

How Much Model Does Condition Monitoring Need? Reading a Six-Axis Arm from Its Control Loop

Wael Hafez

w.hafez@semarx.com

Semarx Research LLC <https://orcid.org/0000-0002-2775-6946>

Amir Nazeri

Semarx Research LLC

Chenan Wei

University of Massachusetts Amherst

Research Article

Keywords: Failure Detection and Recovery, Robot Safety, condition monitoring, information theory

Posted Date: September 22nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11101227/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

How Much Model Does Condition Monitoring Need? Reading a Six-Axis Arm from Its Control Loop

Wael Hafez
Semarx Research LLC
Alexandria, VA, USA
w.hafez@semarx.com

Amir Nazeri
Semarx Research LLC
Alexandria, VA, USA
amir.nazeri@semarx.com

Chenan Wei
University of Massachusetts
Amherst, MA, USA
weichenan0@gmail.com

Abstract—Condition monitoring for industrial robots must either model complex arm dynamics or learn deviations from healthy trajectories — deviations that feedback can mask by preserving motion while control effort changes. We test whether an explicit state–action–next-state representation offers a lower-complexity monitoring view of the compensation that closed-loop fault masking hides. On vora-us-AD, we discretize 48 of 130 telemetry signals into controller-residual states, realized-current actions, and one-step successor states. Ten empirical information scores are read against 360 phase-specific reference moments calibrated on 948 healthy recordings; no iterative optimization or learned weights are used. The configuration-informed detector reaches 88.6 macro AUROC (95% CI 86.3–90.8) across 12 hardware-induced fault categories, versus 93.6 for MVT-Flow, and has the lowest across-category dispersion among eight methods. On the added-mass category the setting-level reading rises with independently estimated load. Removing the action lowers AUROC by 2.49 points, while state-conditioned action shuffling lowers it by 0.70. The result is a way of reducing a complex regulated system by derivation rather than by search — demonstrated on one arm, and claimed no more widely.

Index Terms—Failure Detection and Recovery; Robot Safety; condition monitoring; information theory.

I. INTRODUCTION

Condition monitoring of an industrial robotic arm faces two obstacles that pull against each other. Modelling the arm’s physics is expensive; avoiding that by scoring the arm’s output instead runs into a controller actively removing the evidence from the output. This paper is about a third position that answers both.

The first obstacle: the arm is complex to model. The rigid-body equations are textbook; the quantities that make a real arm depart from them are not. Only a minimal base set of the inertial parameters is identifiable at all [12]; recovering it requires exciting trajectories run off-task and repeated whenever the payload or tool changes [13], [14]; and harmonic-drive joints add flexibility, hysteresis and load-dependent friction that remain an active modelling topic rather than settled engineering [15]. Haddadin, De Luca and Albu-Schäffer list modelling error, friction torque and unmodelled link flexibility among the four disturbances that limit residual-based detection, and note that unmodelled flexibility produces false positives directly [11]. The field’s response has been to model less of the arm and more of its data: learned inverse dynamics replaces the physics with a fitted plant [17], and trajectory-level anomaly detection abandons the plant model entirely, scoring how far a recorded cycle falls from a corpus of healthy ones [1].

The second obstacle: the controller hides the fault. Scoring the output meets a difficulty with a name. Under feedback, a developing fault is compensated rather than expressed, so the closed-loop system “may behave similarly to when it is in healthy conditions,” and degradation can stay undetectable until severe failure [9]. Control loops “mask severe fault modes while compensating for typical disturbances,” which is precisely what prevents early detection of a developing failure [10]. The robot-specific statement is blunter still: mechanical failures such as bearing wear “may go unnoticed as control algorithms compensate by increasing motor current” [16]. Two of the twelve anomaly categories in vora-us-AD are degradations of exactly this kind, produced physically by attaching foam at a joint transition and additional mass to each axis [1].

Closed-loop fault masking is usually treated as a limit on what monitoring can achieve. We read it as a statement about where the evidence went: a controller holding the trajectory under a growing disturbance must do more to hold it, so the fault is not destroyed but *relocated* — out of the output and into the relation between what was commanded and what followed. That relation is a property neither of the arm nor of its output, but of the loop, and it is what this paper takes as the object to model.

Table I sets out why that position is vacant, ordering the established approaches by what must be in hand before a reading exists: identified dynamics, an excitation campaign, a covering dataset, a healthy trajectory corpus, a fault signature. Among the methods evaluated on this benchmark they do not differ in subject: every one models the arm, or the output the arm produced — and the output is where the controller has been clearing. Input–output and closed-loop fault diagnosis do relate inputs to outputs; what is claimed here is narrower, an explicit discrete $p(s' | s, a)$ factorisation read without optimisation on raw industrial telemetry.

The construction is the table’s last row. Four scalar residuals per joint form a discrete state S ; the realised torque-forming current forms a discrete action A ; the same state one step later closes the triple with S' . Over a causal window the joint distribution of (S, A, S') supports a battery of ten information measures, each read as a deviation from a reference frozen on healthy runs. No iterative optimisation and no learned weights are involved, though healthy-data calibration is required: the 360 figure counts phase-specific reference moments, alongside standardisation and bin-edge statistics also estimated from healthy runs.

