Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (34^{th}cycle)

Virtual measurement of backlash in industrial manipulators

By

Eliana Giovannitti

******

Supervisor(s):
Prof. Giovanni Squillero, Supervisor
Dott. Alberto Tonda, Co-Supervisor

Doctoral Examination Committee:
Prof. Lukas Sekanina, Brno University of Technology
Prof. Eric Medvet, University of Trieste

Politecnico di Torino
2024

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Eliana Giovannitti
2024

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

To my little ray of sunshine, Luca.

Abstract

The backlash phenomenon occurs in the joints of industrial manipulators, significantly impacting the performance and functionality of the manipulator. As robots are widely employed in automated production sites, any deterioration in the robot's performance can adversely affect production, leading to economic losses for the company. A careful maintenance program can prevent this situation, and targeted interventions can be timely scheduled by monitoring the backlash status in the joint. But obtaining a measure of backlash is not simple; the sensors typically required for measuring backlash are not included in the equipment of a standard industrial manipulator. So technicians must resort to manual measurements or estimates derived from measurements of other quantities directly related to backlash (i.e., vibrations). This research work stems from an industrial demand for an automatic tool for measuring backlash based on the use of only the sensors available on the robot. The tool must be the core of a predictive maintenance algorithm to be implemented as a service on an IIoT (Industrial Internet of Things) platform.

In standard industrial manipulators, the only available sensors are the motor encoders, dedicated to measure motor shaft's speed and/or position and to provide closed-loop feedback to the robot controller. An encoder alone is not sufficient to measure the backlash. Backlash is an excess space that occurs between the teeth of mating gears, creating a discontinuity in the transmission of motion from the motor to the link. For a direct measurement of the backlash, two encoders are needed, one at the input (i.e., on the motor) and one at the output (i.e., on the link) of the joint. The state of the backlash in the joint is obtained differentially using the readings from the two encoders. With only one encoder available the only way to obtain information about backlash is to look for any disturbances that backlash may cause in the motor's operation. In some of the few available studies in the literature, it has been shown that in simple bench systems, under specific conditions that excite the phenomenon, backlash effects can be observed propagating through the mechanical

structure up to the motor, where they can be recorded by the encoder. However, replicating the same conditions in industrial manipulators is exceedingly challenging. Even in those cases where it is possible, the signs of backlash propagated to the motor are inconspicuous, manifesting as minor disturbances superimposed on the primary and noisy motor position signal. Furthermore, other disturbances of different origins are also detected by the same sensor. These disturbances, coupled with noise, create difficulties in precisely identifying backlash in the encoder signal.

The research activities described in this thesis have demonstrated that it is not only possible to identify excitations in industrial manipulators that make backlash observable on the motor encoder, but also that the observed disturbance has recurrent and well-defined characteristics, allowing for a unique association with backlash. It is therefore possible to define a “signature”of backlash to be used for backlash identification. When the signature is detected within the signal of an encoder, it indicates the presence of backlash in the joint. Moreover, the amplitude of the signature is directly proportional to the magnitude of the backlash in the joint. Based on these findings, the problem of backlash estimation can be reduced to a fitting problem of a model to a signal.

The tools used include Matlab/Simulink, which was employed to develop a simulator for a robotic joint, allowing for in-depth analysis of the backlash phenomenon, and Python for running optimization algorithms based on evolutionary strategies. Additionally, a test bench was specifically developed, replicating a joint of a real industrial manipulator, and used for analyzing the phenomenon and generating simulated data.

The outcome of this work led to the development of a toolchain for backlash estimation, whose effectiveness was validated using both simulated and real-world data. A second phase of method validation was conducted on a real industrial manipulator, revealing strengths and potential areas of future improvement of the identified approach.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Industrial maintenance

The issue of maintaining increasingly advanced, connected, and intelligent equipments is nowadays central to companies that aspire to pursue continuous improvement in productivity and quality [18]. The same issue is even more important in companies that have high production volumes, where any unexpected failure can translate into severe production downtime and substantial economic losses. It is therefore not surprising that *Industrial Maintenance* has gained an increasing importance as support function for production process for ensuring equipment functionality, on-time deliveries, quality products, and also plant safety [32].

The term *Maintenance* is used to refer to the collection of operations aimed at keeping a system in a state where it can fulfill its intended function [23]. Only with a perfectly running equipment the predetermined production volume is sustained and the desired quality of products is guaranteed. Many are the different maintenance strategies available today [33], they can be grouped within three major classes [47] [12]: *Reactive* Maintenance (RM), *Preventive* Maintenance (PM), and *Predictive* Maintenance (PdM).

What defines the different strategies is the rule that is used to set the right time to perform a maintenance intervention. The rule can be based on a very simple logic or on a more complex one. It can be time based or event based. Rules based on simple logics are easy to implement and inexpensive, but can only provide a rough understanding of the machinery's operating state. Complex logics, instead, allow

for a deep understanding but come with more laborious implementations and higher costs. The more complex and accurate the strategy is, the higher its operating cost will be. Determining what is the best strategy to adopt for a given company is a tough choice that must be made on a case-by-case analysis by looking for the best trade-off between performance and cost. To provide a simple guidance, the main features of the different maintenance strategies are explained below.

Reactive Maintenance follows the rule *"fix it when it breaks"* [42, 41]: an action is taken only after the failure has occurred and a machine component has reached the end of its life cycle. It is a simple and straightforward strategy, easy to implement and with limited initial costs. Advantages however, that are accompanied by the risk of incurring in high repair costs once the fault has occurred. A sudden component fault can in fact adversely affect other machine components and cause production stops or even lead to workers safety issues. Further additional costs also come with this strategy and are related to the necessity to maintain an extensive spare part inventory due to the need to react with a timely response to any potential breakdown. As a conclusion, Reactive Maintenance can be considered a good choice when dealing with non-critical assets where the cost of failure is lower than the maintenance costs.

Preventive Maintenance is based on the principle *"perform regular overhauls so that the machine will not fail"*. The aim in this case is to prevent equipment failures before they occur. In the European standard (PrEN 13306, 1998) Preventive Maintenance is presented as: *"Maintenance carried out at predetermined intervals or according to prescribed criteria and intended to reduce the probability of failure or the degradation of the functioning of an item"* [23]. The planning criteria can be driven by time intervals, utilization counters or also event-based triggers. When machine servicing is scheduled at fixed time or usage intervals (e.g., monthly or every 5000 working hours, in accordance with machine manufacturer recommendations) [15, 8], we refer to the maintenance strategy as Time-Based Maintenance (TBM). Differently, when it is not the time but the occurrence of a prescribed event that triggers a maintenance intervention, then we refer to it as Condition-Based Maintenance (CBM). In this case the operating conditions of the machine are inspected and when they meet a given condition then a maintenance intervention is planned [16]. The actual state of operation of a machine is derived by monitoring the machinery through the use of sensors that capture relevant system information. The signals that are collected are then compared to pre-defined thresholds. When

the value of the signal exceeds the thresholds, an alarm is raised and a repair intervention is scheduled in the short term to prevent the machine failure [27]. An example of CBM could be the temperature control of a component, where temperature is measured and when it exceeds an assigned limit or falls outside defined thresholds, then a maintenance intervention is triggered. To conclude, the Preventive Maintenance strategy is a widely conservative approach that greatly reduces the unscheduled downtime by replacing components at fixed time intervals or at the first signs of performance deterioration. Unfortunately, being conservative implies that the strategy is sometimes ineffective, since replacement is performed event though part could potentially work for a longer time.

Predictive Maintenance: follows the rule *"don't fix it unless you know it is going to break soon"*. The goal of Predictive Maintenance (PdM) is to avoid unnecessary maintenance operations by predicting the time to failure of a component, or a machine, and possibly estimating its residual useful life (RUL). Data are continuously gathered from the machines and used to infer the condition of the equipment to finally predict future breakdowns. Historical data and patterns of anomalous behaviours are used to foresee the impending failure. The forecast is made by predictive models based on sophisticated statistical techniques (Machine Learning tools). Hence, the maintenance schedule is set as close as possible before the expected failure point to maximise the useful life of the component. In this way, there are no unnecessary substitutions and machine downtime is reduced, with a positive impact on spare part inventory management, productivity, and maintenance expenses. Unfortunately, PdM is the strategy which comes with the highest operating costs. This costs are due to the need to install sensors to monitor asset conditions and to the effort required to develop accurate predictive algorithms. In the end, PdM remains the most profitable choice for assets that are very important, difficult to repair, or with a high failure frequency, as it leads to substantial savings in the long term.

Both Predictive Maintenance and Condition-Based Maintenance are based on a *"condition monitoring"* process [1], i.e., the continuous monitoring of equipment, to estimate the equipment status and evaluate the degree of component degradation over time. Because of this they are often considered as similar and reported in the literature as belonging to the same group, that of PdM. In this case, the group lists together all the condition-based strategies whether using thresholds or predictive models. In this thesis, on the other hand, we agree with the classification that is reported in the previous pages where CBM and PdM are considered as belonging to two

different groups. They are considered different since their methods are different as different is their final objective: while PdM relies on sophisticated models (analytical, statistical, ML-based, etc.) to predict failures in the future, CBM relies on simple thresholds only aiming to determine whether something is wrong in the present moment. Moreover, CBM is based on conditions, PdM is based on patterns or stored information. Figure 1.1 provides a fast overview of the classification of maintenance strategies described on the previous pagers.

