialisation, are trained and their predictions averaged. Generated with NN-SVG [10].

A single network's accuracy is sensitive to its random initialisation: across seeds, the standard deviation of normalised MAE reaches 0.10 to 0.11 at the longer horizons, comparable to the differences of substantive interest here. Five networks are therefore trained with distinct fixed seeds, and their predictions, not their errors, are averaged:

$$\hat{y}_{\text{ens}}(\mathbf{x}) = \frac{1}{5}\sum_{k=1}^{5} \hat{y}_k(\mathbf{x}). \tag{8}$$

Averaging predictions rather than errors matters because independent initialisation errors partly cancel. Unlike the tree ensemble, the ReLU network continues to change approximately linearly outside the training range. We therefore expect it to extrapolate cable length better than the tree model.

F. Metrics

Performance is reported with five metrics: mean absolute error (MAE), root-mean-square error (RMSE), maximum absolute error (MaxAE), the coefficient of determination (R^2), and MAE normalised by the zero-forecast MAE of the same scenario. For N samples with true value y_i and prediction $\hat{y}_i$,

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

$$\text{MaxAE} = \max_{1\le i\le N} |y_i - \hat{y}_i|$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

with $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$. MaxAE is included because a controller needs to bound the worst case, as it might matter more than the average error.

The normalised MAE compares a model against the zero forecast $\hat{y}_i = 0$ on the same scenario,

$$\text{MAE}_{\text{norm}} = \frac{\text{MAE}}{\text{MAE}_{\text{zero}}} = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{\sum_{i=1}^{N} |y_i|},$$

so a value below one means the model beats predicting no sway at all, and a value near one means it adds almost nothing.

Metrics are reported per scenario and per axis rather than pooled across a family. Mean oscillation amplitude varies by roughly a factor of five between scenarios within a family (Table II), so a pooled metric would be dominated by the highest-amplitude scenarios.

G. Reproducibility

Data exploration, model fitting, and evaluation were carried out in MATLAB R2026a with the Statistics and Machine Learning Toolbox. The dataset is publicly available [11].

III. RESULTS

A. Data characteristics

Every recording follows the same structure (Fig. 3). Command pulses occupy the first 13 s, after which the input returns to baseline and the payload oscillation decays unforced for the remaining 86 s. Roughly 87% of each record is therefore free decay, and the forced regime that matters most for control occupies a minority of the data.

Damping is weak. In scenario 1 the mean peak-to-peak amplitude over the first 10 s is 9.03°, and over the final 19 s it is still 3.02°. The oscillations persist throughout the recording. At 1.0 s ahead, the signal is still large

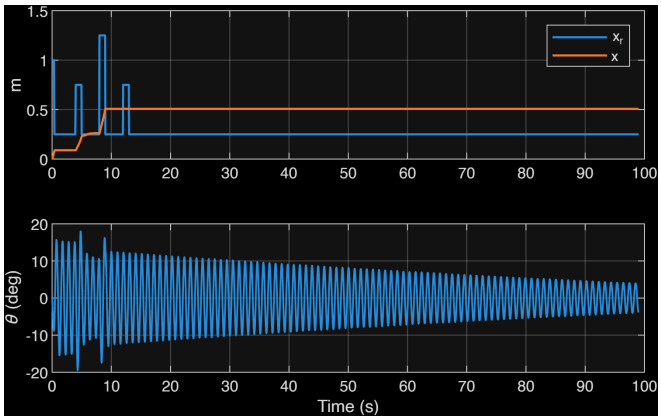


FIG. 3. Full 99 s record of scenario 1. Top: reference command x_r and actual trolley displacement x. Bottom: payload sway θ. Command pulses are confined to the first 13 s; the remainder of the record is free decay.

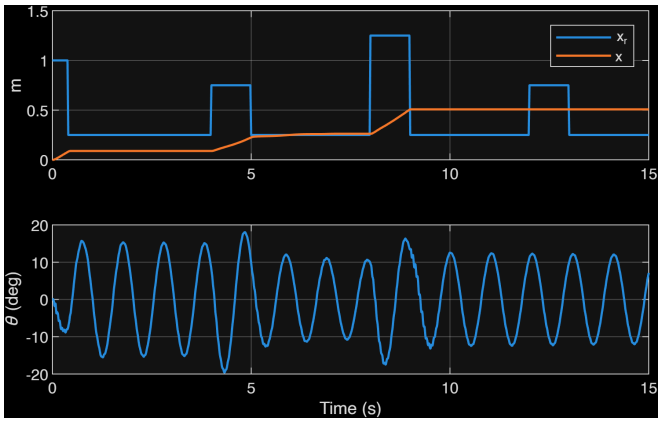


FIG. 4. The same record over the first 15 s. The trolley advances only while the command is held above baseline and holds its new position afterwards, which is inconsistent with x_r being a position setpoint.

TABLE II. Characteristics of the trolley-motion scenarios. L is the encoder-recorded cable length, T the free-oscillation period, and L_{eff} the effective pendulum length from Eq. (1). The offset column is $L_{\text{eff}} - L$.

Sc.	L (m)	T (s)	$\overline{\lvert\theta\rvert}$ (deg)	L_{eff} (m)	Offset (m)
1	0.0986	1.029	5.48	0.2632	0.1646
2	0.1965	1.195	10.94	0.3548	0.1583
3	0.2962	1.348	13.30	0.4516	0.1554
4	0.4008	1.481	5.96	0.5448	0.1440
5	0.5033	1.608	2.51	0.6426	0.1393
6	0.5964	1.710	6.22	0.7264	0.1300
7	0.7122	1.830	4.63	0.8322	0.1200
8	0.8087	1.928	8.45	0.9241	0.1154

enough that a zero forecast is not adequate. This makes long-horizon prediction relevant for this dataset.

