DOI: 10.14464/gammas.v8i1.963 GAMMAS 2026, Vol. 8, No. 1, p. 23–37

International Association
of Applied Mathematics and Mechanics
– Archive for Students –

Functional Network-Based Analysis of Industrial Robot Dynamics

Khalil EL Dabett[a,⋆], Stephan Hansen[b], Charlotte Geier [a]

[a] Dynamics Group, Hamburg University of Technology, Hamburg, Germany
[b] Fraunhofer Institute for Manufacturing Technology and Advanced Materials, Stade, Germany

received 19.01.2026, accepted 27.04.2026, published 10.06.2026

⋆ corresponding author: khalil.el.dabett@gmail.com

Abstract:

Industrial robots are attractive for automating complex manufacturing tasks because they offer flexibility with a favorable ratio between assembly and workspace. However, their performance is limited by interacting sources of error resulting from serial kinematics and gear drives. These include compliance, friction, and backlash, whose combined effects are difficult to capture with conventional interaction models. This study investigates whether a network-based analysis for time series can reveal how mechanical and control variables interact during motion. Functional networks provide an unconventional approach to studying interrelationships between measurements from the different variables. Each measurement time series is represented as a node in the network, and edges are inferred from the dynamics using cross-recurrent measures. In this work, internal measurements of an axis of an industrial robot, that includes encoder positions, torques, and currents, are recorded during controlled experiments. From these signals, functional networks are constructed to describe the coupling structure between variables under varying dynamic conditions. The analysis shows that changes in the velocity and the backlash effect alter the strength and symmetry of these couplings, that reflects different dynamic behavior of the system. The results indicate that network-based representations can provide a complementary perspective for analyzing motion and understanding dynamic interactions in industrial robots.

Keywords: nonlinear time series analysis, cross recurrence, backlash, industrial robot

1 Introduction

Industrial robots are deployed across a wide range of applications, from traditional handling and assembly to complex manufacturing tasks such as welding and machining [23]. As these systems increasingly perform sophisticated operations that rely on advanced path planning and real-time adaptation, there is a growing need for refined analysis techniques to better understand and optimize their dynamic behavior. The advantage of industrial robots lies in their lower cost and smaller installation space compared to conventional Cartesian machines, which tend to be more expensive and require larger installation spaces. Industrial robots typically employ serial kinematics with multiple rotating joints, whereas Cartesian machines rely on orthogonally arranged linear axes in an X–Y–Z configuration, which yields higher rigidity and precision at the expense of increased footprint. These structural differences directly

affect the dynamic properties of the systems. However, their use in manufacturing is also associated with errors that arise from the lower rigidity of the robot structure, geometric tolerances, and the change of the moment of inertia depending on the configuration of the robot [1, 15, 22]. Additional effects such as friction and backlash occur during motion and further increase the complexity of the system. A deeper understanding of these effects and their interaction is therefore essential to ensure high accuracy and reliability in manufacturing applications [18, 22].

Traditional modeling approaches address these issues by focusing on individual components or physical effects. For example, kinematic models [20] capture geometric errors but neglect elastic effects. Compliance and stiffness models [18] describe load-dependent deformation and friction models [14] focus on motion behavior. Although each of these approaches provides valuable insight into specific mechanisms, they remain largely single-effect models. Extending these models takes to account multiple interacting effects but requires precise knowledge of physical parameters such as friction coefficients, joint stiffness values, and backlash geometry. These quantities are difficult to identify reliably and may change with operating conditions, load, or component wear. This increases the need for a complementary data-driven perspective that treats the robot as a dynamical system and to analyze interaction patterns directly from measurements, without requiring complete prior knowledge of all physical parameters.

In other scientific fields, such complex and nonlinear interactions can be analyzed using network-based representations [25]. Networks provide a way to describe dynamical systems using nodes and edges. The nodes represent variables or subsystems, and the edges represent the relationships between them. Depending on the type of network, edges between nodes can be undirected, directed, or weighted. Network representations have been successfully applied in various fields such as neuroscience to study brain synchronization [25], in physiology to reveal coupling between cardiovascular and respiratory rhythms [16], and in climate science to map dependencies among large-scale atmospheric variables [4]. These applications share the common goal of uncovering coordination structures that are not apparent when signals are analyzed in isolation.

This work employs Functional Networks (FNs) [7, 9] to study the dynamic interrelationships between different measurements of an industrial robot. In a FN, nodes correspond to a subsystem or signal variable and edges reflect dynamic relationships inferred from the recurrence structure of their time series. Rather than relying on known physical connections, FN captures patterns of influence that emerge from the system behavior. Internal signals of an industrial robot, including motor currents, joint torques, and encoder positions, are modeled as interacting variables, and network representations are derived to capture their coupling structure across different dynamic regimes. The objective is to evaluate the ability of FN to show meaningful interaction patterns in robot dynamics.

This article presents the first application of functional network analysis to the internal measurements of an industrial robot. The method itself is not new, but its transfer to the field of robotics offers a novel perspective on how control and mechanical variables interact during motion. The results show that this approach can capture coordination patterns across different motion phases and operating conditions, providing a data-driven complement to conventional modeling approaches.

The remainder of this article is organized as follows. Section 2 introduces the fundamental principles of functional network analysis and explains how recurrence-based measures can be used to infer coupling between time series. Section 3 builds on this framework by describing the experimental setup, measurement procedures, and the network construction pipeline applied to the robot data. After that, Section 4 presents the main findings for different motion conditions and highlights how coupling patterns evolve under varying dynamics. Section 5 reflects on the implications and limitations of the approach and outlines potential applications for performance monitoring and modeling of industrial robots. Finally, Section 6 summarizes the main conclusions of this work.

2 Theory

In the FN representation, each node corresponds to a system variable or subsystem, while edges describe the dynamical relationships inferred from the recurrence structure of their time series. The inference of the network edges is an iterative process in which the coupling direction between two nodes is derived from a cross-recurrence embedding of the two respective time series. This section is structured as follows. Section 2.1 introduces the fundamentals of the recurrence network. Section 2.2 presents the inter-system extension used to quantify interactions between subsystems. Building on these explanations, Section 2.3 describes the derivation of functional networks based on cross-network measures, which form the basis for the analysis conducted in this work.

2.1 Recurrence Networks

Recurrence networks form the basis of the functional network analysis. The notion of recurrence networks for time series analysis was first introduced by Donner et al. in [5]. The basic idea is the interpretation of a recurrence embedding. By interpreting each discrete measurement point as a node and each recurrence as an edge, i.e. the recurrence plot, as the adjacency matrix of a network. By interpreting each discrete measurement point as a node and each recurrence as an edge, the time series dynamics are transformed into network topology [17, 25]. For a univariate signal $x(t)$, sampled at constant frequency $f_s \approx 9.947\,\mathrm{s}^{-1}$, the phase space reconstruction yields a set of state vectors $\mathbf{x}_i \in \mathbb{R}^m$, where each $\mathbf{x}_i$ is assumed to represent the full system state at time index t_i. These vectors are constructed using an embedding dimension m and a time delay τ

$$\mathbf{x}_i = [x(t_i), x(t_i+\tau), \ldots, x(t_i+(m-1)\tau)]^\top \in \mathbb{R}^m. \quad (1)$$

such that m measurements taken at distance τ from the original time series are stacked.

Once the embedded trajectory is obtained, a binary recurrence matrix is computed by evaluating pairwise distances between these state vectors. Specifically, two points $\mathbf{x}_i$ and $\mathbf{x}_j$ are considered recurrent if their Euclidean distance is less than a predefined threshold ε, such that

$$R_{i,j}(\varepsilon) = \Theta(\varepsilon - \|\mathbf{x}_i - \mathbf{x}_j\|), \quad (2)$$

where Θ denotes the Heaviside function, and $R \in \mathbb{R}^{N\times N}$ represents the binary recurrence matrix obtained from the embedded trajectory of length N. An entry of the recurrence matrix is defined as $R_{i,j} = 1$ if the two embedded states lie within a radius ε of each other, and $R_{i,j} = 0$ [5, p. 4]. The threshold ε determines the recurrence rate (RR) of the resulting recurrence plot and hence the edge density of the corresponding network. Rather than fixing ε explicitly, a desired recurrence rate is commonly prescribed to ensure comparable network densities across different systems [9]. Finally, a recurrence network is constructed by interpreting the recurrence matrix R as the adjacency matrix A of an undirected network, where each embedded state corresponds to a node and edges are assigned wherever $R_{i,j} = 1$. Figure 1 illustrates the full conversion from a time series to its recurrence network. Starting from the sine signal in Fig. 1a, the phase space is reconstructed using $m = 3$ and $\tau = 10$ samples, yielding the trajectory shown in Fig. 1b. The corresponding recurrence matrix in Fig. 1c is obtained by marking pairs of embedded points as recurrent whenever their distance is below