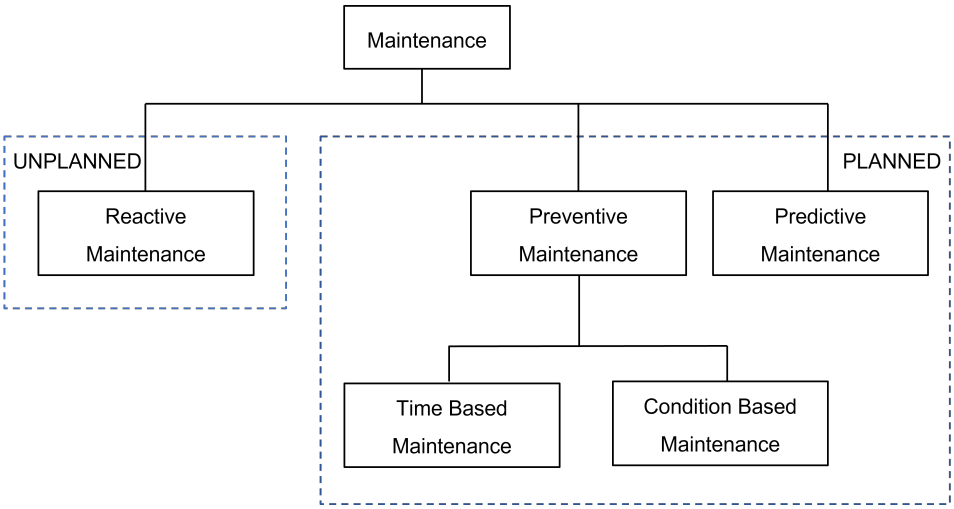


Fig. 1.1 Overview of different maintenance strategies

A clear understanding of the action strategy behind the three main maintenance approaches comes from the so-called P-F curve [7], a graphical model to visualize the asset's progression toward failure. The curve, presented in Figure 1.2, represents how the performance of a machine declines over time, starting from the asset's installation down to the moment of breakdown. The x-axis of the graph represents the time, the y-axis represent the performance, and thus the health, of the machine.

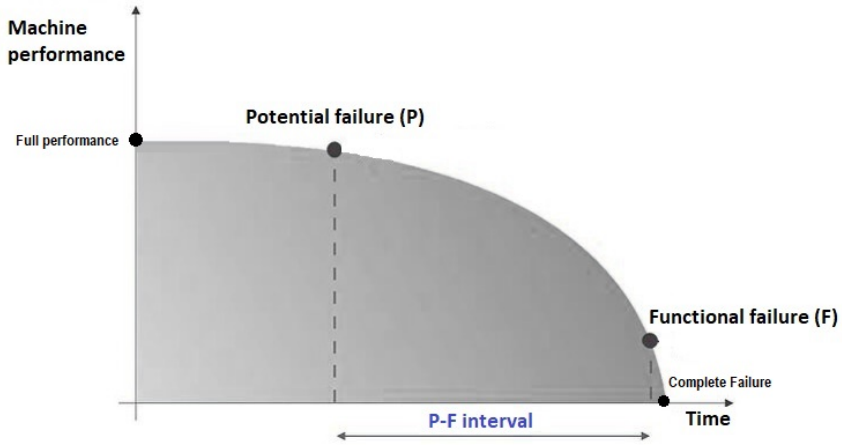


Fig. 1.2 The P-F curve of a machine, or a component, with the two primary points (P) and (F), along with the time interval P-F defined by them.

Starting from the far left of the graph, the $t = 0$ point of the curve correspond to the time of machine installation when the machine performance level corresponds to the level defined in the design phase. Then, at the flowing of time thus moving on the right of graph, performance remains unchanged until the first signs of machine deterioration begin to appear. Observable signs can be an increase in noise while functioning, the appearance of oscillations on movements, a sudden decrease in pressure levels etc... This is the point **P** of the *Potential failure*. Starting from this point machine deterioration slowly starts to increase until reaching the point **F** of the *Functional failure*. This is the moment when the asset becomes unable to fulfill a function with the minimum acceptable performance level as defined by the user. The last representative point is where the curve meets the X-axis. It corresponds to the *Complete failure* of the machine when the equipment will no longer be operational.

The time window between the P and F points is the most important part of the curve. It is called the *P-F interval*, and represents the available time margin to plan and schedule a maintenance intervention before the functional failure point is reached. The exact point in the interval that is chosen as the best moment to perform machine maintenance depends on the maintenance strategy that has been adopted, see Figure 1.3.

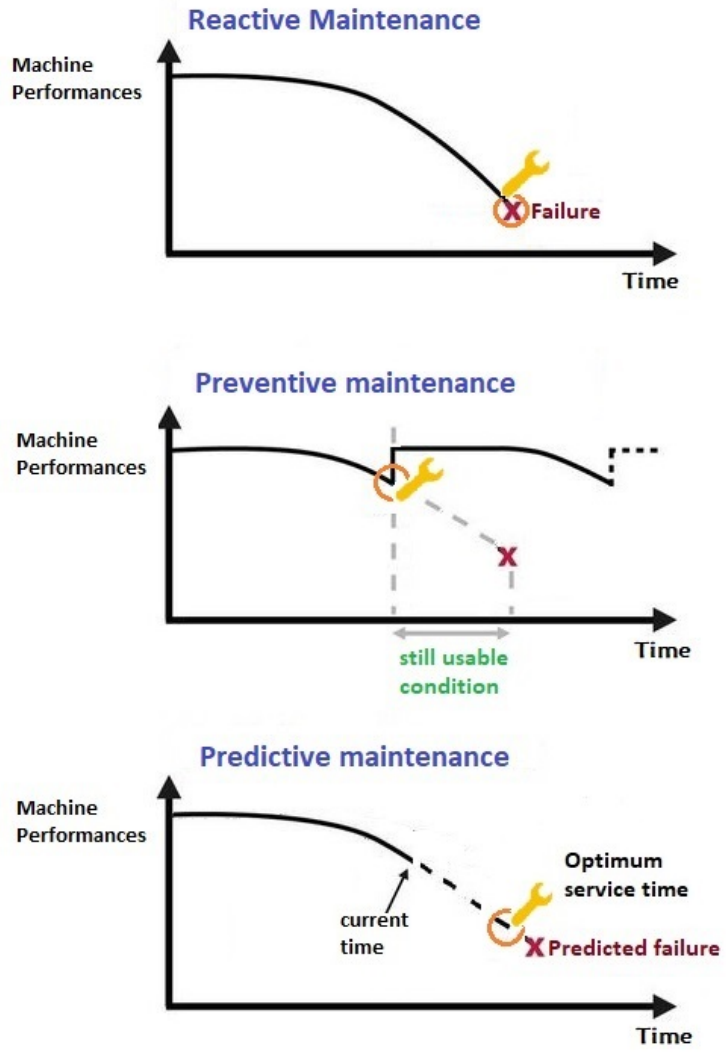


Fig. 1.3 Maintenance strategies comparison

It clearly appears in the Figure above that PdM is the most effective strategy: it allows for a careful planning of the time of intervention and maximizes the useful life of the component/machinery.

As a conclusion to this overview on the approaches for industrial maintenance, the Table 1.1 is given. The table is intended to provide a simple and quick summary of the basic principle behind the strategies presented, togheter with their pros and cons, and the magnitude of the cost of their implementation.

Table 1.1 Maintenance strategies brief overview. The table briefly summarizes the characteristics of the various strategies and their pros and cons.

Strategy	Summary	Cost to implement	Pros	Cons
Reactive	Fix it when it breaks	Low	• Simple to understand and use • Minimal implementation effort • Ideal for low-priority equipment	• Unscheduled down-times • High repairing costs
Preventive	Fix it on a regular basis to keep it up and running (Time-based or Condition-based overhaul)	Average	• Maintenance on a predetermined schedule • The "safest" strategy	• Risk of over-maintenance • Requires the use of sensors (if CBM)
Predictive	Fix it only when it's going to break	High	• Accurate prediction of the time to failure • Maintenance on an as-needed basis • Less production time lost • Components service life maximized	• Expensive to set up. Requires: sensors, software, and specialized skills to be implemented

1.2 Importance of maintenance for industrial robots

The problem of maintenance becomes even more important when it comes to the world of industrial robotics where efficiency, productivity and cost savings are the main focus. Manufacturing company come to the decision to adopt the use of robots to boost their production output. With robots working shifts can be extended and the repeatability of performance can be increased without rising any additional labour cost [3]. However, in order to fully benefit from these improvements, it is important that robots are always in operation and that they are working to their full potential. It is only under these conditions that a high production volume and the quality of the parts produced can be guaranteed. Thus, the great role that an efficient maintenance strategy can play in this context is evident.

The most common form of maintenance program for robots is Preventive Maintenance. It involves pre-planned maintenance activities as lubrication, cleaning, and minor part replacements in order to correct or prevent any problems that may arise. In this way the performance of a robot remains reliable and the life span of robots is increased. The maintenance interval is typically defined through experience or through original equipment manufacturer (OEM) recommendations. So, the intervention schedule can be time-based (i.e., every 12 months) as well as meter-based (i.e., after 50,000 hours in use) that is referred to the utilization of the robot. A typical robot controller always provides different types of meters that can be used for this scope. Meters can count the hours of use, the work cycles performed, the distance travelled by a robotic joint, or even the number of reversals of motion that a joint makes. All these maintenance recommendations are typically reported in the user manual of the robot or in the maintenance sheets of the production line where the robot is installed.