What makes such a model specifiable in advance is the

TABLE I

APPROACHES TO CONDITION MONITORING OF AN INDUSTRIAL ARM, ORDERED BY DECREASING COMMITMENT TO AN EXPLICIT MODEL OF THE ARM. THE DISTINGUISHING QUESTION IS NOT ACCURACY BUT WHAT EACH APPROACH MUST ALREADY POSSESS BEFORE IT CAN PRODUCE ONE READING. THE SEVEN PUBLISHED VORAUS-AD BASELINES MODEL ARM DYNAMICS OR HEALTHY OUTPUT — AND UNDER FEEDBACK THE OUTPUT IS WHERE THE FAULT IS MASKED. THE LAST ROW EXPLICITLY FACTORISES THE OBSERVED CONTROL LOOP.

Approach	Required before the first reading	What is modelled	Parameters held at run time
Dynamic-model residual <i>momentum observer</i> ; <i>inverse-dynamics residual</i> [8], [11]	Identified rigid-body dynamics and a friction model, accurate to better than the fault sought	Arm dynamics	~50–70 identified physical parameters
Parameter identification <i>regressor least squares</i> [12], [13]	The model structure, plus an off-task excitation campaign, repeated on payload or tool change	Arm dynamics	~50–70, fitted by least squares
Learned inverse dynamics <i>Gaussian-process or neural plant model</i> [17]	A dataset covering the operating envelope, plus a retraining policy	Arm dynamics	10^4 – 10^6 learned weights, or $\mathcal{O}(N)$ retained support points
Trajectory deviation <i>I-NN, PCA, CAE, LSTM-VAE, GANF, HMM, MVT-Flow</i> [1]	A healthy trajectory corpus, an architecture, an optimisation run and a threshold policy	The distribution of healthy output	up to 1,015,560 learned weights (MVT-Flow)
Drivetrain signal analysis <i>motor-current signature analysis</i> [18]	A fault signature to look for	One channel’s spectrum	Signature templates and thresholds
This work <i>discrete (S, A, S') loop statistics</i>	Healthy loop statistics, and three discretisation and temporal parameters	The control loop	360 reference moments, none fitted or learned

criterion used to build it: which of the 130 channels to read, and at what resolution, follows from an information argument — bin counts allocated by Lipschitz sensitivity, a window long enough to occupy the joint alphabet — leaving three free parameters. No anomaly data enters the estimator’s construction, though the deployed configuration is *configuration-informed*, four further choices having consulted anomaly outcomes (Section VI-C).

The evidence is obtained on real machine data: voraus-AD is recorded on a physical six-axis collaborative arm running a pick-and-place cycle, with all twelve anomaly categories induced on the hardware itself [1], on the authors’ published split and metric. This matters because the field’s own survey reports that physics-based simulation “often fail[s] to produce data that genuinely reflects real-world behavior,” and that validation on real industrial data “remain[s] limited” [19].

Four contributions follow, in order of what they commit to.

- 1) **Conceptual.** Closed-loop fault masking is read not as a limit on monitoring but as relocation: under feedback the evidence of a developing fault leaves the output and enters the relation among state, realised action and next state. That reading is what identifies the object to monitor.
- 2) **Methodological.** A principled reduction of that relation: an information criterion, not a search, fixes which signals to read and at what resolution, leaving three discretisation and temporal parameters and 360 phase-reference moments, with no iterative optimisation and no learned weights, one reading every 300 transitions (3.03 s) at 0.75 ms per control cycle.
- 3) **Benchmark-grounded account.** A survey of modelling commitment (Table I) identifying an untested position on voraus-AD: explicit factorisation of the observed control loop rather than of arm dynamics or undifferentiated

trajectories.

- 4) **Empirical.** 88.6 mean AUROC (bootstrap floor 86.3) on the published split and metric, the same configuration recovered in 12 of 12 leave-one-category-out folds, the most uniform performance across the twelve fault types of eight methods, and a setting-level reading that rises with independently estimated payload — offered as sufficiency for the first three, not as a placement on the benchmark.

II. HOW AN ARM IS MODELLED FOR MONITORING

This section works through the rows of Table I. The ordering is by decreasing commitment to an explicit model of the arm, and the question asked of each is the same: what must be in hand before a reading exists.

A. Model-based residuals

The classical instrument compares the torque the arm drew with the torque a dynamic model says the commanded motion required. The generalised momentum observer is the canonical form, and its practical strength is that it needs only proprioceptive sensing [8], [11]. Its cost is the model: the inertia matrix, the Coriolis and gravity terms and a friction model, each accurate to better than the fault sought. Haddadin et al. enumerate the four disturbances that erode this margin — measurement noise, modelling error, friction torque and unmodelled link flexibility — and observe that the last produces false positives on its own [11].

B. Identification and learned plant models

The grey-box response is to stop assuming the parameters and estimate them. Only a base set is identifiable [12]; recovering it needs excitation trajectories designed for the purpose and run off-task, and the repeatability of the recovered friction and

inertia terms is itself a reported result [13], [14]. The black-box response is to replace the physics with a fitted plant — Gaussian-process or neural inverse dynamics — which trades the identification campaign for a dataset covering the operating envelope and a retraining policy [17]. Both remain models of the arm.

C. Drivetrain signal analysis

Plant-floor practice reads one channel rather than a model: motor-current signature analysis and vibration monitoring on the joint or gearbox, matched against known fault signatures [18]. The commitment moves from a model to a signature, and the subject narrows from the arm to one of its channels.