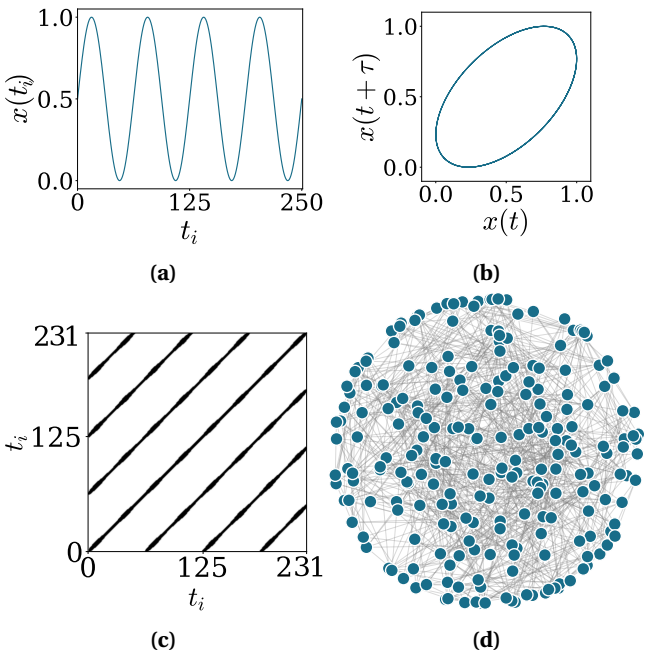


Figure 1 – Illustration of the transformation process from a sine wave to its recurrence-based network representation. (a) Sine wave, where t_i denotes discrete time sample indices; (b) embedded phase space of the two-dimensional projection ($x(t)$ vs. $x(t+\tau)$) of the reconstructed three-dimensional phase space. The full embedding utilizes an embedding dimension $m = 3$ and a delay $\tau = 10$ samples; (c) recurrence matrix; (d) Recurrence network visualization generated using NetworkX [11]. Node coordinates are calculated via a force-directed spring layout [12]. The signal is defined as $x(t) = \sin(t)$, generated over four periods for $N = 251$ samples. The recurrence matrix is computed using a fixed threshold $\varepsilon = 0.15$.

$\varepsilon = 0.15$. Interpreting this matrix as an adjacency matrix results in the recurrence network depicted in Fig. 1d.

2.2 Inter-System Recurrence Networks

Inter-system recurrence networks (ISRN) [7, 9] extend the concept of recurrence networks to the analysis of interactions between two dynamical systems X and Y. Given two time series $\{x_i\}_{i=1}^{N_X} = \{x_1, x_2, ..., x_{N_X}\}$ and $\{y_j\}_{j=1}^{N_Y} = \{y_1, y_2, ..., y_{N_X}\}$, representing the trajectories of the systems in phase space, the corresponding embedded state vectors are constructed following 1 as $\mathbf{x}_i \in \mathbb{R}^m$ for system X and $\mathbf{y}_j \in \mathbb{R}^m$ for system Y. A cross-recurrence plot [7, p. 3505] [9, p. 5] can then be defined as

$$CR_{XY,ij}(\varepsilon_{XY}) = \Theta(\varepsilon_{XY} - \|\mathbf{x}_i - \mathbf{y}_j\|), \quad (3)$$

where $\|\mathbf{x}_i - \mathbf{y}_j\|$ denotes the Euclidean distance between an embedded state of system X and an embedded state of system Y, $\Theta(\cdot)$ is the Heaviside function, and ε_{XY} is the distance threshold. Because system X consists of M_X embedded states and system Y consists of M_Y embedded states, the resulting cross-recurrence matrix has dimensions $CR_{XY} \in \mathbb{R}^{M_X \times M_Y}$. The ISRN adjacency matrix is defined as

$$\boldsymbol{A}^{\mathrm{ISRN}}(\boldsymbol{\varepsilon}) = \begin{bmatrix} \boldsymbol{A}_X(\varepsilon_X) & \boldsymbol{CR}_{XY}(\varepsilon_{XY}) \\ \boldsymbol{CR}_{YX}(\varepsilon_{XY}) & \boldsymbol{A}_Y(\varepsilon_Y) \end{bmatrix} \tag{4}$$

where the diagonal blocks $\mathbf{A_X} \in \mathbb{R}^{M_X \times M_X}$ and $A_Y \in \mathbb{R}^{M_Y \times M_Y}$ represent the recurrence adjacency matrices of systems X and Y, respectively [25, p. 39] [9, p. 6]. The off-diagonal blocks $\boldsymbol{CR}_{XY} \in \mathbb{R}^{M_X \times M_Y}$ and $\boldsymbol{CR}_{YX} = \boldsymbol{CR}_{XY}^{\top} \in \mathbb{R}^{M_Y \times M_X}$ encode cross-recurrences between the two systems, resulting in a complete ISRN adjacency matrix $\boldsymbol{A}^{\mathrm{ISRN}} \in \mathbb{R}^{(M_X+M_Y)\times(M_X+M_Y)}$. Each entry of A_X is obtained from

$$A_{X,ij}(\varepsilon_X) = \Theta(\varepsilon_X - \|\mathbf{x}_i - \mathbf{x}_j\|) - \delta_{ij}, \tag{5}$$

where $\|\mathbf{x}_i - \mathbf{x}_j\|$ denotes the Euclidean distance between two embedded state vectors of system X, $\Theta(\cdot)$ the Heaviside function, and δ_{ij} the Kronecker delta used to remove self-loops. A similar expression holds for A_Y with state vectors $\mathbf{y}_i, \mathbf{y}_j$ of system Y. The thresholds ε_X and ε_Y are selected such that a fixed recurrence rate is achieved within each subsystem.

The off-diagonal blocks CR_{XY} and $CR_{YX} = CR_{XY}^{\top}$ encode cross-recurrences between the two systems. Their entries depend on the selected threshold ε_{XY}, which controls the density of edges between the two subsystems. Each nonzero entry $CR_{XY,ij} = 1$ indicates that the state x_i of system X and the state y_j of system Y lie within a distance ε_{XY}, and are therefore close in the combined phase space. The inter-system threshold ε_{XY} should be chosen smaller than the intra-system thresholds ε_Y and ε_X to obtain a modular network for the two sub-network [7]. Figure 2 summarizes the construction of the ISRN. Signal X is defined as a simple sinusoidal oscillator, $x(t) = \sin(t)$, while signal Y combined an oscillation at a different frequency with an additive coupling term, $y(t) = \sin(1.5t) + \alpha x(t)$, where $\alpha = 0.3$ controls the coupling strength. Both signal are shown in Fig. 2a. Their delay-coordinate embeddings with embedding dimension $m = 3$ and delay $\tau = 10$ samples are displayed in Fig. 2b. Based on these trajectories, the individual recurrence matrices A_X and A_Y as well as the cross-recurrence matrix CR_{XY} are computed, as illustrated in Fig. 2c. Combining these blocks yields the ISRN adjacency matrix defined in Eq. (4), whose corresponding graph representation is shown in Fig. 2d.

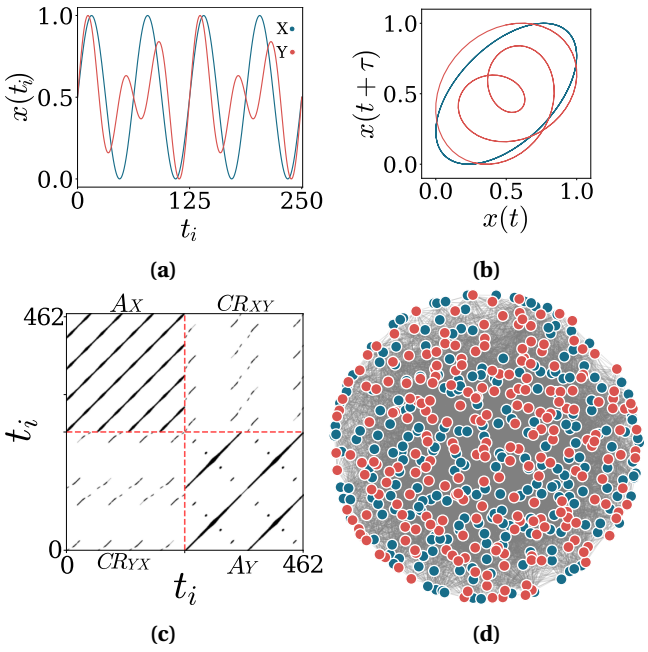


Figure 2 – Illustration of the construction of an Inter-System Recurrence Network (ISRN) from two time series. (a) Time series of the driving sinusoidal signal $x(t) = \sin(t)$ (blue) and the coupled signal $y(t) = \sin(1.5t) + 0.3x(t)$ (red), each generated over four periods with $N = 251$ samples. (b) A 2-dimensional projection ($x(t)$ vs. $x(t+\tau)$) of the reconstructed 3-dimensional phase space for both systems, using an embedding dimension $m = 3$ and a discrete index delay of $\tau = 10$ samples. (c) Individual recurrence matrices A_X and A_Y computed for each system with thresholds $\varepsilon_X = \varepsilon_Y = 0.15$, together with the cross-recurrence matrix CR_{XY} computed using $\varepsilon_{XY} = 0.05$. (d) Resulting ISRN visualization generated using the NetworkX [11] via a force-directed spring layout [12]. Blue nodes correspond to system X and red nodes to system Y.