A typical example of maintenance plan for a robotic manipulator is the following: “on a daily basis, operators should inspect the robot for any damaged parts or any oil or grease leaks. Once a month, users should inspect the ventilation of the controller’s cooling fans to allow for proper air flow. Every three months, cables should be inspected for damage or premature wear and their connections should be secured. In addition to daily, monthly, and quarterly maintenance, a larger preventive maintenance plan is recommended once a year or every 3,850 hours, whichever comes first. This means that if your robot hits 3,850 operating hours before a year, the yearly preventative maintenance will be needed at that point regardless of being

under a year from the last maintenance or implementation. Yearly maintenance tasks include backing up the controller and teach pendant, replacing robot and controller batteries, replacing robot grease, inspecting the brake system, observing and listening for irregular movements or noise, and thoroughly inspecting the overall condition of the robot ". [1]

But robots are very complex and versatile machines and the actual working conditions under which they will operate will differ from case to case. Even two robots of the same model will experience very different working conditions while performing different working cycles, in different load conditions, or different operating temperature, etc. As a result any of the above maintenance prescriptions lose its value of generality and is no longer indicative of the real wear of the robotic components. Thus easily comes the risk of carrying out the planned maintenance when it is actually too early or too late for the robot.

Fortunately, industrial robots maintenance has undergone a great evolution in recent times, with the advent of Industry 4.0 and the IIoT (Industrial Internet of Things), things have started to change. Robot manufacturers have started working on the development of IoT platforms for collecting, storing, analysing and visualising data from industrial assets, and have started to prepare industrial manipulators for communication and data sharing towards these platforms. This has opened up the possibility of a continuous monitoring of the robot's actual conditions and thus the adoption of strategies such as CBM and PdM best suited to the flexible and varied working conditions of robots.

Threshold-based CBM tools are the most common ones in industrial practice. Typical examples of CBM strategies for industrial robots are:

Temperature monitoring where the temperature of critical components such as motors and actuators is monitored. High temperatures may suggest overheating or increased friction, signalling the need for maintenance to avoid failures.

Analysis of current and energy consumption where current and energy consumption of robotic motors and actuators is analyzed. Deviations from normal operating parameters may indicate problems such as mechanical friction, electrical problems or impending component failures.

[1] https://robotsdoneright.com/Articles/preventative-maintenance-for-your-fanuc-robot.html

Vibration monitoring where the vibration level of robotic components is measured through specific sensors. Unusual vibration level may indicate misalignments, wear or other problems.

In all the above cases, the robot's condition is monitored and compared with a priori defined thresholds. When the reference signal exceeds the set threshold, a maintenance intervention is planned in the short term.

PdM strategies for robots are more rare in practice. In PdM data from robot are analyzed to identify patterns that precede failures and predictive models are used to forecast the moment of occurrence. These tools are very effective but at the same time very expensive because they are based on analytics that are highly customised on the specific use case. They therefore happen to be rarely adopted in practice, while a larger number of references can be found in the academic field. An example in literature of the application of Predictive Maintenance to industrial robots can be found in [5], where statistical tools as the Kernel density estimate (KDE) and Kullback-Leibler distance are used to get a measure of wear in robotic joints. Data-batches are collected over time on a repetitive task of a robot and compared with a reference batch collected at the beginning of the robot operation. Through the definition of a fault indicator and through the analysis of its evolution over time the state of the system is monitored.

A different solution is the one provided in [6]. Also in this work, the current signal of robot's motors is the object of analysis. But this time currents are used to obtain information on the positioning accuracy of the robot. Row data are processed and, after a feature engineering step, a predictive model based on multiple linear regression is trained. The predictor variables of the motor are the feature extracted from time-series of the robot's motor currents and the response is the positioning accuracy. One more example of data-driven methods for industrial robotics can be found in [13], where an health monitoring system based on an unsupervised clustering method (K-Means algorithm) and an AI anomaly detection module is developed. Since no faulty data are available, only data recorded on the normal functioning of the robot are used to define the reference clusters. Any deviation of new data form this clusters is considered as anomaly. Moreover, new data instances that are classified as normal are used to update the clusters so as to to follow the dynamism of the system, the resulting deviation of the clusters centers over time is used as metric to predict the moment to set a maintenance intervention on the robot.

Since predictive analytics continues to evolve with advancements in Machine Learning and Artificial Intelligence, industries have increased their interest in this powerful tools to gain insights into problems that can arise and improve decision-making processes.

This orientation also underpins the work in this thesis that is the result of an Industrial Doctorate, a collaboration between industry and academia for the realisation of a research project aimed at industrial application. The topic is related to robot maintenance in Industry 4.0. The aim is to develop the basis for the creation of a predictive maintenance tool for assessing the level of backlash in robotic joints. This tool will be part of a package of services offered by the IIoT platform developed by the industrial company itself (COMAU S.p.A.).

1.3 Maintenance and backlash faults

One of the most complex problems addressed by robot maintenance is backlash. Backlash is a mechanical phenomenon that affects many types of machinery, when considering robotic joints, it refers to the clearance or play between mating components, such as gears or screws. This clearance can result to imprecise motion and reduced accuracy, and to vibrations and oscillations of the robot. When not promptly addressed, backlash can progressively increase up to the system breakdown. The clearance can be caused by several factors, such as the erroneous assembling of the mechanical parts in the joint, or by the wear accumulating over time. A system for condition monitoring that can detect the level of backlash in a joint can offer great help in managing the maintenance of a robot. Unfortunately, direct measures of the backlash are difficult to be obtained since they require dedicated sensors to be installed or the manual intervention of an expert technicians.

The goal of this research project is to create a new backlash measurement methodology with the following features:

- it doesn't require the use of extra sensors beyond those on a standard robot;
- it can be done automatically, reducing the need for human intervention;
- it can be easily integrated in a IoT platform for robot monitoring.

The resulting procedure is expected to provide an estimate of the backlash value in the robotic joint to facilitate efficient planning of maintenance interventions.

Scientific literature about this specific problem is quite scarce, and few are the solutions that can be applied in a real industrial context. Hence the need to start with a joint research activity between industry and academia that would thoroughly investigate aspects of the problem and lead to the development of a new solution compatible with industrial requirements. The activities performed, the experiments conducted, the results obtained and the future steps are described in this thesis.

1.4 Thesis structure

This thesis consist of 9 chapters and is organized as follows: Chapter 1 introduces the concept of industrial maintenance, its benefits and the main maintenance strategies; Chapter 2 narrows down the focus on the maintenance of industrial robots and, even more specifically, on the problem of faults related to backlash; Chapter 3 describes the specific problem under investigation, the backlash in robotic joints, and the verification of the hypotheses underlying the development of the activities; Chapter 4 concerns the development of models used to support activities and the generation of simulated data for use in subsequent phases; then in Chapter 5 the proposed approach to solving the problem and its various building blocks are explained; Chapter 6 provides an overview of the Machine Learning tools hypothesized as elements of the analysis toolchain and a benchmark to guide the final selection; finally, once the best tool is chosen, the results obtained from the first application of the complete toolchain are presented in Chapter 7. Once the validity of the proposed method has been verified, its generalization to a broader use case, i.e., the full robot, is presented in Chapter 8. The contributions and accomplishments of the thesis are outlined in Chapter 9, along with some suggestions for potential areas of future research as an extension of this work.

Chapter 2

Problem definition

This section briefly outlines the phenomenon of mechanical backlash and the complexities associated with its measurement in robotics joints.

2.1 Mechanical backlash

Mechanical backlash refers to the amount of free movement, i.e., *"play"*, that exists in a mechanical system where motion is transmitted from one part to another. It is particularly common in systems that involve gears, linkages, or other moving parts that must engage with one another to produce motion.

When there is backlash in a system, the result can be that the system does not respond immediately to changes in input or load, and there may be a delay or "dead zone" before motion is transmitted. This can lead to uncontrolled behavior in the system, especially when rapid changes in input or load occur. Backlash impacts accuracy and repeatability of the mechanical system as the looseness in the system can cause variations in the output with imprecise and jerky movements.

In general, mechanical engineers try to minimize or eliminate backlash wherever possible, as it can reduce the efficiency, accuracy, and lifespan of a system. Backlash minimization can be done through careful design, material selection, or manufacturing techniques that reduce clearances between components. Control strategies may also be employed to compensate for the effects of backlash during operation. However, completely eliminating backlash may not always be practical or

cost-effective. Typically, a small amount of play is intentionally tolerated because practical manufacturing tolerances involve dimensional variations, and the small amount of backlash provides running clearance preventing the mating parts to get stuck into each other [40]. Moreover, a small backlash also helps to facilitate the movement of the parts because it allows the lubricant to penetrate well into all spaces between the mating parts. Lastly, it reduces system overheating as too tightly meshed gears are more prone to rubbing and heating. So designers have the hard task to define the allowable amount of backlash in the system [26] by balancing the need for a low backlash with with other factors, such as system durability and manufacturing costs. Unfortunately, most of the benefits of design are lost as wear causes the initial distance determined by the project to increase over time. that is why regular monitoring of backlash is crucial, especially in applications where precision is critical, such as robotics.