D. Trajectory-level deviation

The line the voraus-AD board occupies drops the plant model altogether and models the distribution of healthy output. voraus-AD [1] provides 2,122 recordings of a six-axis pick-and-place cycle: 1,367 normal and 755 anomalous across twelve categories, at 100 Hz over 130 signals. The machinery of the seven baselines differs; the premise does not. Each scores how far a recording departs from healthy joint trajectories, taken as a whole: ℓ_1 distance to the closest training recording [2], the residual off a 90-component subspace [3], ℓ_2 reconstruction error [4], negative log-density under a fitted model [5], [6], or a distance between an empirical and a fitted emission distribution [7]. Some are likelihoods and some are deviation scores, but none factorises the recording into state, action and next state; all read the trajectory as one undifferentiated object.

That premise is right whenever a fault writes itself into the signal, which is why the normalising-flow method reaches 99.8 AUROC on miscommutation — an electrical fault that alters the current waveform outright — and 96.6 on added friction.

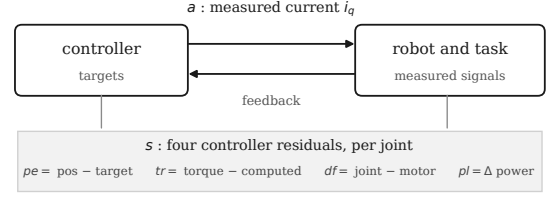
E. The commanded action is present and discarded

The dataset ships the controller’s commands — target position, target velocity, target torque, computed torque, per axis — alongside the measured quantities. Every baseline consumes them as undifferentiated columns of one matrix. The PCA reference states it plainly, describing its input as dimensions “comprising control input and measurements” [3]. No method on this board factorises the problem as $p(s' | s, a)$. The information that distinguishes a command from its consequence is present in the file and is not used as such by anything that reads it.

F. Causality is a property of the protocol

One sentence governs every published baseline: all methods receive the recording padded to a uniform length, and the anomaly decision is a function of the whole padded episode. The unit of detection is one completed operation for all seven, and no per-timestep score is defined anywhere. This is a property of the board, not of the methods — four of the seven are causal in their original formulations [3], [5], [7] — with one exception: the normalising-flow method uses centred padding over the time axis, so the value at t depends on

(a) the loop, as the telemetry reports it



(b) one reading; window t uses samples up to t only

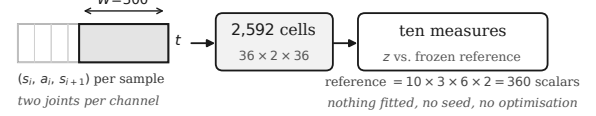


Fig. 1. The construction. (a) The commanded targets are part of the state, not of the action: they are what the controller regulates towards, not what it emits. The action is what was actually done. (b) A reading is a histogram over 2,592 cells and a comparison against a reference of 360 scalars. No parameter is optimised.

samples after t . The millisecond figures reported alongside the benchmark are the cost of scoring a completed recording; no detection latency is reported for any method.

G. Why the loop has not been the subject

Conditioning on a known input is standard on the model-based side, where a momentum observer tests whether observed motion is consistent with the commanded torque [8]; what that tradition requires is the explicit rigid-body model, which is precisely what the data-driven line set out to avoid. On the data-driven side the natural quantity is empowerment, the channel capacity from an agent’s actuators to its later sensor states [22], [23]; its maximisation is exponential in the horizon and has no analytic solution for a general continuous channel [23], and it has been applied to exploration, assistance and morphology design rather than to diagnosis. Section III-A returns to why a capacity is in any case the wrong quantity for a diagnostic. Elsewhere, information theory has entered monitoring on tracks that are not the loop: univariate complexity measures on one signal, and transfer entropy localising a disturbance already detected [24]. The nearest work to diagnosis retains one state–action coupling term among eight metrics for predicting policy failures [25]; related work tracks mutual information among state, action and next state in reinforcement-learning telemetry [26].

III. METHOD

A. Why an information criterion

Ashby framed a regulator as a communication channel in 1958 and argued that its regulatory capacity cannot exceed its capacity as a channel [20]; Touchette and Lloyd gave the bound a physical form, showing that the entropy a closed-loop controller can remove exceeds open-loop control by at most the mutual information the controller obtains [21]. The natural formalisation of “how much can this loop carry” is therefore a capacity — empowerment [22] — and a capacity is a maximum over input distributions, which is what has kept it out of practice and, more importantly, makes it the wrong

quantity here: it measures what a machine *could* do, under a controller assumed “free to act” [23].

A diagnostic needs what the machine *is* doing, and that needs no maximisation. The realised joint distribution of state, action and next state is already in the telemetry; normalising by the entropies it contains yields a bounded, comparable number obtained by counting. The same criterion does the model selection — which channels are read, at what resolution — which is why the construction below has three free parameters rather than a search space.

B. State, action, next state

Fig. 1 shows the construction. The state is built from *controller residuals*: differences between two signals the controller relates to one another, self-contained once the pair is given. They use the commanded targets only as the reference a residual is taken against; no residual is defined by where the robot was asked to go in absolute terms. Four per joint:

<i>pe</i>	joint position — target position
<i>tr</i>	torque sensor — computed torque
<i>df</i>	joint position — motor position
<i>pl</i>	motor mechanical power — load mechanical power

The action is the realised torque-forming current i_q . Everything here is read at the interface the drive exposes: the trajectory layer emits per-axis targets and a computed feedforward torque, and the current layer reports the i_q the drive actually ran. The benchmark lists i_q as an *estimated* signal — derived from the measured phase currents — which makes it a realised quantity rather than a commanded one, and that is the property the construction needs. Targets enter only as the reference inside a residual; they are never the action. Six joints times eight raw signals gives 48 of the 130 available. Each variable is standardised against healthy recordings and discretised on equal-width bins spanning $\pm 2.5\sigma$.