The ISRN is a network consisting of two sub-networks, one for every subsystem. The resulting ISRN structure forms the foundation for quantifying directional coupling through network-based measures such as cross-clustering and cross-transitivity, which are then used to construct the functional networks introduced in the following sub-section.

2.3 Functional Networks

To obtain a more compact system-level description, the ISRN can be summarized into a Functional Network (FN), where each variable or time signal is represented as a node. Directed and weighted edges are then established between nodes based on asymmetries in the ISRN topology, using the measures of cross-clustering

and cross-transitivity coefficients [7, p. 3506] [9, p. 6].

Cross-clustering coefficient. For a node $v \in V_X$ corresponding to measurements from time series X, the local cross-clustering coefficient describes the probability that two of its neighbors in the other subsystem V_Y are also connected to each other within V_Y. For example, let $p, q \in V_Y$ be two nodes that are both connected to v through the ISRN adjacency matrix $\mathbf{A}^{\mathrm{ISRN}}$. If p and q are themselves connected in V_Y, they form a "cross-triangle" together with v. The local cross-clustering coefficient is then defined as

$$C^{XY}(v) = \frac{\sum_{p,q \in V_Y} A^{\mathrm{ISRN}}_{vp} A^{\mathrm{ISRN}}_{pq} A^{\mathrm{ISRN}}_{qv}}{k_v^{XY}(k_v^{XY}-1)}, \tag{6}$$

where the *cross-degree* of node v is given by

$$k_v^{XY} = \sum_{q \in V_Y} A^{\mathrm{ISRN}}_{vq}. \tag{7}$$

The global cross-clustering coefficient is obtained as the average over all nodes in V_X and is defined as

$$C^{XY} = \frac{1}{|V_X|} \sum_{v \in V_X} C^{XY}(v), \tag{8}$$

where $|V_x|$ denotes the number of nodes in the node set V_x. As illustrated in Fig. 3, the local cross-clustering coefficient $C^{XY}(v)$ quantifies how often neighbors in subsystem V_Y that share a connection to a node $v \in V_X$ are also mutually connected.

Cross-transitivity. The cross-transitivity T_{XY} evaluates the probability that two nodes $p, q \in V_Y$, both connected to the same node $v \in V_X$, are also connected to each other within V_Y. In other words, it measures how often a "cross-triple" (v, p, q)—where v is connected to both p and q—is closed into a "cross-triangle" by an additional edge between p and q. Formally, this is defined as

$$T^{XY} = \frac{\sum_{v \in V_X} \sum_{p \neq q \in V_Y} A^{\mathrm{ISRN}}_{vp} A^{\mathrm{ISRN}}_{vq} A^{\mathrm{ISRN}}_{pq}}{\sum_{v \in V_X} \sum_{p \neq q \in V_Y} A^{\mathrm{ISRN}}_{vp} A^{\mathrm{ISRN}}_{vq}}. \tag{9}$$

Here, the numerator counts the number of cross-triangles, while the denominator counts all cross-triples. The ratio therefore gives the probability that these triples are transitively closed across the two subsystems. The concept of cross-transitivity T^{XY} is shown in Fig. 4, where connected triples in subsystem Y formed through nodes in X are used to evaluate the closure of cross-triangles.

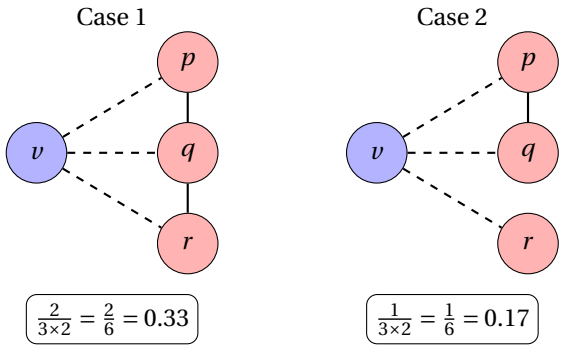


Figure 3 – Illustration of the computation of the cross-clustering coefficient $C^{XY}(v)$ for a node $v \in V_X$ using Eq. (6). A node v is connected to three neighbours $p, q, r \in V_Y$, forming $k_v^{XY}(k_v^{XY}-1) = 3 \times 2 = 6$ possible neighbour pairs. Cross-triangles are counted when a pair of neighbours in V_Y is also connected (e.g., p–q or q–r). The local coefficient is computed as shown in the figure and the global C^{XY} follows by averaging over all nodes in V_X.

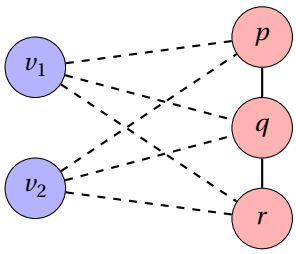
2.4 Directional Coupling and Asymmetry

To determine the dominant direction of interaction, the difference between the two global cross-clustering and cross-transitivity values is taken. Since the global values average the clustering and transitivity values across all nodes, the two differences can be computed as

$$C_{\mathrm{diff}} = C^{XY} - C^{YX}, \quad T_{\mathrm{diff}} = T^{XY} - T^{YX}. \tag{10}$$

Positive values of C_{diff} or T_{diff} indicate that the trajectory of subsystem Y is more strongly constrained by that of X, implying a directional influence $X \to Y$. Negative values suggest the opposite direction. This interpretation follows the logic of inter-system recurrence networks, where a unidirectional coupling $X \to Y$ causes the trajectory of Y to be partially "dragged" into the phase-space structure of X [7, 9]. As a result, cross-triangles comprising two nodes in V_X and one node in V_Y (e.g., $v_1 - v_2 - p$, with $v_1, v_2 \in V_X$ and $p \in V_Y$) occur more frequently than those comprising two nodes in V_Y and one node in V_X (e.g., $p - q - v_1$, with $p, q \in V_Y$ and $v \in V_X$), leading to $T^{YX} > T^{XY}$ and hence a negative T_{diff}.

If the two systems are not coupled, the cross-transitivi ty and the cross-recurrence could be interpreted as resulting from random coincidences, such that their differences lie close to zero $C_{\mathrm{diff}} \approx 0$ and $T_{\mathrm{diff}} \approx 0$. Likewise, in the case of perfectly symmetric bidirectional coupling, both directions are equally constrained and again the asymmetry vanishes. Therefore, the method cannot distinguish between these two situations (no coupling vs. symmetric bidirectional influence) [7, 9]. For this



$$T^{XY} = \frac{\text{cross-triangles}}{\text{cross-triples}} = \frac{4}{6} = 0.67$$

Figure 4 – Illustration of cross-transitivity T^{XY}. Two nodes v_1 and v_2 in subsystem V_X each connect to the same three neighbours p, q, r in subsystem V_Y, generating all possible cross-triples (v, p, q), (v, p, r), (v, q, r). Cross-triangles occur when the corresponding neighbour pair in V_Y is connected (e.g., p–q or q–r). Cross-transitivity is computed for the whole network.

reason, a small threshold μ is introduced. Relations satisfying

$$-\mu < C_{\text{diff}} < \mu, \quad -\mu < T_{\text{diff}} < \mu$$

are treated as bidirectional (or absent), whereas values above or below the threshold indicate a dominant unidirectional dependency.

The functional network obtained from the ISRN in Fig. 2 by applying the C_{diff} measure is shown in Fig. 5, illustrating how directional coordination between the two systems emerges after transforming cross-recurrence structure into asymmetric coupling relations.

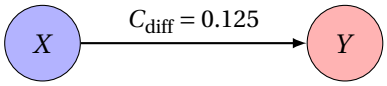


Figure 5 – Functional network derived from the ISRN using the cross-clustering difference measure C_{diff}.

After iterative computation over all pairs of measurement data, the resulting functional network provides a directed representation of interdependencies among multiple time series. Each node represents a signal or subsystem, and the edges describe the inferred coupling structure and direction influence. Unlike model-based approaches, which rely on predefined equations of motion or controller dynamics, this method derives relationships directly from measured data. This offers a new system perspective on the robots dynamic behavior.

3 Methodology

The following section presents the experimental setup, the design of the motion trials, and the data acquisition and preprocessing procedures used to construct functional networks from the robot measurements. The workflow follows the pipeline shown in Fig. 6. The process begins with the definition of the experimental setup, followed by the acquisition of data from the robot controller at a sampling rate of 500 Hz. Subsequently, the recorded signals are preprocessed, where velocity is derived and Min–Max normalization is applied to the original time series. Finally, FNs are constructed based on the normalized time series and visualized as directed and weighted graphs. This will form the foundation for the analyses presented in Section 4.