2.2 Backlash in gears

Backlash in gears is defined as the exceeding space between the thickness of a tooth and the width of the engaging space in the mating gear [23]. It is usually measured in millimiters, along the pitch circle of the gear [42] and is given by the distance the gear teeth has to travel to engage a different mating teeth when motion is reversed, see Figure 2.1

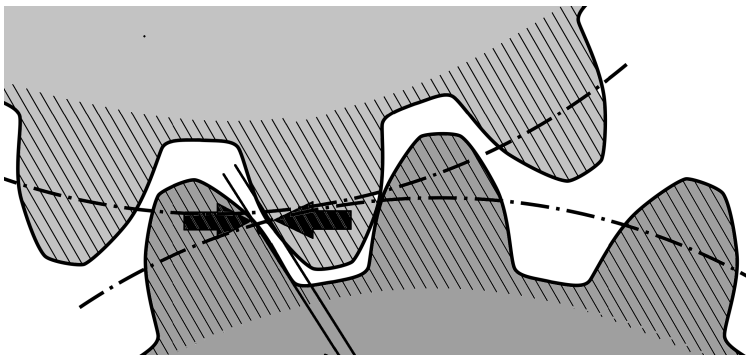


Fig. 2.1 The backlash gap between mating teeth of gears

It is specifically on reversals of motion that the effects of backlash become more evident and the "play" cause the gears to jerk or produce a clunking sound, especially if the gears are under load. The phenomenology of a reversal of motion in the case

of gears with backlash is explained in Figure 2.2. There are three stages in a reversal of motion. In the first stage, which is represented on the left, teeth are mating and the motion is transferred from the driving (lower) to the driven (upper) gear. Then the driving gear reverses the motion, it's the second state, and the driving tooth detaches from the mating tooth on the right and crosses the backlash gap to seek contact with the other tooth on its left. Ultimately, in the final state, the driven gear and driving gear begin to move in reverse as the contact between them is restored.

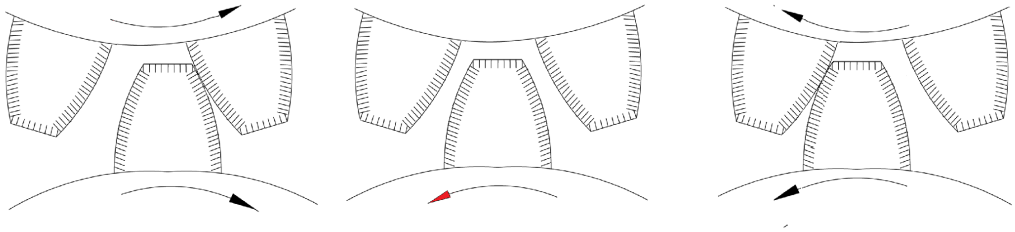


Fig. 2.2 Backlash in reversals of motion.

A measure of the backlash in coupled gears is derived by measuring the traversing space of the drive gear during a reversal of the movement. A method consist in keeping one of the two gears locked and rotating the other until it makes contact with the other tooth while measuring the travelling space with a precision encoder or with a dial indicator, see Figure 2.3.

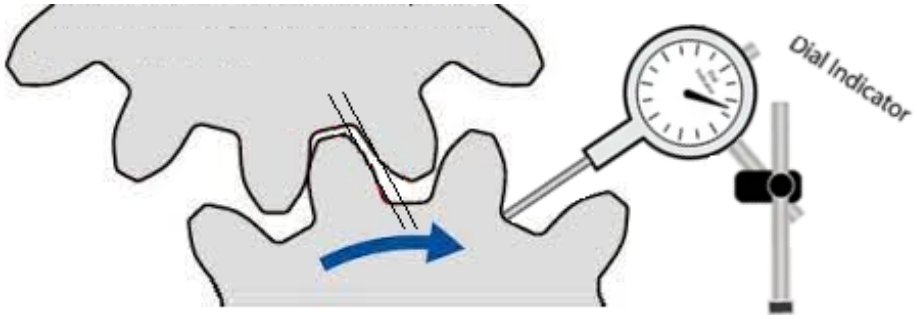


Fig. 2.3 Backlash measurement in coupled gears through a dial gauge

Another possible way to determine backlash is to measure both position of the driving and driven gear and then get the measure of backlash as the relative displacement of the two [58]. This displacement will be zero in the ideal situation of no backlash or when the mating gears are in contact, and non zero when backlash is present. More in detail, when teeth are in contact the driving wheel conducts the driven wheel, which will move by a displacement equal to that of the driving wheel.

In this case the relative displacement will be zero. In case of backlash, instead, there will be a time when the two wheels are not in contact and in that time they will be independent and with different positions. So while traversing the backlash gap, the relative displacement will be non zero and the maximum value of this displacement will be the measure of backlash [35].

To measure backlash in a complete gearbox chain the same logic that applies to a single pair of mating gears can be used. In this case the displacement should be referred to the input and output ends of the transmission since, in this instance, the transmission's backlash is given by the sum of the backlash of all components. This is also the case of robotic joints, mechanical components made of a chain of movable parts and hence also affected by backlash.

2.3 Backlash in robotic joints

Robotic manipulators are articulated structures composed of segments of different lengths, i.e., *links*, that can move relative to each other thanks to movable parts called *joints*. They comprise a mechanical configuration that resembles a human arm. With the use of articulated joints with multiple degrees of freedom, the arm provides a large range of movement. Articulated robots with six degrees of freedom are the most common and are the most widely used in industry. Other types may only have four degrees of freedom or even fewer, an example is shown in Figure 2.4.

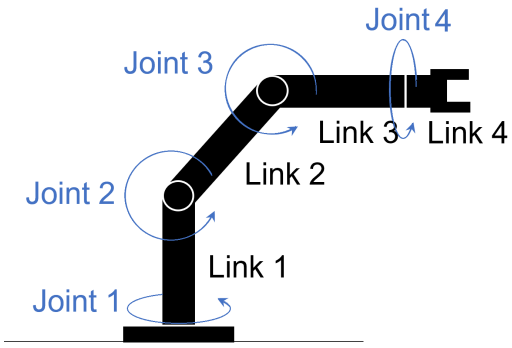


Fig. 2.4 Links and joints of a 4link robotic manipulator

Robotic joints are made by different mechanical components such as gears, belts or screws. Joints are actuated by motors and their role is to transfer the torque

required for movement from the motor to the link. The more the transmission is continuous the more the movement of the link is smooth.

Backlash in robotic systems refers to the clearance or play between mating components within the joint. This mechanical play can impact the accuracy, precision, and repeatability of robotic movements. The effects of backlash on robotic performance are: loss of positioning and path accuracy, (i.e., the ability to reach with precision a programmed position in space or precisely follow a path), loss of repeatability (the ability to return to the exact same position in space again and again). When the level of backlash is near the value imposed by design a good control of link movements can be obtained and the torque provided by motors is smoothly transferred to the link. On the other hand, as backlash arises, more space is formed between gear teeth, and torque transmission becomes no longer continuous. Therefore, there is no longer any guarantee that the movement of the link is fully controlled [28]. Particularly during motion reversals, when the driving wheel changes verse of rotation and changes the mating tooth on the driven wheel, then an interruption in torque transmission occurs. Teeth contact is shortly lost as the driving tooth detaches from the driven tooth to cross the gap space between teeth and find contact again. In this while, the driving and the driven part are not in contact so the load movement is free and no longer driven by the motor. The consequence is that the robot move less precisely and, even worse, that vibrations of the structure may occur due to impacts that can arise when teeth contact is restored.

Measuring robotic joint backlash is not an easy task. Many are the possible strategies depending on the available sensors (i.e. accelerometers, torque sensors, encoders, ...) or measurement procedures (static or dynamic measurements). Backlash in robotic joints is typically measured in units of linear or angular distance, such as millimeters or radians, and represents the amount of movement that one part of the joint can undergo before the other part responds.

The traditional measurement procedure for backlash in robotic joint relies on a static measurement that is manually performed by an expert technician, and is carried out with the help of a dial gauge.

The step-by-step procedure to measure joint backlash with a dial gauge is:

1. Keep the manipulator at rest

2. Brake the motor of the joint under measure to held in place the input part of the transmission and prevent any movement
3. Leave the output part of the joint (i.e., the following link) free to move
4. Secure the dial indicator on a fixed point on the stationary part of the joint
5. Position the stylus of the dial indicator on a reference position on the free to move part of the jont
6. Apply force to the free link to manually rotate it in one direction until it stops
7. Record the maximum displacement shown on the dial indicator
8. Apply force to the free link in the opposite direction until the link movement stops
9. Record the maximum displacement in this direction
10. Calculate the backlash as
    Backlash = (Maximum displacement in one direction + Maximum displacement in the opposite direction)
11. Repeat the process more then once to ensure consistent result.

The measurement setup is reported in Figure 2.5. The measure is relative to joint6 (before the black flange) of the manipulator. The input force is applied through the black bar connected to the robot flange. Then, the resulting link displacement is read on the dial gauge. This procedure is impractical when performed in a production facility, since the intervention of skilled personnel is required and an interruption of production is necessary. High maintenance costs and revenue losses thus occur for the manufacturing company. Moreover, the results obtained with measurements on a static robot can be different from the reality of an operating robot that is continuously moving and therefore subject to heating. The gear tooth's shape, in fact, is affected by thermal factors which can slightly widen or narrow it. This can change the mating behaviour and, consequently, the measure of backlash.