Mean absolute sway varies by more than a factor of five across the family, from 2.51° in scenario 5 to 13.30° in scenario 3 (Table II). This spread is the reason all comparisons below are normalised by the zero-forecast error of the same scenario rather than pooled, and it is also the reason scenario 5 requires care: its small denominator inflates any normalised error computed on it.

B. Effective pendulum length

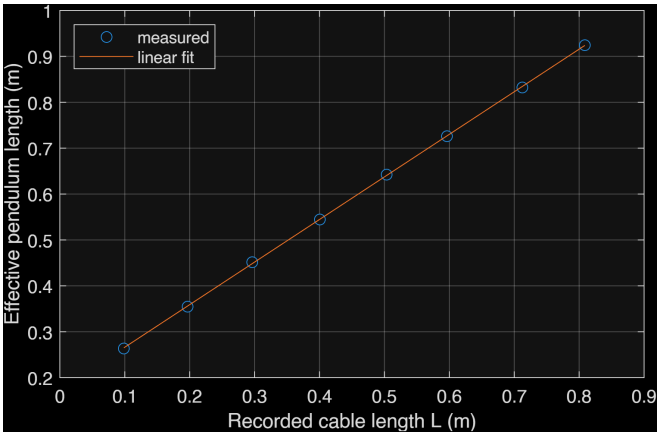


FIG. 5. Effective pendulum length against recorded cable length for the trolley family, with a least-squares line.

Estimating the effective length from the free-oscillation period gives a linear relationship with recorded cable length (Fig. 5). The relationship was measured five times independently, on three excitation types and two axes (Table III). The slopes span 1.3% and the intercepts 1.5 cm. A fit pooling all 30 points gives

$$L_{\text{eff}} = 0.924\,L + 0.166. \tag{9}$$

TABLE III. Five independent measurements of the length relationship $L_{\text{eff}} = aL + b$.

Measurement	Slope a	Intercept b (m)
Trolley, θ	0.9278	0.1735
Jib, θ	0.9239	0.1710
Jib, β	0.9286	0.1586
Simultaneous, θ	0.9166	0.1688
Simultaneous, β	0.9255	0.1591
Pooled	0.9244	0.1662

Because the five measurements come from three different excitation mechanisms and two different axes, this behaviour follows from the dynamics of the system rather than from a lack of data.

Axis asymmetry. The intercepts separate consistently by axis: the fitted intercept along x exceeds that along y by 12.4 mm in the jib family and 9.7 mm in the simultaneous family. Taken per scenario rather than through the fits, the mean difference is 7.9 mm.

Since natural frequency depends on effective length, the two axes oscillate at slightly different frequencies even at the same recorded cable length. Measured period differences range from 0.004 s to 0.041 s across the jib scenarios, producing slow beats in the relative phase of the two angles (Table IV).

TABLE IV. Period difference between axes and the resulting beat period, jib family. The observation window is 84 s.

Sc.	$T_\theta - T_\beta$ (s)	Beat period (s)	Cycles in window
14	+0.026	40	2.1
17	+0.041	53	1.6
15	+0.015	93	0.9
18	+0.019	138	0.6
20	+0.021	160	0.5
19	−0.015	192	0.4
16	+0.004	502	0.2

TThese results support the modelling choice in Sec. II C. A single model conditioned only on L would assign the same frequency to both axes, which is inconsistent with the measurements.

C. Degeneracy of one-step prediction

At the shortest horizon the prediction task is trivially solvable, and this needs establishing before any model comparison is reported. On scenario 1, predicting one step (0.01 s) ahead, linear regression on three lags reaches a mean absolute error of 0.0796°, against a persistence error of 0.3332°, a zero-forecast error of 5.484°, and an encoder resolution of 0.0879° (Table V).

TABLE V. One-step prediction on scenario 1. The linear model reaches the measurement floor.

Method	MAE (deg)
Zero forecast (mean $\lvert\theta\rvert$)	5.4841
Persistence	0.3332
Linear regression, three lags	0.0796
Encoder resolution	0.0879

The linear model's error is below the encoder's resolution. There is little value in comparing the model classes at this stage because the linear model already reaches the encoder resolution. Any remaining difference is mainly related to how the models handle quantisation noise.

This behaviour follows from the system dynamics rather than from a lack of data. A pendulum sampled at 100 Hz moves a negligible fraction of its period in one sample, and a linear autoregression on three lags recovers the recurrence relation of a harmonic oscillator to within noise.

The main question is therefore the useful prediction horizon. We next examine how far ahead the models can predict sway and how their ranking changes with horizon.