3.1 Experimental Setup

The experimental investigations are carried out on a self-built industrial robot developed within the Fraunhofer project Flexmatik 4.1 [21]. The system is designed for path accurate processes, e.g. milling of large and complex components made from carbon fiber-reinforced polymers (CFRP). It features a five-axis serial kinematic architecture with a rotary swivel head and integrated spindle.

Figure 7 depicts the self-build robot in its operational configuration. The system is controlled by a Siemens SINUMERIK 840D sl CNC controller, providing path planning, synchronized data logging, and G-code compatibility for serial robot kinematics [19]. To improve positioning accuracy and reduce the effect of transmission backlash, secondary encoders are installed. The experimental plan follows a structured Design of Experiments (DOE) in which three parameters are systematically varied: the Tool Center Point (TCP) configuration (1.2 m, 1.6 m, and 2 m radial distance), the angular step size (1°, 3°, and 10°), and the joint speed (35°/min, 175°/min, and 350°/min).

The purpose of these main trials is to study how changes in the effective moment of inertia and different velocities influence the relationships between the internal variables of the robot. For each parameter combination, the first base axis (A1) performs a single forward motion from the initial position to the commanded angular step, resulting in a total of $3 \times 3 \times 3 = 27$ primary experiments. Figure 8 illustrates the robot and the three TCP configurations considered in the experiments. The TCP is positioned at radial distances of 1200 mm, 1600 mm, and 2000 mm from the base axis to represent different load and inertia conditions during motion.

In addition to the main experiments, two further test

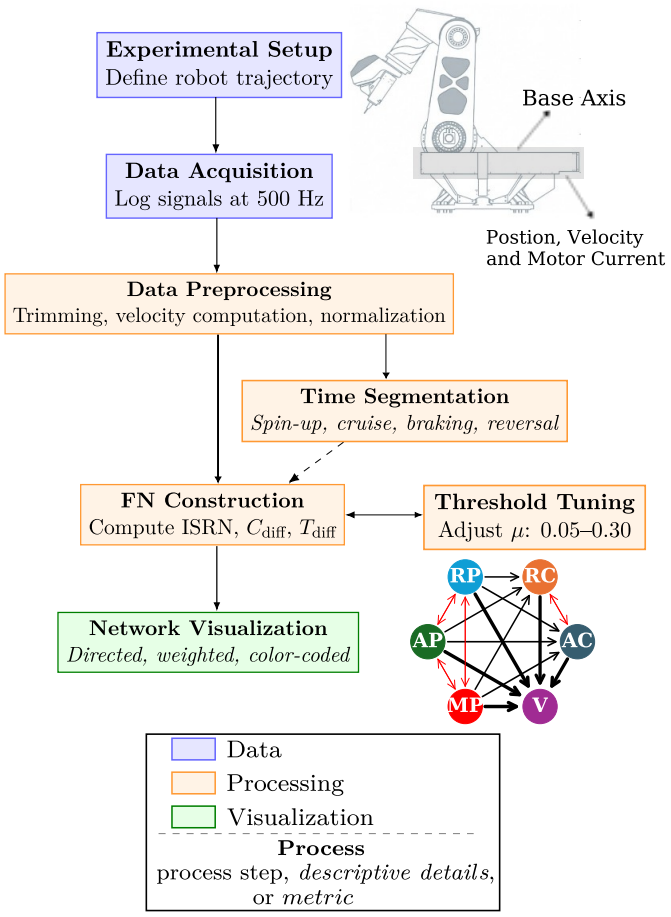


Figure 6 – Overview of the FN analysis pipeline. The process begins with the experimental setup, followed by data acquisition from the robot controller at 500 Hz. The recorded signals include the reference joint position (RP), arm-side position (AP), motor-side encoder position (MP), reference motor current (RC), and actual motor current (AC). The raw time series are then preprocessed through data trimming, velocity computation, and normalization. Next, the signals are segmented into motion phases such as spin-up, cruise, braking, and reversal. Based on these prepared data, functional networks are constructed using recurrence-based measures. Finally, the resulting networks are visualized as directed and weighted graphs.

sets are conducted to isolate physical effects that are difficult to assess under normal operating speeds. First, static friction is examined using slow motions with a 1° step size at a very low speed of 1°/min for each TCP configuration. Second, backlash characteristics are investigated with forward (F) and reverse (R) motion ($\pm 1^\circ$ steps) executed at 35°/min and 175°/min across all configurations. A complete summary of all experimental groups is given in Table 1.

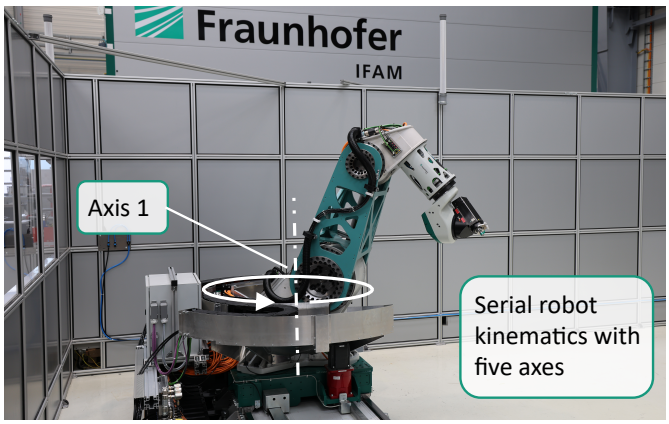


Figure 7 – Self-build robot for precision applications. The system comprises serial kinematics with five axes mounted on a linear axis.

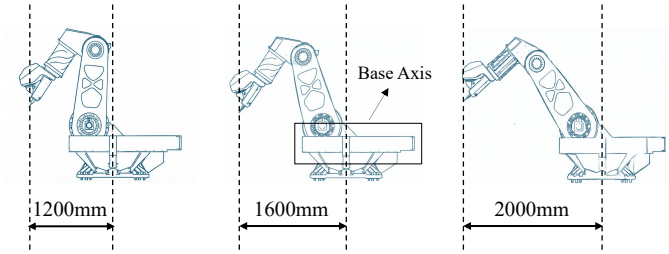


Figure 8 – Experimental setup and TCP configurations. The Flexmatik robot (left) and the three TCP distances of 1200 mm, 1600 mm, and 2000 mm used to vary system inertia (right).

3.2 Data Acquisition

All experimental measurements are recorded at a sampling rate of 500 Hz using the Siemens SINUMERIK 840D sl CNC controller, which provides synchronized logging of internal parameters throughout the motion. For each experiment, the controller records the commanded joint trajectory together with the robot's mechanical response. The dataset consists of the reference joint position (RP), arm-side position (AP), motor-side encoder position (MP), reference motor current (RC), and actual motor current (AC). These signals capture the control behavior and the physical dynamics of the industrial robot. An example of the recorded signals for a single motion sequence is shown in Fig. 9, illustrating the characteristic evolution of position and current during acceleration, steady-state, and braking motion.

3.3 Data Preprocessing

For the data preprocessing, All XML log files are first cleaned by removing descriptive metadata and retaining only the numerical fields corresponding to the recorded signals. After import, the arm-side encoder position is used to compute the joint velocity numerically. A

Table 1 – Design of Experiments

Group	Speed [°/min]	TCP [m]	Step Size [°]	Dir.
Main	35,175,350	1.2,1.6,2	1,3,10	F
Static fric.	1	1.2,1.6,2	1	F
Backlash	35,175	1.2,1.6,2	±1	F+R

central difference approximation is applied to estimate the derivative

$$v(t_i) \approx \frac{x(t_{i+1}) - x(t_{i-1})}{2\Delta t}, \tag{11}$$

where the sampling interval is $\Delta t = 1/500\,\mathrm{s}$. The raw derivative is sensitive to noise, particularly during rapid transients. To suppress these fluctuations while preserving the underlying motion profile, the velocity signal is smoothed using a three-point moving average

$$\bar{v}(t_i) = \frac{1}{3}\left[v(t_{i-1}) + v(t_i) + v(t_{i+1})\right]. \tag{12}$$

Because the recorded variables operate on different physical scales, all signals are subsequently normalized to the interval $[0,1]$ using Min–max scaling [2]. This step ensures that no single variable dominates the distance computations in the recurrence-based embedding and provides consistent weighting across all dimensions of the reconstructed phase space [25].

These processed signals form the basis for the functional network construction in Section 4, where recurrence based measures are applied to extract coupling structure and directional dependencies between the recorded robot variables.

3.4 Functional Network Construction

To investigate coordination patterns and directional dependencies between internal signals in the robot, FN are constructed using the Inter-System Recurrence Network (ISRN) framework, as described in Section 2. The recurrence thresholds ε_X and ε_Y are chosen such that a fixed recurrence rate of $RR = 0.05$ is achieved for the individual systems. The inter-system threshold ε_{XY} is selected to obtain a fixed cross-recurrence rate $CR = 0.05$. For the construction of directed functional networks, a bidirectionality threshold $\mu = 0.05$ is applied to both C_{diff} and T_{diff}.