When moving to dynamic backlash measurements in robotic joints, the simplest method is to use two position sensors, one at the input (on the motor) and the other at the output (on the link) of the joint. So position measures collected during dynamic

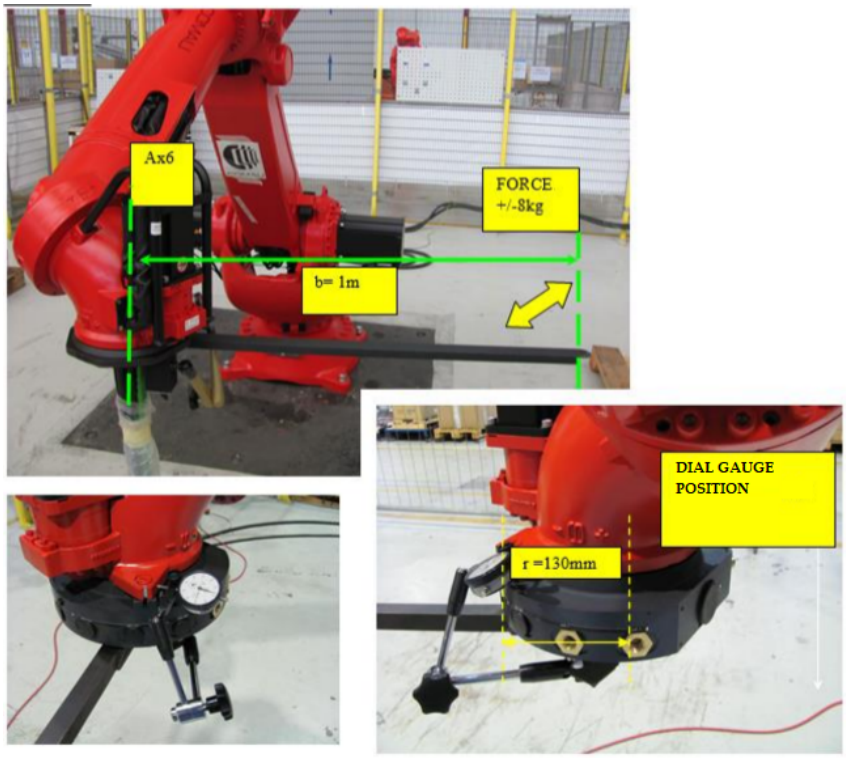


Fig. 2.5 Backlash measurement manual procedure.

measurements can be collected and compared to find the relative displacement between motor and link that identifies the play in the joint. Unfortunately, almost all industrial robots are only fitted with one single encoder, so backlash measurement under dynamic conditions becomes very difficult and require careful study to find a suitable solution.

2.4 Challenges in measuring robotic joint backlash

Great effort is put into industrial research to develop automated, quick, and dynamic backlash measurement processes. The goal is to move towards solutions that are as repeatable, reliable, cost-effective and effortless as possible. The ambition of this research work goes exactly in this direction and aims to develop an automatic backlash measurement tool that can be integrated into the IIoT services platform developed by COMAU (a global company that manufactures robotic manipulators and develop industrial automation solutions), and provide insights to facilitate machine

maintenance planning. The tool should be able to detect when gear play in robot joints is excessive, as in [24], and suggest the best moment to schedule a maintenance intervention on the robot. The tool should be both automated and simple to use, and easily applicable to all of the robots that are currently installed at customers' production sites. It must therefore be based on a method that avoids the robot needing to have extra sensors in addition to those available by original design. This is a very challenging condition to meet because standard industrial manipulators usually have only one position sensor per joint, which is not sufficient to measure the backlash directly [60]. In nearly all of the literature-available methods extra accelerometers, output position sensors, or torque cells are added to be used in conjunction with the standard motor position sensor to determine the backlash gap in joints.

The first reference that can be found in literature about the problem of estimating backlash in a robotic joint is given by the work of [11]. In this work an accelerometer mounted on the link moved by the joint is used in conjunction with measurements taken on the robot's motor to derive an estimate of the backlash. They suggested a method for detecting backlash in robotic systems based on coherence analysis of link acceleration and the motor voltage. Later on, other researchers devoted their interest to the analysis of the backalsh in industrial robots, like [25, 24] and [31]. They too, like their predecessors, chose to use an accelerometer for their studies but relaying on a different technique: the vibration analysis (i.e., wavelet analysis). In [30] an entirely different approach can be found, where a gyroscope mounted on the link is used in place of the accelerometer. Here researchers tried to identify backlash issues by detecting the small impacts in the transmission due to tooth-to-tooth collisions caused by backlash. Also in the work [50], an attempt was made to quantify backlash by analysing impact marks, but this time the sensor used is a torque sensor mounted at the joint input, i.e. on the motor. Another approach based on the use of a torque sensor and a position sensor is that in [22]. Lastly, in [4, 29] a method for an online backlash identification strategy employing two position sensors and a Kalman filter was provided .

All the aforementioned references are based on the use of an extra sensor which is added to the robot's standard equipment. So, these solutions are rarely or never adopted in industrial settings since having an extra sensor means raised costs for robot manufacturer. Integrate a new sensor means new costs to purchase the sensor but also new costs for a more complex robot design. Measuring methods which instead rely on the standard available on-board sensors, the motor-mounted position

sensor, are more widely accepted. Therefore attention should be paid to the literature on indirect backlash measurement, and in particular to the methods using the motor position sensor as the sole measuring sensor. Unfortunately, the literature on this subject is very scarce and the references that can be found mostly refer to simple motor-transmission-load systems and not to robotic joints. An example is the short summary given in the work of [59].

Below is a more thorough summary of some of the research on the use of a single position sensor to estimate the backlash in a mechanical gearbox that is currently available in the literature. Along with the description of the method used, the disadvantages of these methods with respect to the specific use case of a robotic joint will also be illustrated.

In the work of [17] two distinct techniques for backlash identification using only the motor-side position sensor are presented. The first method was based on the hypothesis that the load position was stationary. Under this condition it is only the input part of the transmission, the part connected to the motor, which can move while crossing the free space within the backlash gap. It is therefore enough to know the displacement of the motor to reconstruct the size of the backlash. The method is based on finding the right torque value to be used to drive the motor. The value has to be smaller than the minimum torque needed to move the load, that is supposed to be stationary, when the backlash gap is closed. But also big enough to drive the input part of the transmission through the free play space when the gap is open and the load is disengaged. These impulses are easy to find in the case of a simple test system, as the one used in paper, but very difficult to be find in a robotic joint where there are strong static friction phenomena that do not allow to move the joint with small valued torques. The second method presented in the same work combined the use of the position and current signal of the motor. In particular, the derivative of the current is analyzed to search for the instants at which the gears engage and disengage. Once the current peaks corresponding to these instants have been identified, the intensity of the backlash is determined by measuring the displacement of the motor in the interval between the two peaks. Also in this case the method cannot be easily exported to robotic joints as ad-hoc filtering is needed to smooth the current signals and ease the accurate detection of the tiny peaks connected to the decoupling/engagement condition. The filtering has to be adjusted based on case-by-case testing conditions, making the method hard to generalise and therefore unfeasible for the use in industry.

Similarly, [34] estimated the backlash together with the friction in the joint by analysing the speed and current signals from the motor. Through a combined analysis of these signals, the method identified the two significant time instants when the motor decouples, t_1, and engages, t_2, the load. After that, the velocity was integrated on the $[t_1;t_2]$ time interval to get the backlash gap measurement. In this work, a real robot was used instead of a basic motor-transmission-load test bench. The robot was an industrial SCARA manipulator and ad-hoc test movements were designed to facilitate the identification of instants t_1 and t_2 in the signal. Unfortunately, such conditions cannot be easily replicated in an articulated robot which is the standard industrial manipulator.

As in the previous work, [56] proposed a method relying solely on motor velocity measurement to estimate backlash. Once again, the backlash gap was computed by integrating the velocity signal during the time interval corresponding to the moments of decoupling and engagement in the mechanics. Notably, the determination of these instants did not necessitate the use of the current signal. The system was excited by using a triangular test function to drive the gears to traverse the backlash. Then, by examining the features of the resulting motor speed signal, the engage and disengage instants were identified.

Finally, [52, 53, 54] introduced new approach for identifying and quantifying backlash by observing rapid variations in the speed of the driving gear within the transmission. These speed changes resulted from impacts between the teeth of mating parts that occur when the backlash space is large. The authors additionally demonstrated that the magnitude of the impacts was directly proportional to the size of gear backlash. All the test performed in the work were executed on a open-loop simple test-bench excited by sinusoidal input signals.

The approach introduced in this thesis represents an extension of the aforementioned solution. It broadens its applicability to closed-loop systems and to systems with more intricate mechanical structures as robotic manipulators. Moreover, it streamlines the test conditions by removing the necessity for a sinusoidal input signal. Futhermore, the research work described in this thesis introduces an innovative element that consists in identifying the backlash during a continuous movement along a same direction and not during a reversal of motion. In this scenario, it is not the change in movement direction but the effect of the force of gravity acting on the link that creates the conditions to open and close the backlash gap. The last strong

point of the method here presented is that it can easily be adopted in an industrial scenario since it is fully automated thanks to the use of a meta-heuristic algorithm for the analysis of data.

Chapter 3

Problem analysis and hypothesis testing

3.1 The test bench

The first part of the research activity was dedicated to the problem understanding, and to the verification of the hypothesis behind the whole work: the observability of backlash by the encoder on the motor. This step was essential to verify that traces of the presence of backlash can be found on the signal captured by the motor encoder, and that such traces can be used to provide information about the size of the backlash in the joint.

To this intent, a test bench reproducing the joint of a real COMAU industrial robot was assembled and used for tests. The bench was used to:

- find the best test conditions to highlight the backlash presence on the motor encoder signal,
- provide data,
- be the reference for the construction of a Matlab/Simulink model of the system.

The bench is showed in Figure 3.1. It is a simple mechanical system made with one of the six joints of a small industrial manipulator from the COMAU robot family. The model of the robot cannot be reported due to a confidentiality agreement with the

company. The robot's joint was removed from the robot structure and mounted on a bench. In this way, any potential interference between the joint under test and the other robot joints is avoided and the backlash effect on the encoder can be assessed more clearly. The bench is composed of a motor, a transmission made up of a reducer and a transmission belt with a pulley, and a load consisting of a cast iron mass which simulates the robot link. In the system there is also an encoder that is connected on the motor. The encoder is a 19-bit absolute encoder with a 1×10^{-5} rad resolution. The only measures that are available on the system are the motor position, which is provided by the encoder, and the current absorbed by the motor, provided by the robot controller.