C. Uneven bin allocation

Resolution is not distributed uniformly. Each variable is perturbed by $\pm 0.25\sigma$ and the induced change in the measure recorded, giving a per-variable sensitivity L_j ; bins are allocated in proportion, yielding $pe(2)$, $tr(3)$, $df(3)$, $pl(2)$ — 36 states, 2 actions, 36 next states, 2,592 cells. The allocation is derived on healthy recordings only.

D. The battery

Over a window of W transitions the empirical joint distribution of (S, A, S') supports ten measures, of which the central one is

$$P = \frac{I(S, A; S')}{H(S) + H(A) + H(S')}, \quad (1)$$

alongside $I(S, A; S')$, $H(S)$, $H(A)$, $H(S')$, the forward and backward conditional entropies, their difference, $I(S; A)$, and $\Delta H_{S \rightarrow S'} = H(S') - H(S)$. All entropies carry the Miller–Madow correction $(K_+ - 1)/(2N \ln 2)$, where K_+ is the number of cells with a positive count in that histogram [27].

E. Windowing, reference and reading

W counts *transitions*, not samples: $W = 300$ transitions is formed from 301 consecutive samples per joint, S at positions $1 \dots W$ and S' at $2 \dots W+1$, so two joints per channel give $2W = 600$ triplets per histogram and the earliest readout needs 301 samples — 3.03 s at the dataset’s median interval of 10.067 ms. Windows are causal: window t uses samples up to t only. The step ahead is $\Delta = 1$, the minimum causal lag — a design choice rather than a tuned parameter, with no claim that mutual information is monotone in Δ in general. Joints are grouped into three channels of two, each with its own histogram; channel readings are averaged.

Each window’s measures are converted to deviations from a mean and standard deviation frozen on healthy training recordings at the same window position. The entire healthy reference is therefore ten measures $\times$ three channels $\times$ six window positions $\times$ two moments: **360 scalars**. No quantity here is fitted or learned: there is no iterative optimisation and no random initialisation, and the 360 moments are obtained by averaging over healthy runs. Healthy calibration is required; model fitting is not. The reading itself is a histogram and a division.

A recording’s score is, per measure, the mean absolute deviation over its windows, taken at the worst channel, then averaged over the ten measures.

IV. EVALUATION PROTOCOL

We adopt the benchmark’s published split and metric. Four configuration choices were made with reference to anomaly outcomes and are enumerated in Section VI-B, so the headline figure is reported throughout as *configuration-informed*.

- Calibration uses normal training recordings only: no anomalous recording enters the reference, and no held-out recording contributes to bin placement.
- The split is theirs — 948 normal training recordings, 419 held-out normals as negatives, 755 anomalous as positives — with one score per recording, AUROC per category against the same 419 negatives, and an unweighted mean over the twelve categories, as in the original.

The benchmark reports AUROC alone; no threshold or false-alarm rate is published for any method, and none is reported here.

V. RESULTS

A. What the model costs

The healthy reference is ten measures $\times$ three channels $\times$ six window positions $\times$ two moments: 360 scalars, obtained by averaging. The benchmark’s leading method fits four coupling blocks, each carrying a $\text{Conv1d}(65 \rightarrow 260, k=13)$ and a $\text{Conv1d}(260 \rightarrow 130, k=1)$ over 130 signals — **1,015,560 parameters**, computed from the released architecture and its published configuration, trained for 70 epochs on 948 recordings. Both must be obtained from healthy data; what differs by a factor of about 2,800 is how much is retained, and what differs in kind is whether obtaining it requires iterative optimisation. At read time one channel reading over 600 triplets costs 42 μs , all three channels 125 μs , and a whole

TABLE II

PER-CATEGORY AUROC (%) ON VORAUS-AD; BASELINE COLUMNS AS PUBLISHED, BEST PER ROW IN BOLD. [†]AUTHOR-DERIVED FROM THE TWELVE PUBLISHED CATEGORY AUROCS. THE 95% INTERVAL IS A STRATIFIED BOOTSTRAP OVER RECORDINGS (10,000 RESAMPLES WITHIN CATEGORY AND WITHIN THE HELD-OUT NORMALS); THE BENCHMARK PUBLISHES NO PER-RECORDING SCORES, SO NO EQUIVALENT INTERVAL CAN BE FORMED FOR ITS BASELINES.

Anomaly	n	Ours	1-NN	PCA	CAE	LSTM-VAE	GANF	HMM	MVT-Flow
Additional friction	144	86.9	74.8	76.4	89.4	88.7	88.5	88.0	96.6
Miscommutation	89	87.7	80.8	87.0	99.1	98.1	98.8	93.7	99.8
Misgrip of can	11	98.2	100.0	100.0	100.0	100.0	47.6	71.0	95.3
Losing can	74	85.9	68.7	70.1	72.6	70.4	72.1	88.8	96.2
Additional axis weight	156	85.5	75.0	79.2	93.5	82.7	93.2	89.6	94.1
Collision with foam	72	91.0	69.6	73.9	81.5	81.5	81.2	89.8	87.5
Collision with cables	48	93.0	74.5	75.7	79.6	77.3	82.7	91.8	84.7
Collision with cardboard	22	81.6	82.8	83.6	78.6	82.8	77.5	86.3	88.3
Varying can weight	80	84.8	63.7	64.2	72.3	71.4	68.9	90.9	85.1
Cable routed at robot	10	89.2	63.0	71.6	83.1	96.0	76.6	84.4	100.0
Invalid gripping position	12	93.6	93.4	92.1	88.8	97.7	86.6	91.8	100.0
Unstable platform	37	86.2	83.6	85.9	83.9	93.6	84.6	82.5	96.1
Mean	755	88.6	77.5	80.0	85.2	86.7	79.9	87.4	93.6
95% CI (bootstrap)		[86.3, 90.8]	—	—	—	—	—	—	—
SD across categories [†]		4.6	11.3	10.1	9.2	10.5	13.2	6.1	5.7
Signals used		48	130	130	130	130	130	130	130
Iterative optimization		none	—	fit	60 ep.	90 ep.	fit	fit	70 ep.