The functional network algorithm is implemented using the Python codebase [8] provided alongside the methodology introduced in [9] which makes use of the pyunicorn library [3].

To adapt the code for the current experimental context, several modifications are introduced. First, a weighted representation of the network is implemented by assigning edge weights proportional to the average directional influence between nodes.

$$w_{xy} = \frac{1}{2}\left(|C_{\mathrm{diff}}(x,y)| + |T_{\mathrm{diff}}(x,y)|\right). \tag{13}$$

Here, w_{xy} denotes the weight of the directed edge from node x to node y, defined as the mean of the absolute asymmetry measures in coupling strength C_{diff} and T_{diff}.

The resulting networks are visualized using a circular layout [10], with each node representing one measured variable (e.g., current, position, or velocity). Directed edges indicate functional influence, with edge thickness scaled according to the computed weight w_{ij} to reflect interaction strength based on average of C_{diff} and T_{diff}. Unidirectional edges are shown in black, while bidirectional connections are highlighted in red. An example of a FN is illustrated in Fig. 6.

4 Results

In this section, the results of the FN analysis are presented, as applied to experimental data collected from the first axis of an industrial robot under various dynamic conditions. Building upon the theoretical and methodological framework established in Sections 2 and 3, this analysis aims to uncover dynamical dependencies between control and mechanical signals while investigating how these relationships evolve across motion phases and experimental configurations.

This section is structured as follows. Section 4.1 introduces a baseline experiment at moderate velocity and standard configuration, Section 4.2 analyzes the influence of velocity across four operating speeds from quasi-static to high-speed motion. Finally, Section 4.3 highlights the impact of mechanical backlash on system behavior.

4.1 Baseline Experiment

The baseline experiment, characterized by a joint velocity of 35°/min , a step size of 3° and with a TCP distance of 1600 mm, serves as a baseline case for analyzing robot motion. This configuration corresponds to a moderate velocity and a central workspace position, making it suitable for observing typical interactions between control and mechanical signals. Figures 9 shows the raw and normalized time series data for six key parameters measured during the baseline experiment: RP, AP, MP, V, RC, and AC. Below the time series the FNs extracted for four distinct motion phases: early spin-up (0–220 milli-seconds (ms)), late spin-up (220–1000 ms), cruise (1000–5000 ms), and braking (5000–5700 ms) are

shown. Through these networks, the evolution of variable dependencies is characterized as the robot progresses through different motion phases.

In each FN, directed edges are used to indicate the dominant direction of influence, as inferred from pairwise inter-system recurrence structures, while the thickness of each edge reflects the strength of the corresponding interaction. Bidirectional edges can appear in two situations. Either, when mutual influence exists between joints (bidirectional coupling) or when no meaningful relationship is present (no interrelation), as discussed in Section 2 and originally described by [9]. Following the approach of [9], a small threshold ($\mu = 0.05$) is initially applied to detect symmetric relationships.

During the initial spin-up phase shown in Fig. 9, the robot accelerates from standstill and must overcome static friction and the initial moment of inertia. The FN shows strong outward edges from RP to AP and MP, indicating that the motion is initiated by the reference position. A bidirectional edge between AP and V reflects the rapid increase of velocity relative to arm movement, without a clear coupling. RC and AC form a coupled pair, that is consistent with the faster current control loop.

Outgoing edges from RC and AC to all other variables indicate that the current control circuit, as the innermost and fastest cascade, is already active at the onset of motion. In the static friction region, the controller generates torque without producing immediate movement. Once the static friction is overcome, additional torque is applied to accelerate the arm.

In the late spin-up phase, after overcoming the static friction, the acceleration becomes more steady. Bidirectional edges emerge between the position variables (RP, AP, and MP), reflecting constant position lag between the actual and reference position. Outgoing edges from the position parameters toward V, RC, and AC indicate that the position dynamics govern both velocity development and torque demand. The velocity and current signals exhibit similar oscillatory patterns. Compared to the earlier phase, the current provides the torque required to sustain acceleration against inertia.

During the cruise phase, shown in Fig. 9, the position cluster remains intact, indicating sustained synchronization between the reference input and the mechanical output. The time series demonstrate that these three signals evolve almost identically, suggesting constant position lag during steady motion. Directed edges from the position cluster to V, RC, and AC persist and appear stronger, reflecting the system ability to maintain velocity through continuous torque regulation. Velocity and current signals continue to exhibit similar oscillatory structures, with changes in current slightly preceding fluctuations in velocity.

In the braking phase, shown in Fig. 9, the FN reorganizes as the system transitions from steady motion to deceleration. RC and AC now show outward edges toward all other motion variables, reversing the earlier direction. This reflects the transition from position-guided motion to controlled deceleration, where the velocity control loop provides damping and current control supplies the necessary torque. AP projects outward to both MP and RP, reflecting feedback relations as the system reduces motion. RP appears to play a diminished role during braking. This might be due to a decrease in error and the controller shifting emphasis toward controlled deceleration rather than trajectory execution.

Across the full motion cycle, the FN captures a coherent progression in system dynamics. Early spin-up is dominated by RP-initiated motion and current required to overcome resistance. In late spin-up and cruise, the position cluster drives velocity and current behavior under steady tracking. During braking, current control becomes dominant as the robot departs from equilibrium and transitions to torque-based deceleration. These baseline results establish a reference for interpreting variations introduced by changes in velocity and reversal motion, which are examined in the following sections.

4.2 Influence of Velocity Variation

To examine how functional coordination changes with motion dynamics, FNs are constructed for four velocities: 1°/min (V1), 35°/min (V35), 175°/min (V175), and 350°/min (V350). These velocities range from quasi-static motion to conditions representative of high-speed industrial operation and correspond approximately to feed rates from 1–10 m/min at an arm length of 1 m. All experiments use a fixed robot TCP position of 1600 mm (P16) and follows the steps of 3° on the programmed path. Only experiment V350 uses larger steps of 10° because of its shorter cruise phase. For all cases, the signals are normalized within the selected time windows to eliminate scale differences between variables and to ensure consistent network construction across experiments. Each motion phase, spin-up, cruise, and braking, is analyzed using fixed time windows (280, 600, and 600 ms, respectively) to ensure comparability between speeds.

The FNs generated from Min-max normalized signals for all velocities and motion phases are shown in Figure 10. In each cell, the normalized time series is presented above the corresponding FN, allowing changes in network topology to be interpreted alongside variations in signal dynamics. The analysis focuses on identifying