D. Error against prediction horizon

TABLE VI. Mean absolute error (deg) against prediction horizon, trolley family. Bold marks the best method at each horizon. This sweep uses the fixed split of Sec. II D, scenarios 1–6 for training and 7–8 for testing, so its values are not comparable with the leave-one-scenario-out results of Table IX.

h (s)	Zero	Persist.	Linear	Trees	Linear/Zero
0.01	6.55	0.43	**0.160**	0.198	0.024
0.02	6.55	0.64	**0.259**	0.304	0.040
0.05	6.55	1.25	**0.466**	0.498	0.071
0.10	6.55	2.29	0.920	**0.855**	0.140
0.20	6.56	4.30	2.113	**1.622**	0.322
0.35	6.56	7.10	4.276	**2.824**	0.652
0.50	6.56	9.48	5.912	**3.398**	0.901
0.75	6.57	12.16	5.997	**3.193**	0.912
1.00	6.58	12.78	7.712	**3.550**	1.172

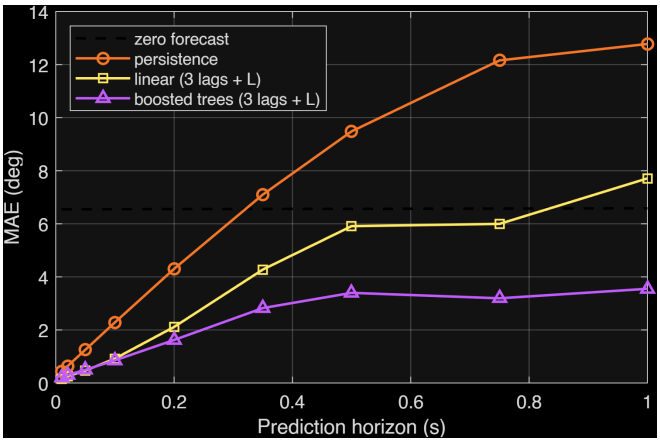


FIG. 6. Mean absolute error against prediction horizon. The dashed line is the zero forecast. Linear regression crosses it near 0.85 s. The sweep uses the fixed split of Sec. II D, not leave-one-scenario-out.

Three changes are visible from Table VI and Fig. 6.

At one step the tree ensemble is worse than linear regression (0.198 against 0.160), which is consistent with Sec. III C: where the target is nearly a linear function of recent history, added model capacity only fits noise.

The crossover lies near 0.07 s. At 0.05 s linear regression leads; at 0.10 s the trees are better. Below this horizon the additional flexibility of a nonlinear model is harmful.

Near 0.85 s linear regression crosses the zero forecast. At 1.0 s its error is 7.71° against a zero-forecast error of 6.58°, so predicting no sway at all would be more accurate than using the model. The tree ensemble stays below the zero forecast at every horizon tested.

Persistence exceeds the zero forecast from roughly 0.20 s onward, as expected: once the horizon approaches a quarter period, holding the current angle constant is systematically worse than assuming the pendulum is near its mean.

Adding L to the linear model changed its error only in the fourth decimal place. In Eq. (4) the cable length enters through a single additive coefficient, while the frequency of the fitted recurrence is carried by the three lag coefficients, and those are shared across every scenario in the training set. An additive term can only shift the predicted level, and the mean sway is close to zero in every recording, so there is nothing for it to shift. The linear model is therefore fitted at one compromise frequency and applies that frequency to every cable length.

This is consistent with the error analysis in Sec. III F. Advancing the phase over a horizon requires the frequency; the frequency reaches a model only through L; and in the linear case L cannot enter the part of the model that would carry it. The same feature is available to every method compared here, but only the nonlinear ones are able to act on it.

E. Interpolation against extrapolation

TABLE VII. Normalised MAE (MAE divided by the zero-forecast MAE of the same scenario) under leave-one-scenario-out validation, split by whether the held-out cable length falls inside the training range. Six interpolation folds, two extrapolation folds.

	Linear		Trees	
h (s)	Interp.	Extrap.	Interp.	Extrap.
0.1	**0.122**	**0.144**	0.178	0.205
0.2	**0.224**	**0.347**	0.311	0.375
0.5	**0.485**	0.936	0.509	**0.665**
1.0	0.726	1.304	**0.503**	**0.831**

At 1.0 s the linear model exceeds the zero forecast on extrapolation folds (1.304), and the tree ensemble degrades by 65% relative to interpolation (0.831 against 0.503). Both methods lose accuracy outside the training range, and the loss grows with horizon (Table VII, Fig. 7).

Why trees fail outside the training range. Section II E established that a tree's prediction is constant beyond the range of feature values seen in training. This can be seen directly: when scenario 8 ($L = 0.81$ m) is held out and the training set spans up to 0.596 m, every threshold on L has been crossed, and the ensemble returns the value

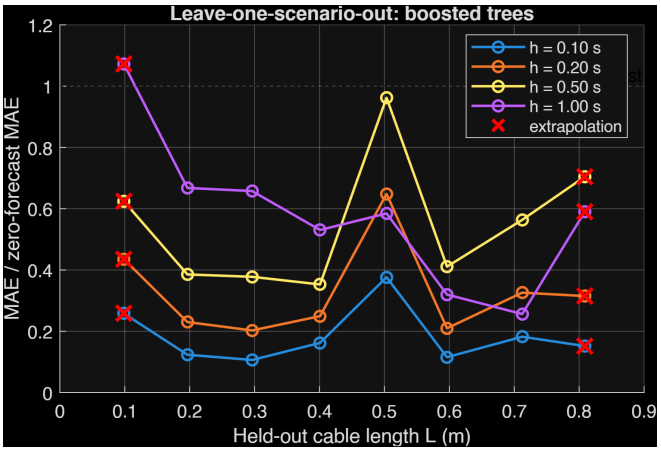


FIG. 7. Normalised error against held-out cable length. Crosses mark extrapolation folds. The peak at $L = 0.5$ m is partly a normalisation artefact: scenario 5 has the smallest mean sway in the family.

it would return for a 0.596 m cable. The model does not extrapolate; it treats the longest unseen cable as the longest cable it knows.

Measuring distance in frequency. The distance from the training range is not equivalent in metres and in frequency, because ω depends on L nonlinearly. In frequency space the gap below the trolley training range is 0.847 rad s^{-1} and the gap above it 0.175 rad s^{-1}, a ratio of 4.8. In both families the short-cable end is the harder extrapolation, and in both the asymmetry is invisible if distance is measured in metres.