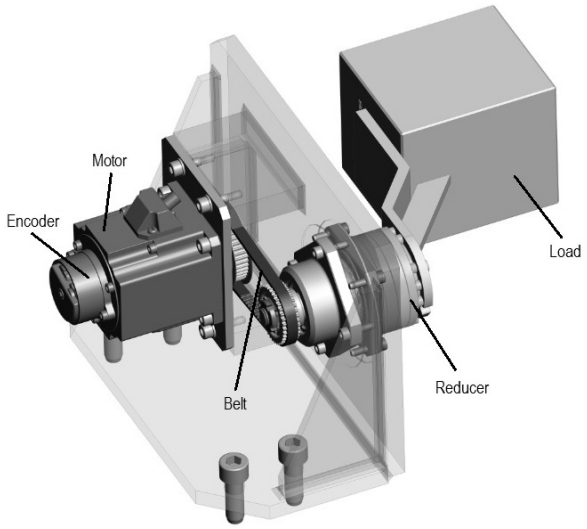


Fig. 3.1 The test-bench specially designed to study the backlash phenomenon. CAD drawing form COMAU

The reducer used for test bench was known to be affected by backlash. It was object of an endurance testing campaign, performed in the COMAU facility, in which it was exposed for a long time to highly demanding working conditions.The test concluded after a predetermined time interval, and the reducer was deemed unusable due to exhibiting a level of backlash beyond the acceptable range. Unfortunately, all the valuable information about backlash evolution over the endurance test are not available since the activity was performed before the start of this research work.

3.2 Observability of backlash through the motor encoder

The test bench was employed to evaluate the practical feasibility of measuring backlash using the motor encoder. To this intent, a variety of test conditions have been investigated to determine which one caused the backlash effects on the encoder signal to become visible. Tests were conducted by exciting the system with many different types of input signals, and setting different working conditions. Tests were carried out at different speeds, on short and impulsive movements or on long and gradual movements, on reversals of motion and on continuous movements in one direction only. Relying on what was reported in the literature it was expected to see traces of backlash on the reversals of motion. Surprisingly, it was noticed that the only circumstance in which the presence of the backlash was clearly evident was not that linked to the reversals of motion but the one linked to the continuous movements in one direction only. The experiments proved that backlash disturbance became clearly visible on the motor speed signal when running the motor at a constant speed while having the rotation axis orthogonal to the gravity force. While moving at constant speed the system operates in a steady-state condition: speed is constant and effects like inertial phenomena or static frictions are absent. The gravity force is the only external force acting on the system and is fully compensated by the motor control. These conditions are very favourable for signal analysis and to the identification of possible disturbances such as those generated by the occurrence of backlash. When the system is affected by backlash it is gravity that triggers some small impacts between mating teeth in gears and makes the backlash phenomenon visible. These impacts propagate through the transmission chain up to the motor and are then sensed by the encoder. The backlash presence in the robotic joint is thus visible and appears as a disturbance superimposed on the speed signal of the motor.

The speed signal is shown in the figure below, Figure 3.2. Superimposed on the constant speed a very small disturbance is visible. The disturbance has a well-defined shape and, since is not present in a factory new joint, it is presumably associated with backlash. The proof will be given through the use of a Matlab/Simulink model of the system in Chapter 4.

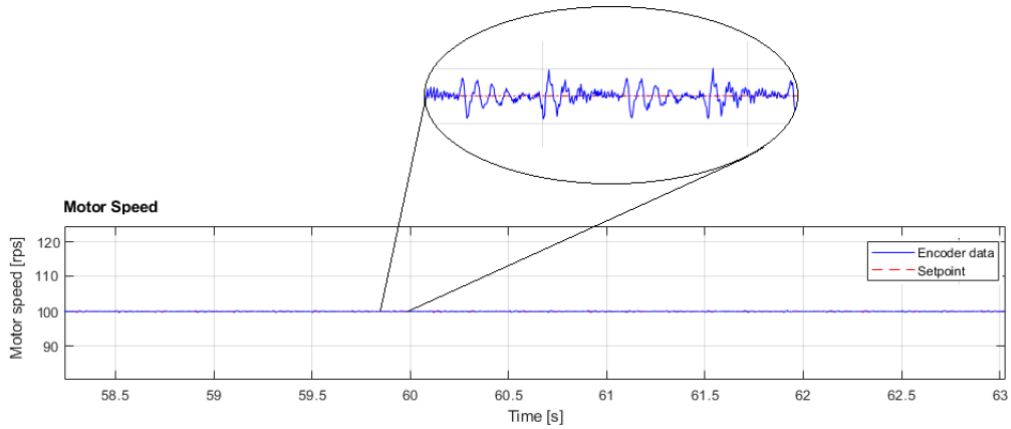


Fig. 3.2 The motor speed signal and the superimposed backlash disturbance

Since the disturbance has a well-defined and recognizable shape it can be considered as the *"signature of the backlash"*. Every time this signature is detected in the speed signal of the motor then an excessive amount of backlash is present in the joint. Moreover, an estimate of the value of backlash in the joint can be obtained by looking at the characteristics of the signature (ref to Chapter5).

As will be seen later in this thesis, if a mathematical model of the disturbance signature is developed then it can be used as a reference pattern in a fault detection system. The system compares the pattern with the signals coming from the robot encoder and provides an estimate of the backlash in the joint (ref Chapter 5).

3.3 The backlash signature

The backlash disturbance has a very characteristic shape that depends on the small impacts that occur between the teeth of the gears and that are generated each time the backlash gap opens and then closes again. This is why it has the characteristics of a percussion phenomenon: an impulse and a consequent oscillation that decays over time. The disturbance takes the shape of a sequence of many of these damped oscillations and appears as superimposed onto the constant motor speed signal, see Figure 3.2. A single backlash-induced oscillation can be represented with the mathematical model:

$$d_b(t) = \begin{cases} 0 & t < t_1 \\ A\, e^{-(t-t_1)\tau} \sin(\omega t) & t_1 \leq t \leq t_2 \\ 0 & t > t_2 \end{cases} \tag{3.1}$$

A is the amplitude, τ is a damping factor, t_1 and t_2 are the oscillation's beginning and ending times, respectively. Many of these oscillation are repeated in the disturbance sequence with alternated sign. The amplitude is the same for all them, small variations are due to signal noise. The alternated sign depend on the test conditions used. The test is performed by running the motor in the same direction for a long time, at constant speed. In each load turn two impacts are generated: one at the start of the load movement's descending phase and the other at the start of its ascending phase. In both of these times the backlash gap opens and then closes again, so two opposite oscillations are generated for every load turn.

The two oscillations contained in one load turn can then be represented by the time limited function $f(t)$:

$$f(t) = d_b(t) - d_b(t - t_d) \tag{3.2}$$

where t_d is the point in time where the second oscillation starts.

Given that multiple copies of this function are visible during the observation window, the complete model for the disturbance $h(t)$ is taken as the sum of n time-shifted replica of $f(t)$:

$$h(t) = \sum_{i=1}^{n} f(t - (i-1)t_f) \tag{3.3}$$

where t_f is the time shift that corresponds to a full load rotation (2π rad).

A further trimming of the model was realized by defining the reference starting point of the disturbance, t_0, and redefining all significant instants of the disturbance with respect to t_0,

$$t_1 = t_0 + T_1,\ \ t_2 = t_0 + T_2,\ \ t_d = t_0 + T_d,\ \ t_f = t_0 + T_f \tag{3.4}$$

Equation 3.1 then changes to

$$d_b(t-t_0) = \begin{cases} 0 & t < t_0 + T_1 \\ A\, e^{-(t-(t_0+T_1))\tau} \sin \omega(t-t_0) & t_0 + T_1 \leq t \leq t_0 + T_2 \\ 0 & t > t_0 + T_2 \end{cases} \tag{3.5}$$

while $f(t)$ becomes

$$f(t-t_0) = d_b(t-t_0) - d_b(t-(t_0+T_d)). \tag{3.6}$$

As a consequence, the model for the disturbance is described by 7 parameters

$$\begin{aligned} &h(t, A, t_0, \tau, \omega, T_1, T_2, T_d, T_f, n) = \\ &\textstyle\sum_{i=1}^{n} f(t, A, t_0 + (i-1)T_f, \tau, \omega, T_1, T_2, T_d). \end{aligned} \tag{3.7}$$

Equation (3.7) is taken as the *signature of the backlash* in the robotic transmission.

The mathematical model and its parameters are shown in Figure 3.3. In the example, $n = 2$ replica are considered, so a total of 4 damped oscillations is represented. The starting point of the disturbance t_0, and the meaningful time intervals T_1, T_2, T_d, T_f are reported. The first two dampened oscillations, $d_b(t)$ and $d_b(t-T_d)$, are related to a first turn of the load $(n = 1)$. The following two oscillations are related to the second $(n = 2)$ turn of the load.