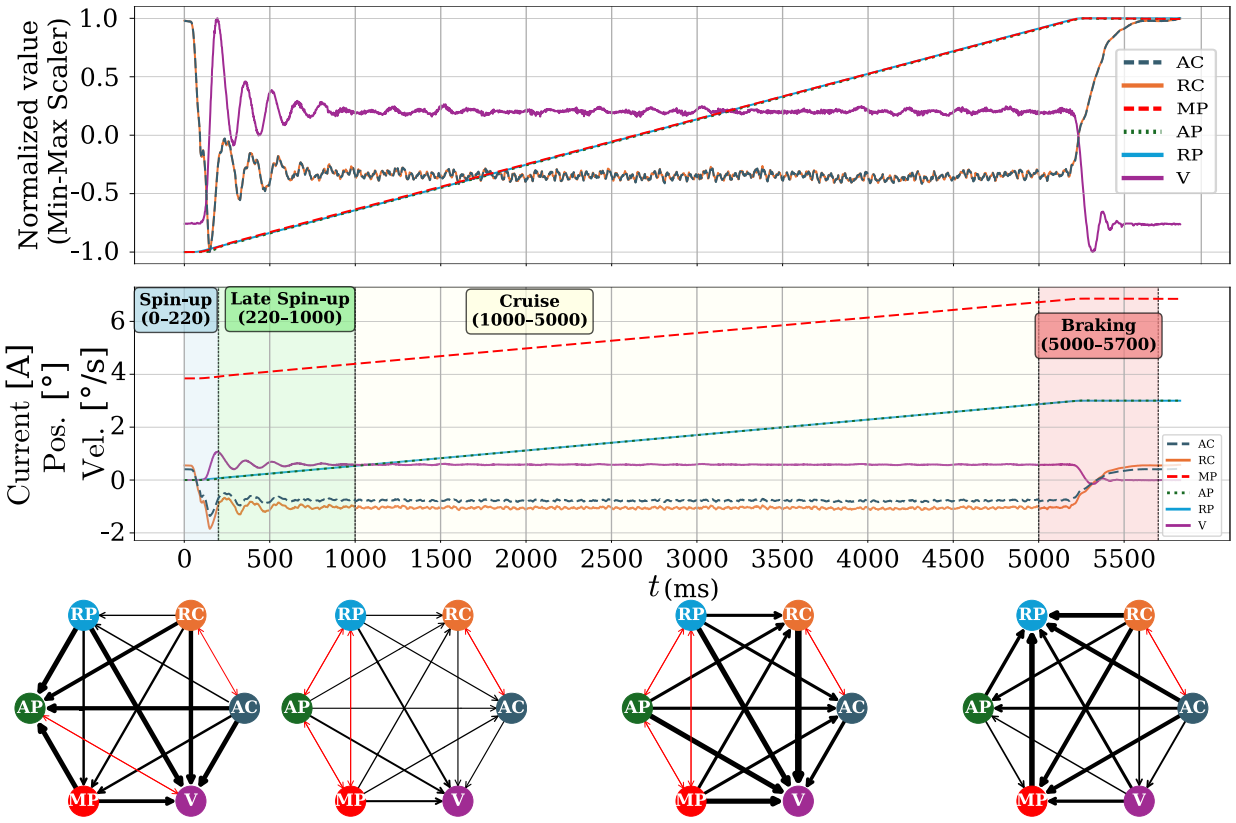


Figure 9 – Time series of the baseline experiment. (a) Corresponding Min–Max normalized signals, used for the construction of functional networks. (b) Raw signals, with motion phases separated into early spin-up, late spin-up, cruise, and braking. Below the two signals, FNs are constructed for each motion phase of the baseline experiment (V35). Spin-Up Phase (0–220 ms), Late Spin-Up (220–1000 ms), Cruise Phase (1000–5000 ms), and Braking Phase (5000–5700 ms). Directed edges represent inferred influence between variables; edge thickness encodes the strength of coupling. For presentation purposes, the speed is shown in °/s instead of °/min.

shifts in recurrent coordination patterns that occur with changes in velocity.

When compared to the baseline case at V35, both V175 and V350 exhibit slightly altered FN structures in the spin-up phase. At these higher velocities, the bidirectional edge between AP and V observed at V35 disappears. Instead, AP and V show alternating edge orientations between outward and inward connections. During the cruise phase, the FNs at V175 and V350 largely resemble the baseline structure, with (RP–AP–MP) clustering and consistent velocity regulation. However, edge thickness decreases as velocity increases, and the direction of edges shifts from velocity toward the current cluster, in contrast to the patterns seen at lower speeds. In the braking phase, the overall structure remains consistent to V35, with the current signals maintaining strong mutual coupling and V projecting weaker edges toward position variables. Overall, there is a clear trend across all motion phases: as velocity increases, the connections between parameters generally become thinner.

Low speed motion (V1) also produces a distinct structural pattern. During the spin-up phase, the FN for V1 and V35 are much the same, except for a shift in the direction of connections between RP and the current signals. In cruise, two relatively independent clusters are observed. The first consists of the position variables (RP-AP-MP), and the other consists of the velocity and current variables (V–RC–AC). During braking, most position variables have a bidirectional edge with velocity, but there is an outward connection from MP to AP.

Certain FN features remain invariant across different velocities. The bidirectional edge between RC and AC is preserved in all conditions, which may be the effect of the high sampling rate of the current control loop. Likewise, braking-phase networks consistently exhibit outward influence from RC and AC to the motion variables, reflecting the torque-dominated nature of deceleration control. The position cluster also remains well-defined during cruise across all trials, indicating persistent kinematic coordination in this phase. These consistent features suggest that FNs are capable of extracting speed-independent coordination structures.

Overall, the results demonstrate that FN structures vary systematically with velocity while retaining certain

invariant features. At higher velocities, edges generally become thinner, and influence shifts from velocity toward the current signals. Whereas at lower velocities, the position and current–velocity loops tend to form more distinct clusters. Despite these variations, stable patterns such as the RC–AC coupling and the cruise phase position cluster are preserved across all trials. These observations indicate that velocity primarily modulates the relative strength and organization of connections, without altering the general parameter coordination present in all motion phases.

4.3 Influence of Backlash

To investigate the effect of backlash on parameter coordination, FNs are constructed within the reversal region and compared with the established spin-up, cruise, and braking phases. The results are shown in Figure 11, where the first graph presents the complete motion, including spin-up, cruise, reversal, and braking, and the second graph provides a zoomed in view of the reversal interval between 1700 and 2000 ms. In addition, sliding windows shifted by ±20 ms and ±40 ms are included to illustrate how the FN reorganizes at different phases of the reversal motion. The FN obtained during the reversal exhibits patterns distinct from the spin-up and braking dynamic observed previously. In this phase, velocity emerges as the dominant driver, projecting outward connections to all other parameters. This stands in contrast to spin-up, where the current signals (RC and AC) consistently dominate, and to braking, where the current variables project outward toward the motion signals. The position variables also exhibit a distinctive reorganization: MP projects outward to both AP and RP, while AP projects outward to RP. This contrasts with the tightly synchronized position clusters observed during steady cruise and the spin-up configuration, where position variables primarily follow the reference input.

The sliding-window analysis further highlights how this reversal signature develops over time. At $t = -40$ ms, the network resembles the cruise phase structure but with some differences. RC and AC project outward edges to both velocity and the position cluster. At $t = -20$ ms, bidirectional edges appear between velocity and the current signals, indicating a transitional stage where motor and velocity jointly influence each other. At 0 ms the configuration described before is shown. At $t = +20$ ms, this configuration remains largely present, while at $t = +40$ ms the network begins to return toward the cruise motion phases. AP and RP cluster together, MP projects toward both AP and RP, and velocity reconnects with RC and AC.

These observations indicate that backlash produces a distinct FN signature that develops gradually as the motion passes through reversal, with velocity dominance and position reorganization at its center and transitional patterns on either side.

5 Discussion

The results presented in Section 4 show how FNs can be constructed from robot motion data and how their structure varies across motion phases, velocities, poses, and step sizes. In this chapter, these findings are discussed in more detail. The purpose is to reflect on what they reveal about robot behavior and to assess the usefulness of FNs as a method. In the following section, the main findings are discussed together with methodological limitations and potential applications of the approach.

5.1 Analysis of Results

Phase-dependent coordination. Across all experiments, the FNs reveal characteristic and recurring phase specific structures. During the first spin-up phase of each experiment, a new RP and corresponding RC are specified to the system, driving the subsequent movement of the robot arm. This procedure is captured by the FN through influence patterns from RP and RC toward the remaining variables, as expected from the cascade control structure. An additional hierarchy is evident in the tight grouping of the two current variables (AC/RC), reflecting their faster dynamics typical of current control loops operating at 1–2 kHz bandwidth [13], which exceed our 500 Hz sampling resolution.

As the system enters the cruise phase, bidirectional position interactions emerge with stable lag between RP, AP, and MP confirming effective trajectory tracking with minimal error. Position dominance over velocity and currents aligns with the control hierarchy of position → velocity → torque. During braking, the currents shift to influence all other variables, consistent with torque-based deceleration, where the inner loops damp the motion independently of the reference trajectory. These phase transitions demonstrate that FNs reliably reflect the known control structure while revealing dynamic propagation across motion phases.

Influence of velocity. The results show that the FNs reveal distinct shifts in the interactions of the parameters at different velocities and motion phases. A general trend observed across the motion phases is that edges become thinner at higher speeds. Thinner edges may indicate higher kinetic energy in the system, which can