F. Decomposition of the error

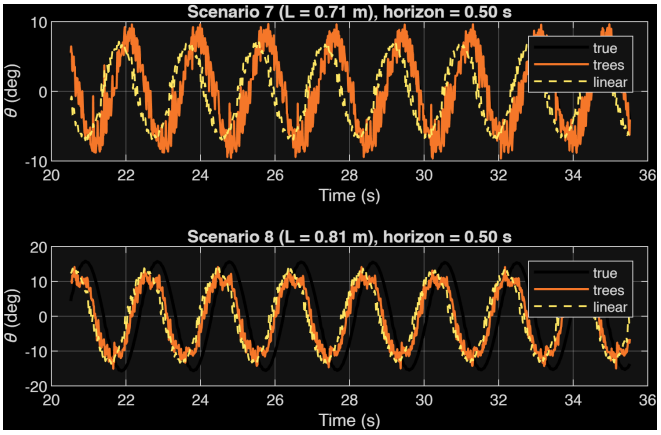


FIG. 8. Predicted against true sway over time at $h = 0.5$ s for the two evaluation scenarios. Amplitude is broadly reproduced; the predicted waveform lags the true one.

MAE alone cannot show whether the model predicts the correct waveform with a time shift or predicts the

waveform incorrectly. Decomposing the error at $h = 0.5\,\mathrm{s}$ (Fig. 8, Table VIII) shows amplitude reproduced to between 79% and 93%, while the time offset reaches 0.32 s, or 10% to 17% of the oscillation period. The error is mostly a phase error. The predicted oscillation therefore lags behind the true signal at longer horizons.

TABLE VIII. Error decomposition at $h = 0.5\,\mathrm{s}$. Amplitude ratio is predicted over true peak-to-peak amplitude; shape agreement is the correlation of the two waveforms after the time offset is removed.

Scenario, model	Ampl. ratio	Shape agr.	Offset (s)
7, trees	0.93	0.90	0.02
7, linear	0.86	0.59	0.27
8, trees	0.79	0.74	0.21
8, linear	0.84	0.49	0.32

The same behaviour appears as a regression slope below unity that worsens with horizon (Fig. 9). A slope below one means predictions are systematically compressed toward the mean. This is expected under squared-error training: when the future phase is uncertain, predictions are pulled towards the mean.

Read together with Sec. II C, this identifies what the models are failing at. Advancing an oscillator's phase by a full second requires knowing its frequency precisely, and frequency reaches the model only through the feature L.

G. Full model comparison

Table IX gives the full comparison. Two results are particularly relevant here.

The network ensemble reaches a plateau while the other methods degrade. Its normalised error runs $0.220 \to 0.416 \to 0.416$ from 0.1 to 1.0 s, unchanged between half a second and a second, while the trees run $0.188 \to 0.588 \to 0.612$ and the linear model $0.128 \to 0.605 \to 0.883$.

MaxAE reverses the ranking at the short horizon. At 0.1 s the linear model has the best MAE (0.861°) and the worst maximum error among the fitted models (14.47° against 11.82° for the network). Its ratio of MaxAE to MAE is 17:1, against 7.5:1 for the network. The linear model is usually almost exact but occasionally fails badly, which for control may matter more than the average.

H. Reproduction on other motion families

The analysis was repeated on the jib-rotation family (scenarios 14–20) and the simultaneous-motion family (scenarios 24–31), with separate models per axis.

The ordering matches the trolley family (Table X): linear regression is competitive at the shortest horizon, the network leads at the longest, and the trees sit between them.

The extrapolation failure also continues. Scenario 14 ($L = 0.096\,\mathrm{m}$, extrapolating below the training range) gives normalised errors of 1.02 and 1.01 at $h = 1.0\,\mathrm{s}$, exceeding the zero forecast. Scenario 20 ($L = 0.712\,\mathrm{m}$, extrapolating above) gives 0.56 and 0.55. The short-cable end is again the harder direction, consistent with the frequency-space argument of Sec. III E.

Under simultaneous motion, θ is roughly twice as hard to predict as β (Table XI), which was not the same for the jib family (0.494 against 0.492 at $h = 1.0\,\mathrm{s}$). The explanation follows from the excitation structure described in Sec. II C. Trolley motion displaces the payload along x only; jib rotation displaces it along both axes. Under simultaneous motion, θ therefore receives excitation from two independent sources while β receives it from one.

Ordering the five prediction targets by number of excitation sources (Table XII), the one target driven by two sources is the hardest, and the four driven by one source cluster between 0.37 and 0.49.

The method ranking is identical in all three families. Within the motion families tested here, this ordering is consistent across the different manoeuvres.

I. Forced and free-decay regimes

Section II C predicted that the prediction task is well posed only during free decay, because the forced segment depends on commands absent from the feature vector. Splitting the error at the 15 s threshold confirms this (Table XIII).

Accuracy is 1.4 to 2.8 times worse under command than during free decay. The effect is consistently larger for β than for θ, and for β it grows with horizon. This limits the direct use of the current model in a closed-loop control setting. The forced segment is the part of the motion where a controller would need the prediction most, but it is also where the models perform worst. It accounts for only 14% to 15% of the available data.

The lower accuracy in the forced regime is expected from the current formulation of the prediction task. The future command is not included in the feature vector, so the model does not have access to information that directly affects the future payload motion. A predictor used in a feedforward loop should therefore also receive the commanded trajectory as an input. The controller already knows this trajectory in advance.