cycle at stride 150 costs 0.75 ms, on one core of a commodity ARM CPU with no accelerator. The operation is a `bincount` and a division.

B. Time to first readout

W counts *transitions*, so a reading requires $W + 1 = 301$ samples and yields $2W = 600$ triplets across the two joints of a channel. At the dataset’s median sample interval of 10.067 ms the first readout is therefore available 3.03 s into a cycle of roughly 11 s, and a further one every 1.51 s; at $W = 10$ the first arrives at 0.10 s and the mean falls by 2.5 points (Fig. 2). The benchmark’s leading method has no position on this axis at all, its score being defined only over the completed episode.

Two limits, stated rather than worked around. The AUROC reported here aggregates all of a recording’s windows, so it measures separability given the whole cycle and does not by itself establish a decision at 3.03 s; prefix-restricted evaluation is future work. And the dataset labels no fault onset, so detection latency cannot be computed on it for any method, ours included. We therefore report a first *readout* time — a property of the construction — and make no claim about a verdict.

C. Sensitivity to the three parameters

Three discretisation and temporal parameters are analysed here: bin allocation, window length and Δ . They are not the whole configuration — Section VI-B lists the four further choices that saw anomaly outcomes. Δ is fixed at 1 by the argument in Section III-D. Window length was swept from 10 to 450 transitions (Fig. 2), on the same calibration as the headline result; the mean varies between 85.3 and 88.6 with no cliff and a plateau from 100 onward. $W = 300$ and $W = 450$ tie at 88.6, and 300 is taken because it reaches that value at half the latency. Beyond 450 the recordings become the limit: at $W = 900$ only two windows remain per recording and

the mean falls to 84.0, a window-count effect rather than a window-length one.

Bin allocation has a cliff on either side: six bins on every state variable, giving 3.36 million cells against 600 samples, costs 13.3 points, while collapsing the window to a single transition still returns 83.5 — the floor of the construction, not chance. Two controls bound the result. Replacing the state with three signals the controller does not act on returns 22.0; an AUROC below 50 is a *reversed* ordering rather than an absence of information, which is what a monitor reading quantities outside the loop should be expected to produce. Destroying the temporal correspondence between state and next state while preserving both marginals removes roughly three quarters of the separation, locating the result in the conditional structure.

Estimator support. The nominal alphabet is $36 \times 2 \times 36 = 2,592$ cells and each histogram holds $N = 600$ triplets; the Miller–Madow correction uses the *occupied* support K_+ , not the nominal alphabet size, and on healthy runs K_+ averages 84.2, of which 39.4 hold a single count. The reported values are therefore empirical information scores on a declared alphabet, not estimates of a population entropy. Three estimators on the frozen configuration — plug-in 88.8, Miller–Madow 88.6, Krichevsky–Trofimov 87.5 — span 1.3 points, leaving every conclusion unchanged.

Stride is not an accuracy parameter while enough windows remain: a tenfold increase in window count at $W = 10$, or threefold at $W = 300$, moves the mean by 0.2 points, overlapping windows sharing most of their samples.

D. Adequacy on the benchmark

Table II gives per-category results. The mean of 88.6 places the method ahead of six of the seven published baselines — HMM 87.4, LSTM-VAE 86.7, CAE 85.2, PCA 80.0, GANF 79.9, 1-NN 77.5 — and 5.0 points behind the benchmark’s normalising-flow method at 93.6. Two categories lead every

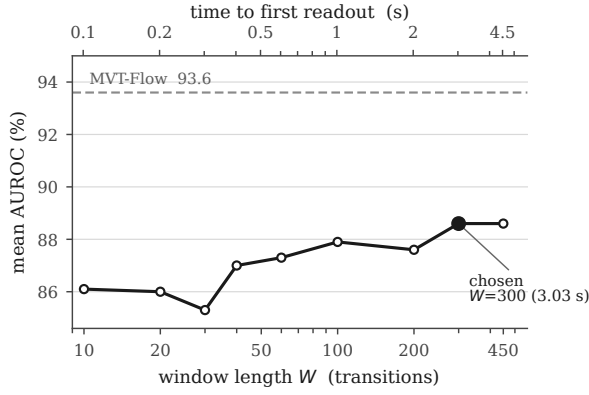


Fig. 2. Mean AUROC against window length, and so against the earliest instant a reading exists. Performance is stable across the range and approximately flat from $W = 100$ onward; the mean varies by 3.3 points over the full sweep. $W = 300$ transitions (3.03 s of an ≈ 11 s cycle) and $W = 450$ tie at the top, so 300 is taken for its shorter latency. The benchmark leader has no position on this axis: its score is defined only over the completed episode.

published method: collision with foam at 91.0 against 87.5, and collision with cables at 93.0 against 84.7.