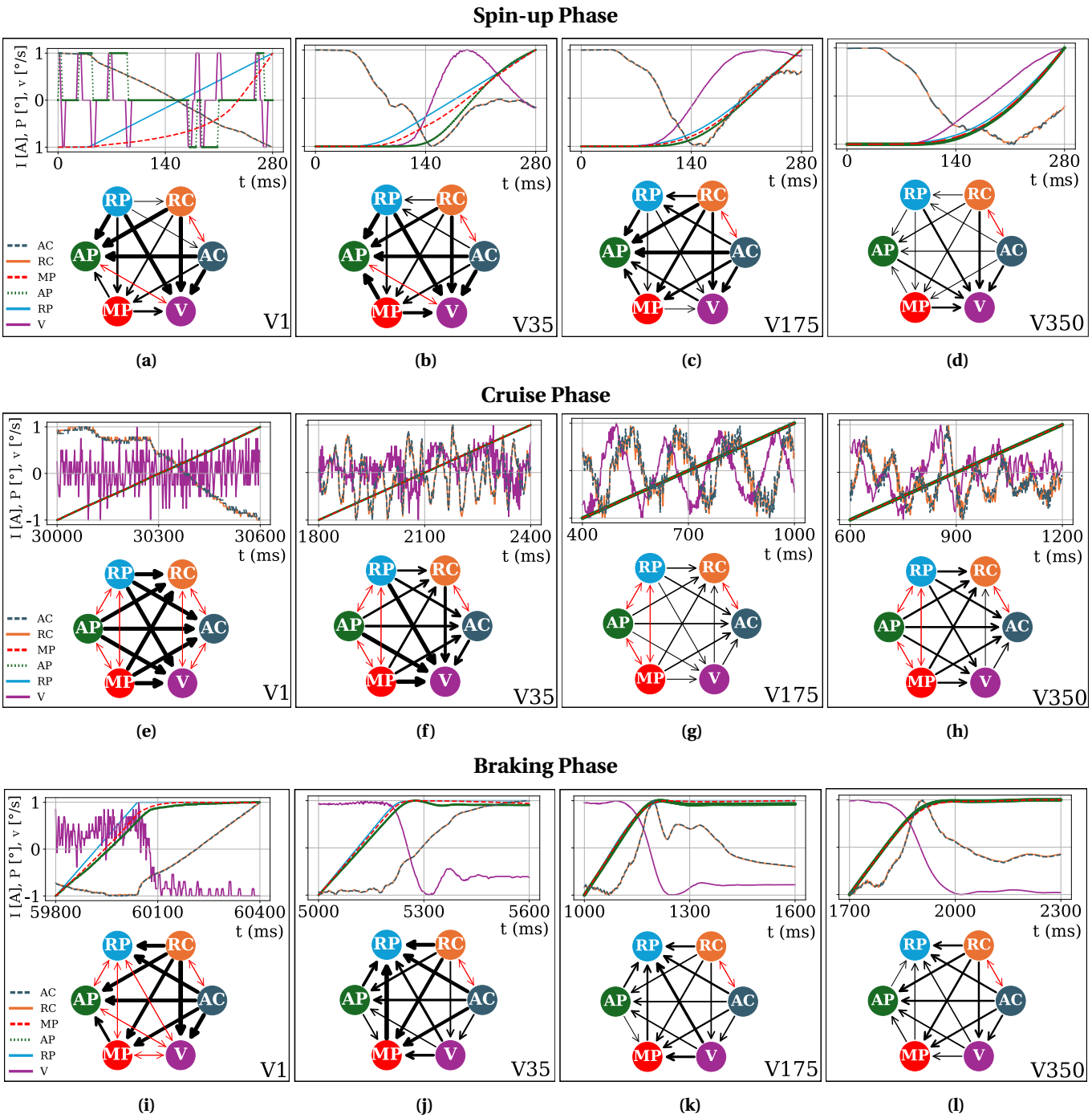


Figure 10 – Functional networks across four velocities (1°/min (V1), 35°/min (V35), 175°/min (V175), 350°/min (V350)) shown for three motion phases: Spin-up (top), Cruise (middle), and Braking (bottom). Each subfigure displays the time-series segment and its corresponding FN. The signals consist of the reference joint position (RP), arm-side position (AP), motor-side encoder position (MP), reference motor current (RC), and actual motor current (AC).

reduce tracking errors between parameters and lead to more independent or less tightly coupled trajectories.

In addition to the general trend, there are some shifts in some edges between parameters at different motion phases. During spin-up, the change of influence between AP and V with increasing velocity can be understood as a reflection of how different physical effects shape the robot's motion. At lower velocity (V35), the friction effect is more prominent. Motion develops slowly, velocity remains small and irregular, and both signals respond to the same small disturbances. In this regime, a bidirectional edge emerges between AP and V. At intermediate velocity (V175), acceleration is higher and the movement of the load becomes smoother. Un-

der these conditions, velocity develops into a clearer and more repeatable profile, while AP follows with a constant lag. The FN therefore show velocity exerting the stronger influence. At the highest velocity (V350), additional mechanical effects become visible. Elastic compliance in the drivetrain produces small torsional deflections that accumulate when torque demand increases. These compliance variations are expressed more strongly in AP.

As a result, the FN shows a reversal, with AP influencing V. This change is valuable because it shows how the FN approach can capture not only steady couplings but also changes in the dominant physical effect.

A similar transition is observed in the cruise phase. At low velocities, the current signals influence the velocity. This reflects that small variations in current are sufficient to sustain motion, while velocity adapts to these inputs. As the commanded speed increases, the relative importance of the reference trajectory becomes stronger, and the velocity develops into a more regular profile that guides the interaction. In contrast, the current signals fluctuate more as they adjust to the torque demands of faster motion. In this phase, the FN show velocity exerting the stronger influence, since the stable velocity trajectory organizes the recurrence structure of the more variable current signals.

Taken together, these findings show that FNs shows reorganizations in parameter interactions across different velocities. The observed edge thinning, along with the directional reversals between velocity, position, and current, indicates that the FN structure is changing as motion velocity changes. These insights highlight how the FN structure changes at different motion phases and velocities.

Reversal Motion The experiments with reversal motion show that the FN network structure is different dur-

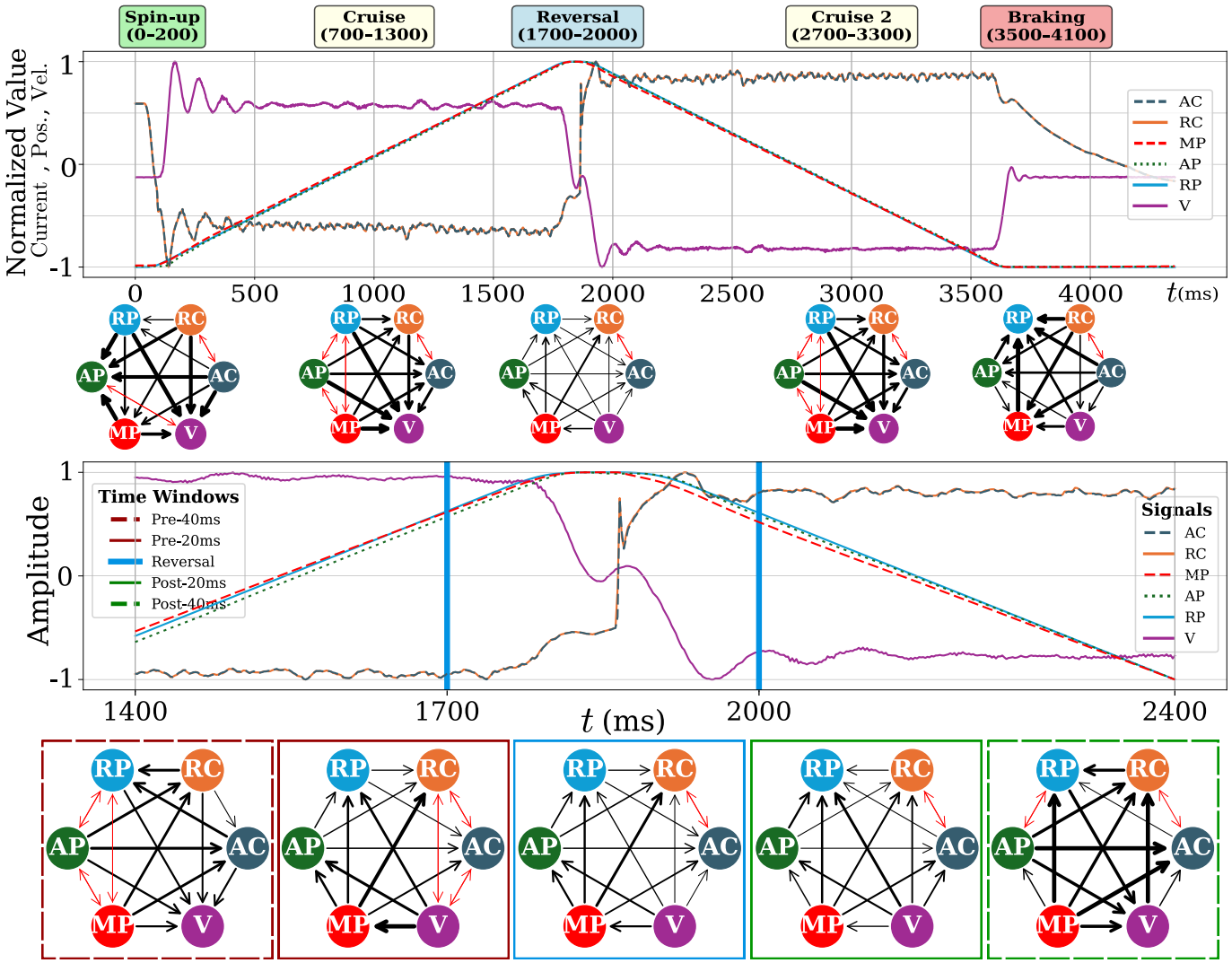


Figure 11 – FNs in the presence of backlash. Figure (a) shows the full motion sequence including spin-up, cruise, reversal, and braking, while (b) shows zoomed view of the reversal region (1700–2000 ms) with two sliding windows before and after reversal.

ing this motion phase compared to the others. During the reversal, velocity becomes the main driver, projecting outward edges to the other parameters. The position variables, on the other hand, show a sequential ordering, with MP affecting AP and AP influencing RP. This stands in contrast to spin-up and braking, where the current signals dominate, and to cruise, where the position variables form a tightly clustered group. The sliding-window analysis shows that this structure does not appear instantaneously but develops over time. Bidirectional edges between velocity and the current signals occur just before reversal, the dominance of velocity emerges at the center, and a partial return to the cruise phase structure is observed shortly after.