J. Negative and corrected results

This section collects results where a plausible hypothesis was tested and not confirmed. Each constrains what the remaining findings can mean.

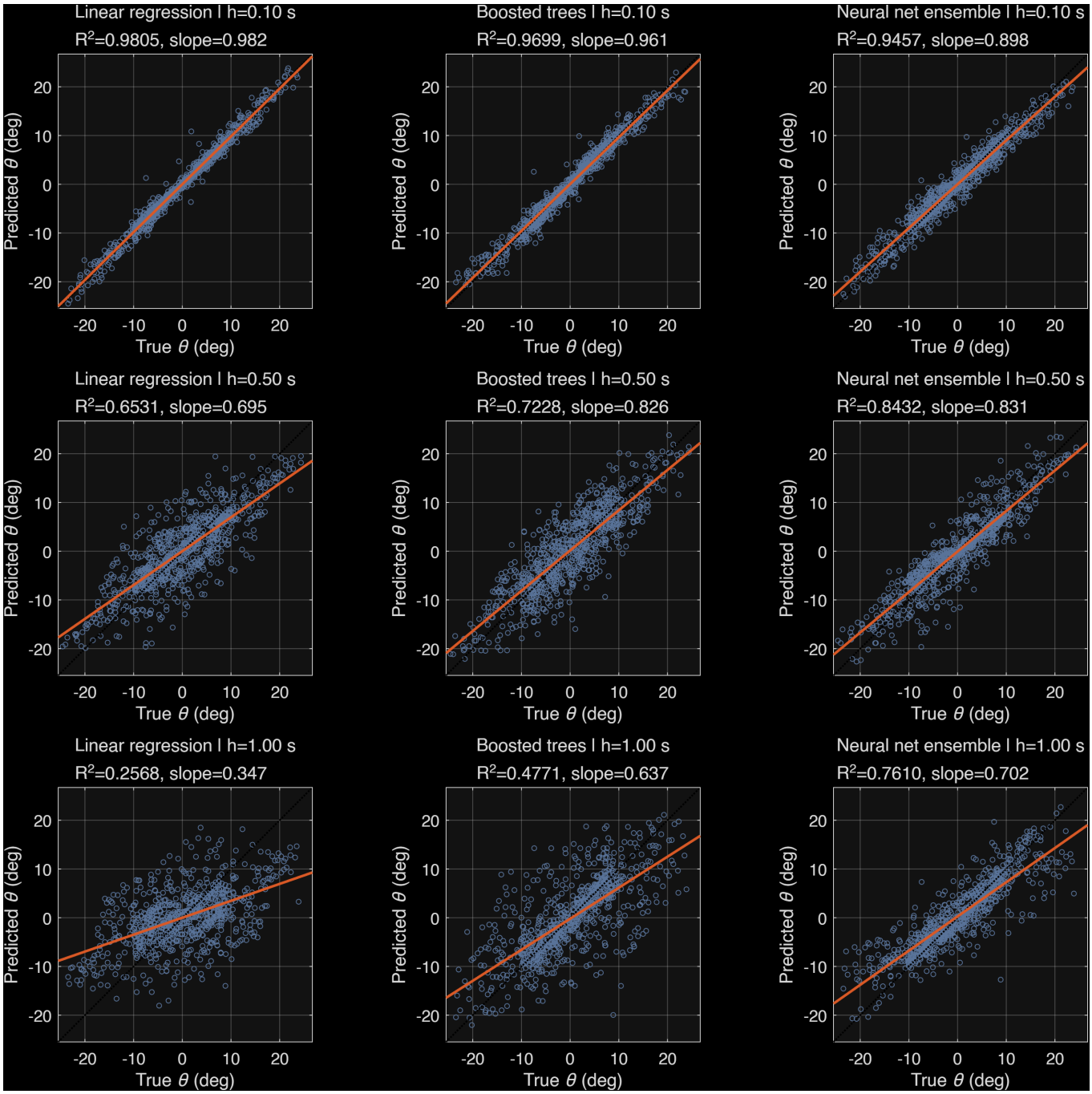


FIG. 9. Predicted against true sway for three models at three horizons, with the fitted slope and R^2 on each panel. A slope below unity indicates predictions compressed toward the mean.

Widening the lag window. Three lags at 100 Hz span 0.03 s, under 2% of the oscillation period, which suggests the window is too narrow for a 1.0 s horizon. Lags at 1, 5, 10, 20, 35, and 50 steps were tested. The result was mixed: an improvement at 0.5 s and a degradation at 1.0 s, with no clear gain.

The same pattern appears more strongly when the other axis's lags are added in the two-angle families (Table XIV). The penalty is positive in all six cases and grows with horizon. Additional features consistently hurt, confirmed independently in three families with different feature sets.