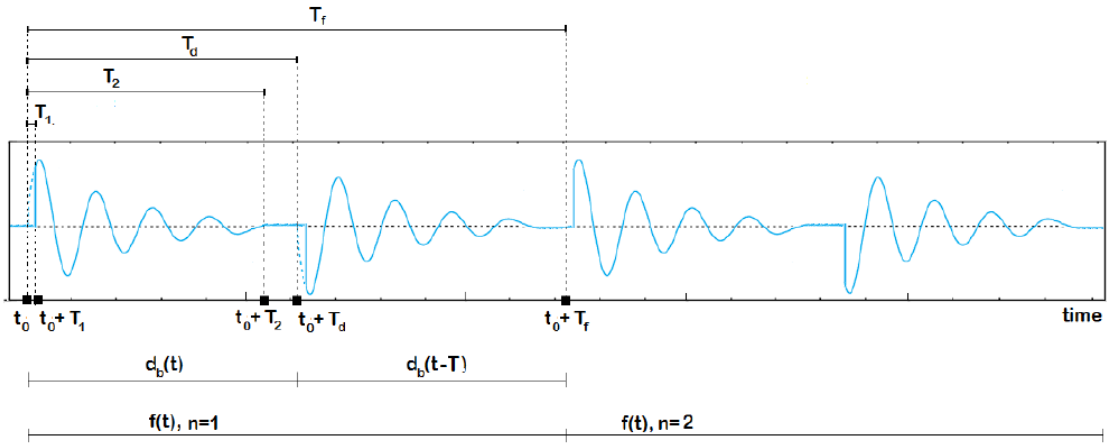


Fig. 3.3 A graphical representation of the function $h(t)$ and its parameters.

.

Multiple replicas of the $f(t)$ function were considered in the model in order to prevent identification errors caused by noise. With this choice a *de facto* identification of the mean value of the parameters comes. In all the experiments performed the value for number of disturbance repetition was set to $n = 12$, while T_d was set as $T_d = T_f/2$ because its known from the test conditions that there are two impacts in a full load rotation.

Since in a joint the backlash increases as the system's working hours increase, presumably the backlash signature will also change accordingly. To investigate how the signature changes as the backlash increases a Matlab/Simulink model of the system was developed. Thought a model-based analysis the association between the joint's backlash quantity and the backlash signature's features was examined.

3.4 Data collection

The test bench was also used to derive experimental data to be used for the validation of the identified backlash measurement method. The system was powered with a standard COMAU robot controller, and left running continuously under maximum stress conditions for 8 months to induce wear and the increase of backlash. The stress cycle was interrupted every three weeks to perform a specific test cycle and gather information on the status of the backlash by collecting encoder recordings. This activity allowed the creation of a dataset of machine-taken acquisitions. The dataset was then used to validate the analysis pipeline developed during the research work and described in Chapter 5.

Chapter 4

Physical-based model of the system

Experiments on the test bench showed that by selecting a proper excitation for a joint affected by backlash, a disturbance with a well-recognisable shape can be detected on the motor speed signal. It is now intended to demonstrate that this disturbance is exclusively attributable to backlash, and not to other possible mechanical phenomena originating within the coupling. To this intent, a simulated model of the robotic joint was developed. The model considers the main phenomena acting on the joint as friction, inertia, gravity, joint elasticity and also backlash. The goal is to demonstrate that by applying identical excitation to both the model and the real system, the resulting signals exhibit a comparable disturbance. This disturbance aligns with the one observed in the signal recorded from the real system, and disappears when the backlash is eliminated.

The simulated model serves also additional purposes as:

- confirming the generality of the obtained results across various types of couplings, such as gears, belts, harmonic drives, and more,
- investigate the changes in the backlash signature as the backlash increases,
- easily generate simulated data.

4.1 Model of the robot joint

A Matlab/Smulink model of a robotic joint was developed to validate the insights obtained from the test-bench and to gain a deeper knowledge about the backlash phenomenon and its effects on the system dynamics.

The main blocks of the model and their interconnections are reported in Figure 4.1. The system comprises four main blocks: a motor with an encoder, a transmission, a load, and the control loops block for the motor's speed and position. The motor position is read using the encoder and fed back into the control loop. The motor speed is obtained by taking the derivative of the encoder's position. Just like in the real system, there are no sensors on the load or at the output of the transmission. The backlash phenomenon is modeled within the transmission block, while the effect of external forces, such as gravity, are modeled within the block related to the load through the equations of the load dynamics. More details on how the motor and load have been modeled are provided in Figure 4.2 and Figure 4.3. The details regarding the model used for the transmission affected by backlash are described in detail in the following paragraph.

4.1.1 The joint transmission model

The model of backlash is added to the transmission system modeled with a typical linear dynamic model with two masses and elastic coupling. The first mass represents the motor and its moment of inertia J_m, the second mass is the load with its moment of inertia J_l. The system and its components are reported in Figure 4.4, the parameters used in the model are explained in Table 4.1. Motor and load are coupled through an elastic shaft that is considered mass-free and is described with its torsional spring, with stiffness K_s, and dumping D_s. The transmission ratio is N. The backlash angular gap is 2δ, it models the total effect of all the loosely connected elements in the transmission.

The movement of motor is transferred to the load when the backlash gap is closed, and that is when all the mating elements of the transmission are in contact. In this case Equation (4.1) shows that the load torque τ_l is proportional to the angle difference $\Delta\theta$ and the speed difference $\Delta\omega$. If the motion is reversed the backlash