A stratified paired bootstrap over recordings puts the macro mean at 88.6 with a 95% interval of [86.3, 90.8]. That interval does not reach the published 93.6; it is not an interval for the *difference* from MVT-Flow, which would require that method’s per-recording scores. Three categories carry only 10 to 12 positives and their intervals are correspondingly wide — cable routed at robot spans 9.8 points — a property of the benchmark rather than of any method on it.

Taking the standard deviation of each method’s twelve published category AUROCs gives 1-NN 11.3, PCA 10.1, CAE 9.2, LSTM-VAE 10.5, GANF 13.2, HMM 6.1, MVT-Flow 5.7 and, for the present method, **4.6**: the most uniform across fault types of the eight. These are author-derived from the published per-category values; they land close to the $\pm$ figures in the benchmark’s mean row, but we do not rely on that correspondence and make no claim about what those figures denote. Separately, the measurement is deterministic — repeated evaluation on identical input returns identical values, there being no random initialisation and no sampling anywhere in it. That is repeatability, not a statement about sampling uncertainty, which the bootstrap interval supplies.

E. What the action contributes

The chain rule splits the numerator, $MI(S, A; S') = MI(S; S') + MI(A; S' | S)$, and on a rigid arm running a fixed programme the first term dominates: 2.212 bits against 0.068 on healthy runs. State persistence is most of the predictive structure, as it must be for a steel mechanism under gravity. The question is what the action adds on top.

Two controls answer it, both on the frozen configuration and paired by recording. *A-shuffled* permutes the action within strata of S inside each window, leaving $p(a)$, $p(a | s)$, $MI(S; A)$ and the whole (S, S') relation untouched and destroying only what the action adds beyond the situation. *No-A* removes the action entirely.

Detector	Mean	Increment (95% CI)
Full	88.6	—
<i>A</i> shuffled within S	87.9	+0.70 [+0.38, +1.03]
No action at all	86.2	+2.49 [+1.75, +3.27]

Both increments are positive in 100% of 5,000 paired bootstrap resamples and are computed on unrounded paired values. The +2.49 figure is the combined contribution of *all* action-dependent statistics under this detector; +0.70 is narrower, the conditional action-alignment contribution — what the action adds about the *outcome* once the situation is given. The gap is the shape expected if compensation is doing the work: under fault the controller holds its error and pushes differently, so much of the change appears in how it acts rather than in what the action achieves.

F. Which measures carry the result

The mutual information between the state–action pair and the next state reaches 92.3 on its own, 1.3 points from the benchmark leader. Two measures carry nothing, and healthy data alone says why: $\Delta H_{S \rightarrow S'} = H(S') - H(S)$ is pinned at its floor, since S' is S one sample later and in a stationary window their marginals cannot differ (healthy mean 0.0070 bits against measures of order 1 to 3 bits), while $H(A)$ is pinned at its ceiling, the binary action alphabet sitting at 0.977 of its maximum. Removing both moves the mean by -0.3 points. We nonetheless report the ten-measure average, since excluding a measure on the strength of its AUROC would be selection on fault data; the exclusion argument uses only healthy statistics.

G. Whether the reading scales with the disturbance

Detection separates faulty from healthy; an instrument is more useful if the reading also tracks how much the loop is absorbing. The benchmark’s authors attached 115 g, 231 g or 500 g to each of the six axes [1], but which setting is which is undocumented. We recovered it from the machine: subtracting the healthy mean joint torque at each action phase leaves a residual load, and the eighteen settings fall into six axis-major triples, ranked within each axis by that load. Mass is assigned per setting, so the setting — not the run — is the unit of analysis throughout.

Two independence checks precede the result. Re-deriving the labels from `torque_sensor_b`, which is *not* among the detector’s 48 signals, agrees in 16 of 18 settings — the exceptions inside the base-axis triple, where loads differ by under 0.05 Nm — with the two estimates correlating at $\rho = 0.992$. The trend was then recomputed on cached state variants excluding torque entirely.

State	ρ	p	heaviest top
<i>pe, tr, df, pl</i> (deployed)	+0.56	0.042	5/6
<i>pe, ve</i> (torque-free)	+0.83	$< 10^{-4}$	6/6
<i>pe</i> only	+0.77	$< 10^{-4}$	6/6
<i>tr, df, pl</i> (no <i>pe</i>)	+0.46	0.104	5/6

Removing torque *strengthens* the trend and removing the position-error residual removes it — the opposite of a shared-signal artefact. On the deployed configuration the setting-level mean reading is 1.374, 1.575 and 2.365 by ascending

mass, the within-axis rank correlation $+0.56$ with $p = 0.042$ from exact enumeration of all $6^6 = 46,656$ within-axis label arrangements, and an axis-clustered interval of $[0.11, 0.87]$, positive in 98.6% of draws.

Why only one category. Correlating the reading against the setting index is invalid: miscommutation is graded by *axis* and carries no severity, yet the index-ordered test returns $+0.66$ on it. Index is confounded with axis, and distal axes deviate more. Index-ordered results for the three other graded categories are therefore not reported.

VI. DISCUSSION

A. Where the two assumptions differ

Modelling the loop rather than the output predicts that the advantage should concentrate where the controller absorbs the disturbance. The two largest margins over the benchmark leader are collision with cables ($+8.3$) and with foam ($+3.5$), both compliant contacts; the largest deficits are miscommutation (-12.1) and cable routed at robot (-10.8), where the fault is written into the signal directly. Across all twelve the association with a proxy for marginal visibility runs in the predicted direction without reaching significance (Spearman -0.51 , $n = 12$; -0.59 required at $\alpha = 0.05$), so it is offered as consistent with the account rather than as an established mechanism.