Initialisation variance and three withdrawn conclusions. An initial comparison using a single random seed found the network outperforming the trees by 39% at $h = 0.5$ s, averaged over eight folds (0.346 against 0.569). Repeating across seeds showed the standard deviation of normalised error across initialisations to be 0.10 to 0.11

TABLE IX. All five methods on the trolley family under leave-one-scenario-out validation, averaged over eight folds. MAE, RMSE and MaxAE in degrees. Relative MAE is normalised by the zero forecast of the same scenario.

h (s)	Model	MAE	RMSE	MaxAE	R^2	Rel. MAE	RMSE/MAE
0.1	zero	7.196	8.134	16.17	0.000	1.000	1.137
	persistence	3.407	3.924	**9.16**	0.769	0.462	1.160
	linear	**0.861**	**1.220**	14.47	**0.968**	**0.128**	1.511
	trees	1.150	1.514	12.65	0.946	0.188	1.322
	net_ens	1.567	1.917	11.82	0.936	0.220	1.287
0.5	zero	7.205	8.143	16.17	0.000	1.000	1.137
	persistence	12.562	14.315	28.33	−2.024	1.717	1.147
	linear	4.060	4.754	19.37	0.543	0.605	1.194
	trees	3.749	4.470	**18.60**	0.567	0.588	1.203
	net_ens	**2.750**	**3.358**	21.62	**0.767**	**0.416**	1.250
1.0	zero	7.213	8.147	**16.17**	0.000	1.000	1.136
	persistence	10.233	11.687	23.32	−1.578	1.475	1.220
	linear	6.123	7.040	24.63	0.096	0.883	1.167
	trees	4.542	5.594	21.77	0.501	0.612	1.242
	net_ens	**3.060**	**3.726**	18.78	**0.769**	**0.416**	1.257

TABLE X. Mean absolute error (deg) by horizon and axis, jib family. Normalised values are the ratio of the mean errors.

h (s)	Axis	Linear	Trees	Network	Net./Zero
0.1	θ	0.770	0.965	**0.757**	0.083
	β	1.029	**1.001**	1.099	0.121
0.5	θ	5.298	4.213	**3.389**	0.369
	β	5.322	4.084	**3.864**	0.426
1.0	θ	7.822	5.551	**4.564**	0.494
	β	7.560	5.184	**4.466**	0.492

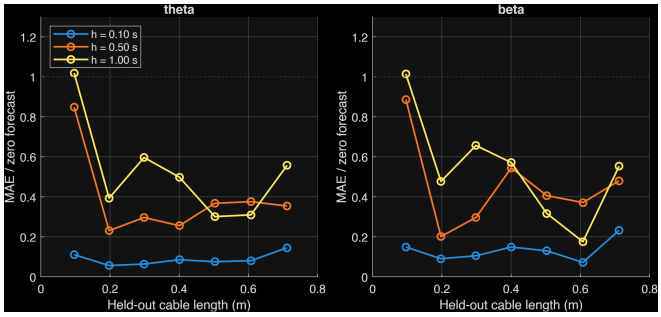


FIG. 10. Normalised error against held-out cable length for both axes, jib family. The failure at the shortest cable is visible in both panels.

TABLE XI. Mean absolute error (deg) by horizon and axis, simultaneous-motion family.

h (s)	Axis	Linear	Trees	Network	Net./Zero
0.1	θ	**0.966**	1.070	1.002	0.106
	β	0.865	0.859	**0.842**	0.097
0.5	θ	7.436	**6.277**	6.651	0.701
	β	4.414	3.164	**2.644**	0.304
1.0	θ	7.902	7.727	**7.193**	0.757
	β	6.352	4.806	**3.248**	0.372

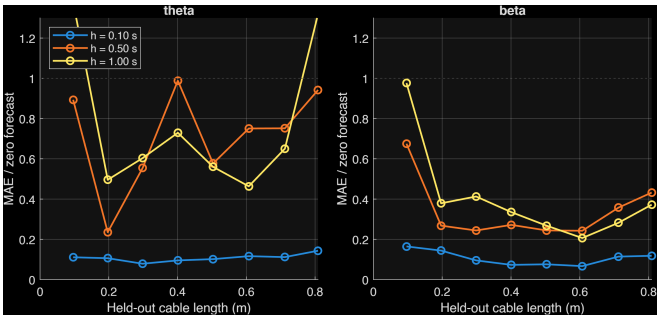


FIG. 11. Normalised error against held-out cable length, simultaneous motion. The θ panel exceeds unity at the extremes; the β panel does not.

TABLE XII. Normalised error at $h = 1.0\,\mathrm{s}$ against number of independent excitation sources.

Target	Normalised error	Sources
Simultaneous, θ	0.757	two
Jib, θ	0.494	one
Jib, β	0.492	one
Trolley, θ	0.416	one
Simultaneous, β	0.372	one

TABLE XIII. Network-ensemble MAE (deg) split by regime, simultaneous-motion family.

h (s)	Axis	Under command	Free decay	Ratio
0.1	θ	1.625	0.892	1.8
	β	1.508	0.724	2.1
0.5	θ	8.623	6.312	1.4
	β	5.608	2.134	2.6
1.0	θ	9.559	6.800	1.4
	β	7.172	2.596	2.8