The importance of this finding lies in the fact that the FN reflects a mechanical property of the system. When the drive direction changes, mechanical clearance in the gear train must first be taken up before torque is transmitted. During this interval, the usual dynamics are temporarily interrupted, and velocity becomes the organizing variable in the network. The loss of the bidirectional edges between positions reflects the loss of the constant position lag once torque transmission is suspended. In this way, the FN provides a structure of backlash that cannot be directly inferred from the raw signals.

However, the observation also poses unanswered questions. The velocity dominance and the reordering of the position variables are consistent with the expected effect of mechanical clearance, and this pattern appeared reliably across the six reversal experiments conducted at two speeds and three TCP configurations. Nevertheless, it is not yet clear how general this signature is beyond the tested conditions. Its robustness across different robot axes, payloads, drivetrain stiffnesses, or gear types remains to be examined. A more systematic variation of backlash conditions would be required to determine whether this characteristic FN structure consistently emerges and whether it can serve as a reliable diagnostic indicator for wear or transmission faults. If confirmed, FN analysis could provide a valuable, data-driven method for detecting backlash without additional sensors, complementing existing approaches by capturing the dynamics directly at the level of parameter coordination.

5.2 Methodological Limitations

The construction of directed edges depends on the threshold μ applied to the asymmetry measures C_{diff} and T_{diff}. The threshold acts as a control to distinguish directed from bidirectional interactions. Although this flexibility can be advantageous, it also introduces ambiguity, as there is no unique criterion for selecting the threshold that best represents the underlying physical effect.

A further limitation concerns the interpretation of bidirectional edges. By definition, an edge is classified as bidirectional when C_{diff} and T_{diff} remain below the chosen threshold [7, 9]. This can arise either from mutual coupling or from the absence of a directional dependency. From a physical perspective, strongly synchronized systems may therefore appear effectively uncoupled in the resulting network. The method itself does not provide a criterion to distinguish between these cases.

Finally, Min–max normalization was applied to ensure comparability across variables and motion phases. Although this preserves the relative temporal structure, it may reduce the sensitivity to small amplitude differences between conditions [17]. Alternative normalization strategies could be considered in future work.

6 Conclusion

This study provides a network-based perspective on the dynamics of an industrial robot. In particular, FNs are constructed from multivariate motion and control measurements and analyzed across different motion phases and operating velocities. The results show that the resulting FN structures capture coordination patterns that are consistent with established physical interpretations of robot dynamics, while at the same time revealing changes in parameter interactions that are less apparent in the raw time series. FNs describe robot motion through interaction patterns between signals rather than individual signal values, highlighting dynamic effects like backlash that are less visible in the raw time series. This makes them a complement to conventional analysis with potential for diagnostics and deeper insight into robot behavior.

There are several potential applications of these findings. Extending the analysis to different robot systems, multiple axes and payloads would allow its robustness to be tested under more realistic operating conditions. In robotics, FNs could be used to monitor how coordination changes under load or to detect abnormal motion regimes such as backlash without requiring additional sensors. In this context, another promising direction is to link FN analysis with adaptive control tuning, where changes in the network structure could guide online adjustments of controller parameters. Beyond this, another promising line of research is the use of multilayer networks [24], where different sensor signals or time scales are represented as separate layers. Such an approach could reveal interactions that are not visible or

could be used to verify the results obtained from FNs. Overall, these directions highlight the potential of FN analysis for both analysis and future development.

Research Data: The experimental data in figures 9, 10 and 11 are openly available in the Zenodo repository [6]. The dataset includes the recorded time series of the reference joint position (RP), arm-side position (AP), motor-side encoder position (MP), reference motor current (RC), actual motor current (AC), and joint velocity (V) used in the analysis.

Acknowledgements: C.G. is thankful to the DFG for support through Project No. 510246309.

CRediT Author Statement: Khalil El Dabett: Investigation, Software, Formal analysis, Writing - original draft. **Stephan Hansen:** Conceptualization, Investigation, Writing - review and editing, Supervision, Project Administration. **Charlotte Geier:** Conceptualization, Methodology, Writing - review and editing, Supervision, Project Administration.

References

[1] E. Abele, M. Weigold, , and S. Rothenbücher. Modeling and identification of an industrial robot for machining applications. *CIRP Annals*, 56(1), 2007.

[2] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[3] J. F. Donges. Pyunicorn, 2008. pyunicorn [Accessed: 2025-05-03].

[4] J. F. Donges, Y. Zou, N. Marwan, and J. Kurths. The backbone of the climate network. *EPL (Europhysics Letters)*, 87, 2010.

[5] R. V. Donner, Y. Zou, J. F. Donges, N. Marwan, and J. Kurths. Recurrence networks—a novel paradigm for nonlinear time series analysis. *New J. Phys.*, 12, 2010.

[6] K. El Dabett, Stephan Hansen, and Charlotte Geier. Experimental data for: Functional network-based analysis of industrial robot dynamics, 2025. Zenodo [Accessed: 2025-08-15].

[7] J. H. Feldhoff, R. V. Donner, J. F. Donges, N. Marwan, and J. Kurths. Geometric detection of coupling directions by means of inter-system recurrence networks. *Physics Letters A*, 376(46): 3504–3513, 2012.

[8] C. Geier. Code for paper exploring localization in nonlinear oscillator systems through network-based predictions (v1.0). zenodo, 2024. 10.5281/zenodo.15470395.

[9] C. Geier, M. Stender, and N. Hoffmann. Building functional networks for complex response analysis in systems of coupled nonlinear oscillators. *Journal of Sound and Vibration*, 590, 2024.

[10] A. Hagberg, D. Schult, and P. Swart. Circular layout, 2008. Circular Layout [Accessed: 2025-08-15].

[11] A. Hagberg, D. Schult, and P. Swart. Networkx, 2008. NetworkX [Accessed: 2025-08-15].

[12] A. Hagberg, D. Schult, and P. Swart. Spring layout, 2008. Spring Layout [Accessed: 2025-08-15].

[13] P. K. Khosla. Choosing sampling rates for robot control. *Carnegie Mellon University*, 2018.

[14] C. Lehmann, B. Olofsson, K. Nilsson, M. Halbauer, M. Haage, A. Robertsson, and U. Berger O. Sörnmo. Robot joint modeling and parameter identification using the clamping method. *IFAC Proceedings Volumes*, 46(9):813–818, 2013.

[15] M. Makulavičius, S. Petkevičius, J. Rožėnė, A. Dzedzickis, and Bučinskas. Industrial robots in mechanical machining: Perspectives and limitations. *Robotics*, 12(6):160, 2023.

[16] N. Marwan, N. Wessel, U. Meyerfeldt, A. Schirdewan, and J. Kurths. Recurrence-plot-based measures of complexity and their application to heart-rate-variability data. *Phys. Rev. E*, 66 (2), 2002.

[17] N. Marwan, M. C. Romano, M. Thiel, and J. Kurths. Recurrence plots for the analysis of complex systems. *Physics Reports*, 438 (5):237–329, 2007.

[18] U. Schneider, M. Drust, and M. et al. Ansaloni. Improving robotic machining accuracy through experimental error investigation and modular compensation. *International Conference on Advanced Intelligent Mechatronics (AIM)*, 85:3–15, 2016.

[19] Siemens-AG. *SINUMERIK Run MyRobot /Direct Control - Commissioning Manual*, 2018. Technical Documentation.

[20] Y. Song, B. Lian M. Liu, Y. Qi, Y. Wang, J. Wu, and Q. Li. Industrial serial robot calibration considering geometric and deformation errors. *Robotics and Computer-Integrated Manufacturing*, 76, 2022.

[21] E. Uhlmann, S. Reinkober, P. von Stürmer, J. Hansmann, and M.Epping. Innovative machining robot with torque drives (Innovativer Fräsroboter mit Direktantrieben. *Ingenieurs Spiegel*, (3):71–73, 2017. Fraunhofer Publica.

[22] A. Verl, A. Valente, S. Melkote, C. Brecher, E. Ozturk, and L. T. Tunc. Robots in machining. *CIRP Annals*, 68(2):799–822, 2019.

[23] W. Wang, Q. Guo, Z. Yang, Y. Jiang, and J. Xu. A state-of-the-art review on robotic milling of complex parts with high efficiency and precision. *Robotics and Computer-Integrated Manufacturing*, 79, 2023.

[24] Z. Yi, X. Peng, and M. Small. Reciprocal characterization from multivariate time series to multilayer complex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(1): 013137, 2020.

[25] Y. Zou, R. V. Donner, N. Marwan, J. F. Donges, and J. Kurths. Complex network approaches to nonlinear time series analysis. *Physics Reports*, 787:1–97, 2019.