B. Reading the severity trend

The trend was hypothesised before it was measured, from the account that motivates the construction: a regulator absorbs variety in proportion to the disturbance it rejects [20], so a heavier mass should displace the loop’s statistics further from healthy. We state that as a hypothesis the data are consistent with, not an entailment — requisite variety constrains regulation, and does not determine the monotonicity of a discretised, ten-measure aggregate under binning and channel aggregation. What the result establishes is narrower: on one machine, in one fault category, the reading rises with the mass the loop is carrying, most clearly when the detector reads no torque at all.

C. Where the configuration came from

Every configuration choice is listed below with the evidence that set it, since some consulted anomaly outcomes and some did not.

Choice	Value	Set by
Δ	1	causal lag
Bins	2/3/3/2	healthy L_j
Span	$\pm 2.5\sigma$	a priori
Reference	moments	healthy runs
State signals	pe, tr, df, pl	anomaly AUROC
Grouping	3 of 2	anomaly AUROC
Rule	worst-then-mean	anomaly AUROC
Window	$W = 300$	anomaly AUROC

The four in the lower block are exposed to the evaluated anomalies. Two leave-one-category-out analyses re-derive them blind. For the state signals, grouping and rule, the selection was repeated twelve times over 54 candidate configurations, each fold blind to one category and scoring only

that category: **all twelve folds select the same state signals and the same grouping**, only the rule varies, and the mean over the twelve held-out categories is **88.1** against an all-data optimum of 89.0. For W , the same procedure over the sweep gives **87.8**, with every fold choosing $W = 300$ or $W = 450$ — the two values that tie at the top. The reported configuration is third of the 54 rather than the maximum.

For contrast, the classical statistical-process-control baseline loses 13.3 points when made blind (73.5 against an oracle 86.8), its oracle selecting a different signal per category. A fully joint leave-one-out over all four choices was not run; the two marginal analyses are reported in its place.

D. Limitations

The result is 5.0 points behind the benchmark leader and is not a state-of-the-art claim; adequacy, not placement, is what it is offered to establish. Four configuration choices were made with reference to anomaly outcomes; both leave-one-category-out analyses above address them, but neither is a joint control over all four simultaneously, and 88.6 should be read as a configuration-informed result. The scope of the recipe is a repeated-cycle geared manipulator with joint-level telemetry — controller residuals, a realised actuation quantity and a reference frozen at each window position all presuppose that setting; the dataset provides one such robot performing one task, so nothing here establishes transfer even within it. The discretisation is coarse by design, and how far the result depends on that coarseness beyond the reported sweep is not established. Finally, we report AUROC only, as the benchmark does; no threshold, operating point or false-alarm rate is published for any method on it, and none is claimed here.

The severity result rests on one category, three mass levels and eighteen settings on one machine, at $p = 0.042$ on the deployed configuration; it shows the reading rises with the disturbance, not that it grades every fault type, is linear, or supports a remaining-life estimate. The dataset labels no fault onset, so nothing here bears on how early a developing fault becomes visible — the question a predictive-maintenance deployment would ask.

We also do not compare against a matched physics model on the same four residuals: such a comparison would inherit a representation that exists only because the information criterion produced it. What is under test here is the criterion, not a score — a selection procedure that had every opportunity to return a representation discriminating nothing, and did not. A constructive reduction of this kind is reproduced rather than ablated.

VII. CONCLUSION

Monitoring an industrial arm has meant modelling the arm, or modelling what the arm produced. Under feedback the second of those is the place a developing fault is least visible, because the controller has been removing it there. This paper models the loop instead — the relation the fault was displaced into — and the point of doing so is how little model it needs: three parameters, 360 reference moments, no fitted or learned quantity, 48 of 130 signals, a reading 3.03 s into the cycle

rather than after it, and a value reproducible to the decimal. On real machine data with faults induced on the hardware, that model reaches 88.6 mean AUROC on a benchmark whose leader reaches 93.6, and its reading rises with the mass the loop is carrying — enough to establish that the loop is a legible object, which is what it was built to show.

The wider point is not the score but how it was arrived at. The representation was not searched for on held-out faults; it follows from a criterion — which signals carry the loop’s regulatory state, and at what resolution — that could have produced something discriminating nothing and did not. That is a way of reducing a complex regulated system by derivation rather than by search, and nothing in the derivation mentions robots. We nevertheless claim it no more widely than we have shown it: one arm, one task, one benchmark. Whether the reading is also *non-redundant* with trajectory-level scores is not settled by separate AUROCs, and is left as future work with prefix-restricted evaluation and transfer to a second machine.

DECLARATIONS

Data availability. All results use the public voraus-AD dataset [1], released under CC BY-NC-SA 4.0 and used here for non-commercial research only, following the authors’ published split and metric. No new data were collected. **Code availability.** The pipeline, the parameter sweeps and the scripts reproducing every table and figure are provided as anonymised supplementary material and will be released on acceptance. **Reproducibility.** The method contains no fitted or learned parameters and no random number generation, so repeated evaluation on identical input returns identical values; this is exact repeatability and makes no claim about sampling uncertainty, which is reported as bootstrap intervals in Table II. **Ethical approval.** No human subjects, personal data or animal experimentation. **Conflicts of interest.** None declared.