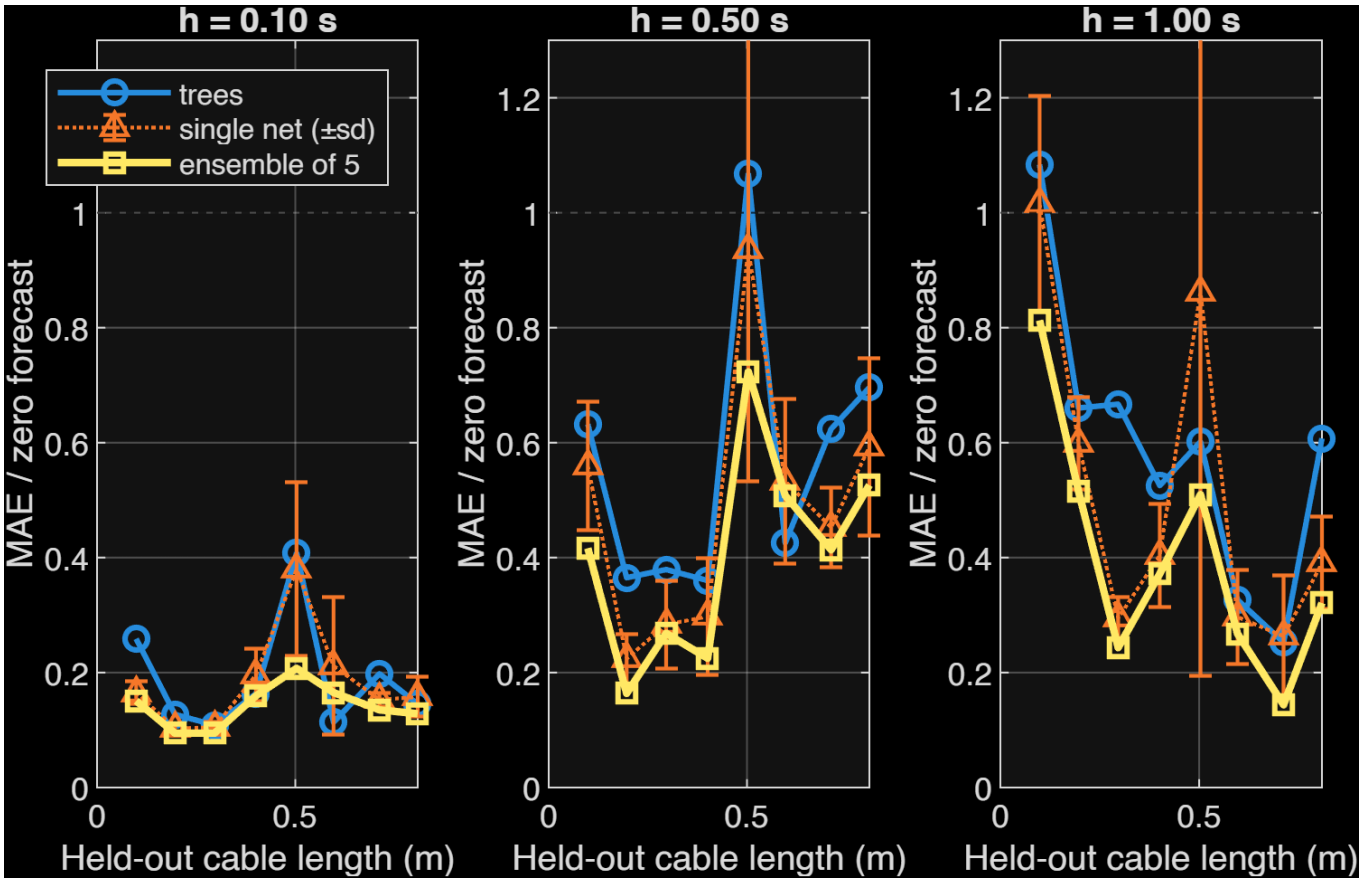


FIG. 12. Tree ensemble, single network with spread, and the five-network ensemble. The ensemble tracks the lower edge of the single-network spread.

TABLE XIV. Penalty from adding the other axis's lags, jib family, in units of normalised MAE. A positive value means the seven-feature model is worse than the four-feature model.

h (s)	θ	β
0.1	+0.155	+0.233
0.5	+0.373	+0.599
1.0	+0.516	+0.988

at long horizons, comparable to the differences the conclusions rested on. Three claims were withdrawn.

The claim that ω does not help the network was not supported: the difference between feature sets is 0.048 at 0.5 s and 0.089 at 1.0 s, both smaller than the seed spread of 0.113 and 0.102 respectively, and at 0.1 s the sign reverses. The correct statement is that the effect is not resolvable at this sample size.

The claim that the network performed exceptionally on scenario 5 was a property of one seed: with `rng(0)` the normalised error was 0.425, while across three seeds it was 0.766 ± 0.537, a full range of 1.008.

The claim that the network showed a larger gap between regimes than the trees reversed on repetition: one seed gave 1.42 against 1.24, three seeds gave 1.57 against 1.67.

After averaging over seeds the network advantage persists but is smaller than the single-seed estimate suggested: 22% at 0.1 s, 35% at 0.5 s, and 26% at 1.0 s (Table XV).

TABLE XV. Normalised MAE, trees against the network, averaged over five seeds. The final column is the standard deviation across seeds for the network.

h (s)	Trees	Network	Gain	Net. s.d.
0.1	0.190	0.148	22%	0.027
0.5	0.569	0.371	35%	0.113
1.0	0.591	0.439	26%	0.102

Ensembling. The ensemble beats the average single network at every horizon but does not beat the best single network selected in hindsight (Table XVI, Fig. 12). Its value is not higher peak accuracy but the removal of dependence on a fortunate initialisation.

TABLE XVI. Normalised MAE of the tree ensemble, a single network, and the five-network ensemble. The last two columns give the ensemble's gain over the mean single network and over the best single network chosen in hindsight.

h (s)	Trees	Single net	Ensemble	vs. mean	vs. best
0.1	0.190	0.184 ± 0.050	**0.142**	+0.042	−0.010
0.5	0.569	0.485 ± 0.138	**0.406**	+0.079	−0.080
1.0	0.591	0.516 ± 0.166	**0.398**	+0.118	−0.044

IV. DISCUSSION AND CONCLUSION

A. Interpretation

The results can largely be explained by differences in the natural frequency of the payload. Several cases with lower prediction accuracy occur when the natural frequency changes. The phase lag discussed in Sec. III F can be interpreted as a frequency error that accumulates over the prediction horizon. The poor extrapolation performance in Sec. III E is also consistent with this interpretation, since a tree model cannot directly represent a frequency that was not present in the training data. The difference between short and long cables can be explained by $\omega \propto L^{-1/2}$, which means that the same change in cable length corresponds to different changes in natural frequency.