REFERENCES

- [1] J. Brockmann, M. Rudolph, B. Rosenhahn, and B. Wandt, “The voraus-AD dataset for anomaly detection in robot applications,” *IEEE Trans. Robotics*, vol. 40, pp. 438–451, 2024.
- [2] M. Amer and M. Goldstein, “Nearest-neighbor and clustering based anomaly detection algorithms for RapidMiner,” in *Proc. RapidMiner Community Meeting and Conf.*, 2012.
- [3] R. Hornung, H. Urbanek, J. Klodmann, C. Osendorfer, and P. van der Smagt, “Model-free robot anomaly detection,” in *Proc. IEEE/RSJ IROS*, 2014, pp. 3676–3683.
- [4] T. Chen, X. Liu, B. Xia, W. Wang, and Y. Lai, “Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder,” *IEEE Access*, vol. 8, pp. 47072–47081, 2020.
- [5] D. Park, Y. Hoshi, and C. C. Kemp, “A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1544–1551, 2018.
- [6] E. Dai and J. Chen, “Graph-augmented normalizing flows for anomaly detection of multiple time series,” in *Proc. ICLR*, 2022.
- [7] D. Azzalini, A. Castellini, M. Luperto, A. Farinelli, and F. Amigoni, “HMMs for anomaly detection in autonomous robots,” in *Proc. AAMAS*, 2020, pp. 105–113.
- [8] A. De Luca and R. Mattone, “Sensorless robot collision detection and hybrid force/motion control,” in *Proc. IEEE ICRA*, 2005, pp. 999–1004.
- [9] L. Zhang, H. Yang, K. Dandago, and Y. Li, “The negative impact of closed-loop control on fault detection,” in *Proc. 4th Int. Conf. on Advanced Engineering, Technology and Applications on Power Systems*, 2026, pp. 360–373.
- [10] C. Sconyers, Y.-K. Lee, K. Kim, S. Oh, D. Mavris, N. Oza, R. Mah, R. Martin, I. A. Raptis, and G. J. Vachtsevanos, “Diagnosis of fault modes masked by control loops with an application to autonomous hovercraft systems,” *Int. J. Prognostics and Health Management*, vol. 4, no. 1, 2013.
- [11] S. Haddadin, A. De Luca, and A. Albu-Schäffer, “Robot collisions: a survey on detection, isolation, and identification,” *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1292–1312, 2017.
- [12] M. Gautier and W. Khalil, “Direct calculation of minimum set of inertial parameters of serial robots,” *IEEE Trans. Robot. Autom.*, vol. 6, no. 3, pp. 368–373, 1990.
- [13] L. Ding, H. Wu, Y. Yao, and Y. Yang, “Dynamic model identification for 6-DOF industrial robots,” *J. Robotics*, vol. 2015, art. 471478, 2015.
- [14] A. Raviola, R. Guida, A. De Martin, S. Pastorelli, S. Mauro, and M. Sorli, “Dynamic and friction parameters of an industrial robot: identification, comparison and repetitiveness analysis,” *Robotics*, vol. 10, no. 1, art. 49, 2021.
- [15] S. Kim, J. Lee, and H. Kim, “Analytical modeling method for joint friction in harmonic drive robot,” *J. Mech. Sci. Technol.*, vol. 38, 2024.
- [16] R. Guida, A. C. Bertolino, A. De Martin, and M. Sorli, “Comprehensive analysis of major fault-to-failure mechanisms in harmonic drives,” *Machines*, vol. 12, no. 11, art. 776, 2024.
- [17] D. Nguyen-Tuong and J. Peters, “Model learning for robot control: a survey,” *Cognitive Processing*, vol. 12, no. 4, pp. 319–340, 2011.
- [18] A. H. Sabry and U. A. B. Ungku Amirulddin, “A review on fault detection and diagnosis of industrial robots and multi-axis machines,” *Results in Engineering*, vol. 23, art. 102397, 2024.
- [19] M. U. Elahi, R. T. A. Khan, M. H. Yazdani, and H. S. Kim, “Unsupervised learning for industrial robot health monitoring: trends, techniques, and challenges,” *Mathematics*, vol. 14, no. 13, art. 2397, 2026.
- [20] W. R. Ashby, “Requisite variety and its implications for the control of complex systems,” *Cybernetica*, vol. 1, no. 2, pp. 83–99, 1958.
- [21] H. Touchette and S. Lloyd, “Information-theoretic limits of control,” *Phys. Rev. Lett.*, vol. 84, no. 6, pp. 1156–1159, 2000.
- [22] A. S. Klyubin, D. Polani, and C. L. Nehaniv, “Empowerment: a universal agent-centric measure of control,” in *Proc. IEEE Congress on Evolutionary Computation*, 2005, pp. 128–135.
- [23] C. Salge, C. Glackin, and D. Polani, “Empowerment — an introduction,” in *Guided Self-Organization: Inception*. Springer, 2014, pp. 67–114.
- [24] M. Bauer, J. W. Cox, M. H. Caveness, J. J. Downs, and N. F. Thornhill, “Finding the direction of disturbance propagation in a chemical process using transfer entropy,” *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 1, pp. 12–21, 2007.
- [25] X. Yang *et al.*, “Tri-Info: generalizable, interpretable failure prediction for VLA models via information theory,” arXiv:2606.19998, 2026.
- [26] C. Reid, W. Hafez, and A. Nazeri, “Mutual information tracks policy coherence in reinforcement learning,” arXiv:2509.10423, 2025.
- [27] G. A. Miller, “Note on the bias of information estimates,” in *Information Theory in Psychology*. Free Press, 1955, pp. 95–100.