This matters when selecting the prediction horizon. Cable length alone does not fully describe the distance between the training and test conditions. For example, a model trained between 0.2 and 0.8 m is better positioned to predict a 0.9 m cable than a 0.1 m cable, even though both test cases are separated from the training range by a similar distance in metres. The corresponding difference in natural frequency is not the same.

The input-feature results show a similar pattern. Below 0.07 s, the linear autoregressive model is already accurate enough and performs better than the tree ensemble. In this range, the extra flexibility of the tree model does not provide an advantage and may instead fit quantisation noise. Adding more history also did not improve the results. Both a wider lag window and additional history from another axis reduced accuracy, and the reduction became more noticeable at longer prediction horizons.

B. Limitations

Workspace. The crane is laboratory-scaled: maximum trolley displacement is 0.5 m and cable length ranges from 0 to 0.8 m. These bound the range over which any model trained on this data can operate.

Payload. All scenarios used the default 300 g solid cylindrical payload, so no claim here extends to other masses or shapes. The intercept in Eq. (9) is geometric and should shift with payload dimensions, which is testable.

Sample size. Each family contains six to eight scenarios, so leave-one-scenario-out gives few folds and correspondingly noisy per-fold estimates. Comparisons resting on single folds, such as the difference between scenarios 14 and 20, should be read as indicative.

Feature window. All results use three lags spanning 0.03 s, retained even at the 1.0 s horizon. Wider windows were tested and did not help, but the window was not optimised per horizon.

Command signals. The reference signals are not used as features, which is the direct cause of the regime gap in Sec. III I. Their units and baseline behaviour also remain partly unresolved.

Single rig. All data come from one installation. The length relationship in particular is a property of this rig's suspension geometry and should not be transferred numerically to another crane.

C. Future work

There are several possible ways to improve the model in future work. One option is to include the commanded trajectory as an additional input. Since this trajectory is already known to the controller, it could help reduce the difference between the different operating regimes and improve prediction accuracy. Another step would be to estimate the damping coefficient using the same period-based approach that was used to obtain L_{eff}. This would give a more complete second-order description of the rig. The model could also be trained to predict the difference between the measured angle and the response of an ideal damped pendulum, instead of predicting the angle directly. In this case, the model would only need to learn the part of the motion that is not explained by the basic physical model. Finally, scenarios 10–12 and 22–23, among those excluded in Sec. II A 4, change the payload mass and shape while keeping the cable length fixed. Comparing their intercepts with Eq. (9) would provide a relatively simple way to check whether the proposed geometric interpretation of the low-error offset is valid.

D. Conclusion

Payload sway on this rig can be predicted from three lags of the sway angle and the recorded cable length. However, the useful prediction horizon is shorter than required for control. Below roughly 0.07 s the task is solved by linear autoregression and the choice of model does not matter; at one step it is solved to below the resolution of the encoder. Past 0.85 s the linear model is worse than assuming no sway at all. Between these

two points, model performance depends on the prediction horizon. The network ensemble is the only model that stops degrading, with a normalized error of about 0.4. Cable lengths outside the training range cost accuracy in every method, and how far outside is a question about frequency and not about metres.

Two results are particularly important for using the predictor in a control loop. The residual error at long horizons is mainly a phase error, so the models do not provide the accurate phase prediction needed by a feedforward shaper. Accuracy under command is 1.4 to 2.8 times worse than during free decay, which is the regime in which a controller would use the predictor. Both results point to the same missing input: the commanded trajectory, which is known to the controller but was not provided to the predictor.

[1] M. Muddassir, T. Zayed, A. H. Ali, *et al.*, Automation in Construction **170**, 105889 (2025).

[2] M. R. Mojallizadeh, B. Brogliato, and C. Prieur, Annual Reviews in Control , 100877 (2023).

[3] A. B. Alhassan, Z. Mohamed, A. M. Abdullahi, A. A. Bature, A. Haruna, and N. M. Tahir, Jurnal Teknologi **80**, 61 (2018).

[4] S. M. Fasih, Z. Mohamed, A. R. Husain, L. Ramli, A. M. Abdullahi, and W. Anjum, Measurement and Control **53**, 1171 (2020).

[5] S. M. F. ur Rehman, Z. Mohamed, A. R. Husain, *et al.*, Automation in Construction **154**, 104963 (2023).

[6] K. Talapiden, A. B. Alhassan, Z. Mohamed, M. A. Shehu, and T. D. Do, Journal of Vibration and Control 10.1177/10775463251411778 (2026).

[7] M. A. Shehu, A. B. Alhassan, Z. Mohamed, M. O. Efe, K. Talapiden, and T. D. Do, ISA Transactions **173**, 648 (2026).

[8] A. B. Alhassan, S. M. U. Arif, A. Haruna, K. Talapiden, M. A. Shehu, and T. D. Do, Data in Brief **61**, 111822 (2025).

[9] Inteco, Tower crane, `https://www.inteco.com.pl/products/tower-crane/` (2025), accessed 2025.

[10] A. LeNail, Journal of Open Source Software **4**, 747 (2019).

[11] A. B. Alhassan, S. M. U. Arif, A. Haruna, K. Talapiden, M. A. Shehu, and T. D. Do, Trajectories and payload oscillations data of tower crane systems with varied payload masses, shapes, and cable lengths for machine learning-based control, system identification, Mendeley Data (2025).