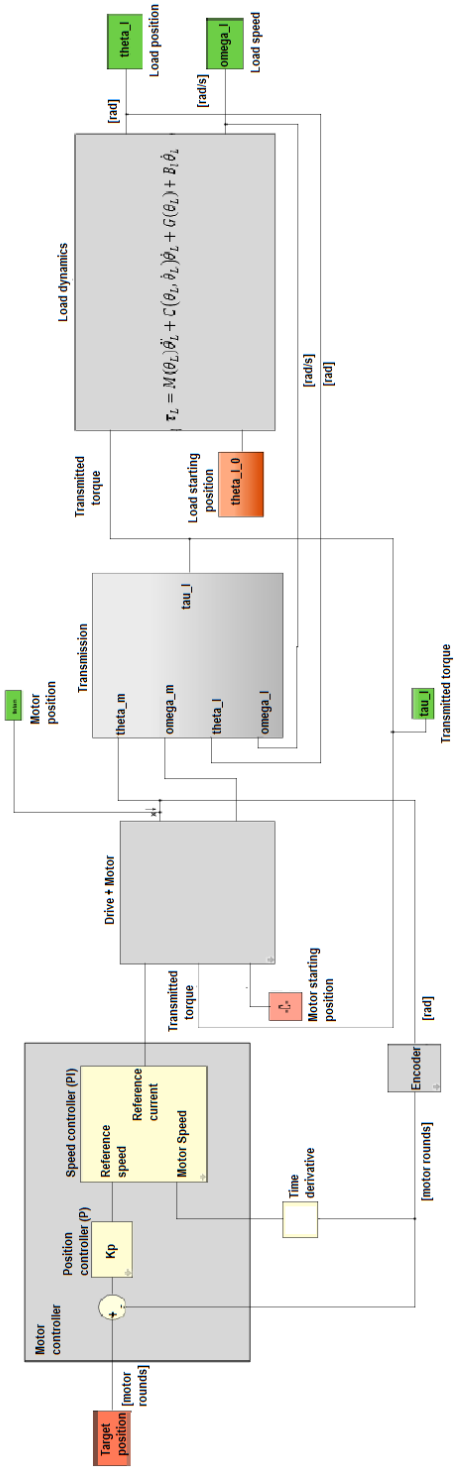


Fig. 4.1 Matlab/Simulink model of a single robotic joint

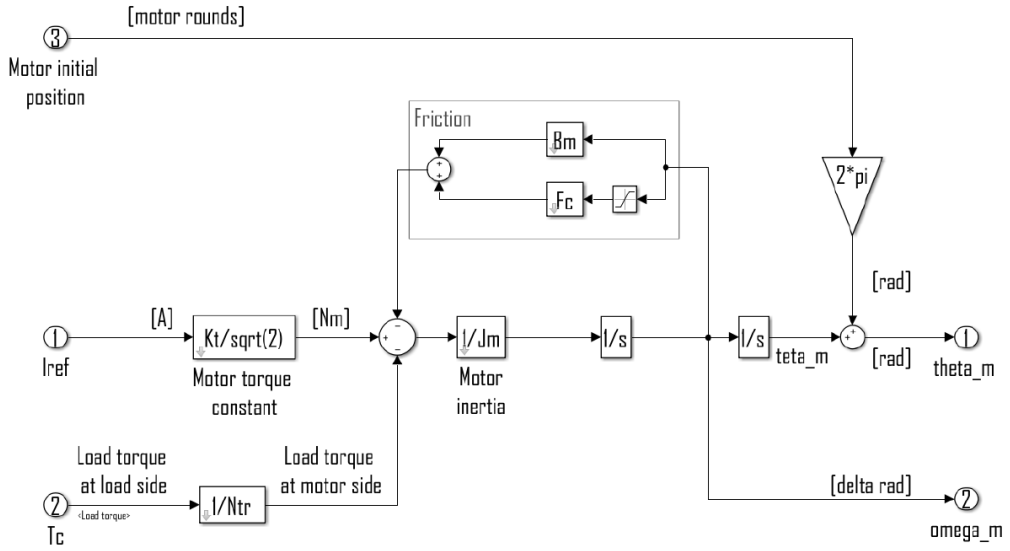


Fig. 4.2 Matlab/Simulink model of the motor

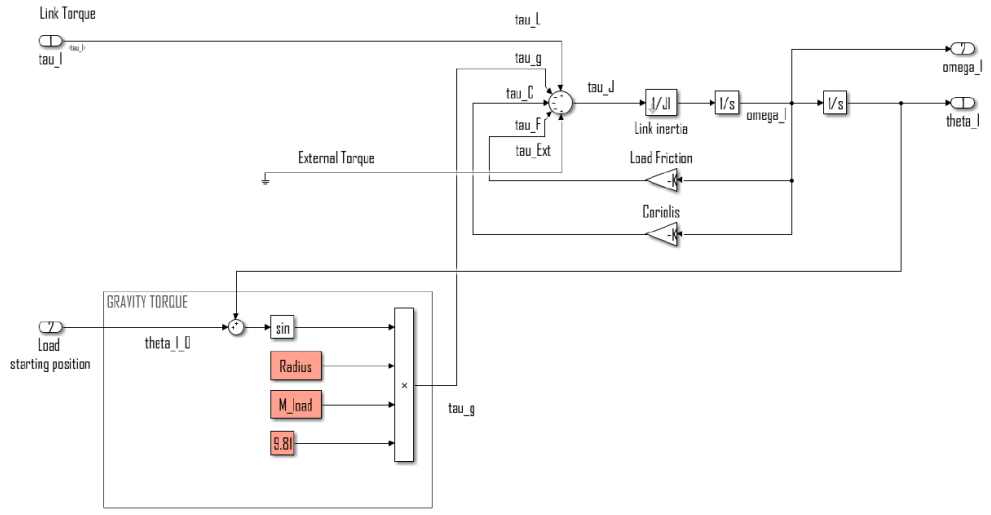


Fig. 4.3 Matlab/Simulink model for the link

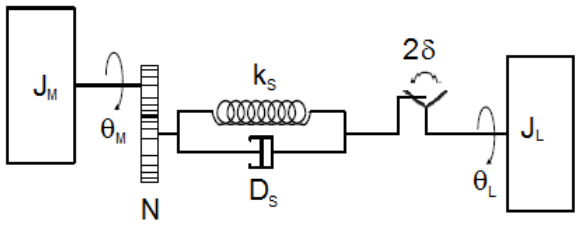


Fig. 4.4 Two rotary inertia system with elastic coupling and backlash.

Symbol	Description	Units
θ_m, θ_l	Motor/Load angular position	rad
ω_m, ω_l	Motor/Load angular velocity	rad
τ_m, τ_l	Motor/Load torque	Nm
J_m, J_l	Motor/Load inertia	Kgm^2
K_s	Shaft stiffness	Nm/rad
D_s	Shaft damping coefficient	Nm s/rad
δ	Backlash angle	rad
N	Gear ratio	-

Table 4.1 System parameters.

gap opens and the motor loses contact with the load, the value of resulting torque transmitted to the load is then zero, Equation (4.1).

$$\Delta\theta = \frac{1}{N}\theta_m - \theta_l \tag{4.1}$$

$$\Delta\omega = \frac{1}{N}\omega_m - \omega_l \tag{4.2}$$

In a transmission without backlash the torque transmitted to the load is

$$\tau_l = K_s\Delta\theta + D_s\Delta\omega \tag{4.3}$$

while, in presence of backlash two different cases have to be considered

$$\tau_l = \begin{cases} K_s(\Delta\theta - \delta\,\mathrm{sign}(\Delta\theta)) + D_s\Delta\omega & |\Delta\theta| > \delta \\ 0 & |\Delta\theta| \leq \delta \end{cases} \tag{4.4}$$

This is a simple representation of the backlash effect, in this research work a more sophisticated model was chosen instead to account for it. The details of the model used are described below.

4.1.2 The backlash model

Several backlash models have been proposed in literature, the most commonly used are the classical *dead zone model* and *hysteresis model*, reported in the work of [44] and [43] respectively. The backlash model used in this work is a *modified dead zone model* where backlash is modeled in terms of a variable stiffness. This model was selected because, unlike others dead zone models, it does not have discontinuities. In simulation discontinuities can cause system instability, so continuity is a very helpful feature to have a controlled simulation. The model used is described in the work [45, 46]. It relies on a variable shaft stiffness that suddenly assumes the zero value when the backlash gap opens. The *arctan* function is used to provide sudden but continuous variation of the stiffness. An α factor is used to adjust the arctan slope. Here is the formula for the variable stiffness $K_{BL}(\Delta\theta)$

$$K_{BL}(\Delta\theta,\delta) = \frac{K_s}{\pi}[\pi + \arctan(\alpha(\Delta\theta-\delta)) - \arctan(\alpha(\Delta\theta+\delta))] \tag{4.5}$$

that makes the stiffness assume the constant value K_s outside the backlash gap ($|\Delta\theta| > \delta$), and the 0 value inside the gap ($|\Delta\theta| \leq \delta$), see Figure 4.5.

Hence, by considering the variable stiffness formula, Equation (4.4) becomes

$$\tau_l = [\Delta\theta - \delta \cdot sign(\Delta\theta) + \frac{D_s}{K_s}\Delta\omega] \cdot K_{BL}(\Delta\theta,\delta) \tag{4.6}$$

assuming zero value within the backlash gap, and being proportional to $\Delta\theta$ and $\Delta\omega$ outside the backlash gap.

4.2 Model setup and simulation results

The Simulink model's parameters were determined through the processing of experimental data gathered on the test bench. Model parameters identification was

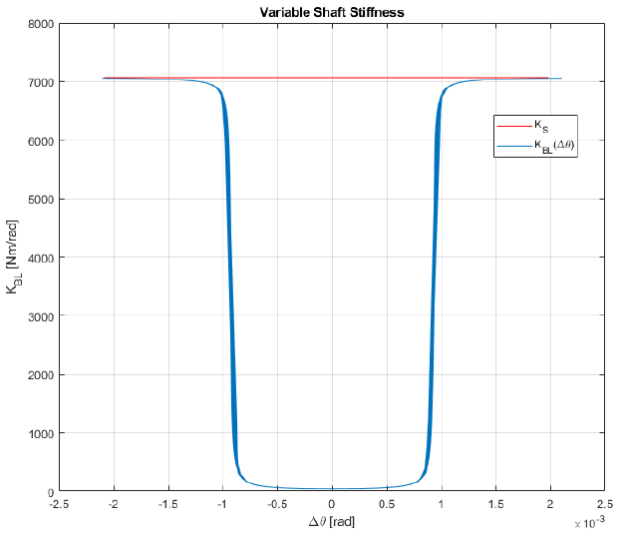


Fig. 4.5 The variable stiffness value as the displacement between motor and load changes.

performed to bring the model as close as possible to the real system. The simulator's final validation involved a comparison between the current and encoder signals obtained from Simulink and those obtained from the test bench. Figure 4.6 depicts the comparison, with signals from the Simulink model shown in black and signals measured on the test bench presented in light blue. Signals as motor position, motor speed, and motor current are represented. The impact of backlash disturbance is evident in both simulated and actual data, exhibiting identical characteristics. The plots distinctly demonstrate the simulator's ability to reproduce the backlash behavior in the joint.

4.3 Data availability problem and simulated dataset

Another use of the simulator was to create a simulated dataset to be used in the development of the predictive maintenance tool.

It is not easy to obtain a consistent amount of data about the evolution of a backlash fault in a robotic system. Backlash is a phenomenon that evolves very slowly over time; it usually takes years before a new robot starts to show wear-related backlash. It also takes a long time to go from fault to failure. Moreover, failure events are infrequent in production systems due to the expensive and hazardous con-

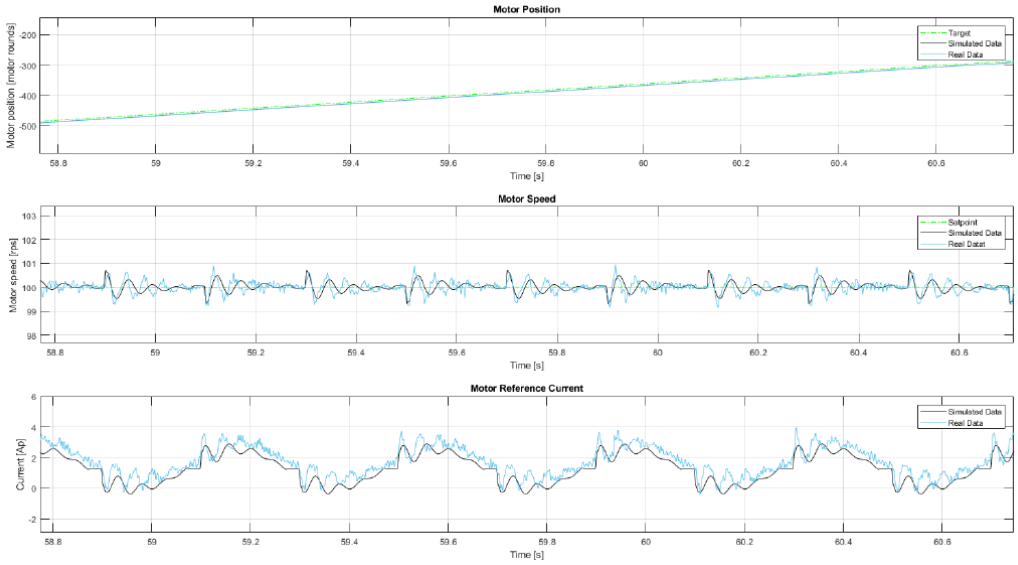


Fig. 4.6 Comparison between simulated (black) and real (blue) signals.

sequences of machinery operating under faulty conditions, along with the potential for a prolonged degradation process before a significant failure occurs [6]. The idea was then to exploit the flexibility offered by simulation to quickly generate the data.

4.4 Dynamics of backlash signature

It was decided to use the simulator to mimic the progressive aging of the system. The backlash in a joint increases with wear over time, and the progression of this phenomenon is very slow. To study the evolution of the backlash signature with varying backlash values, it would have been necessary to observe the test bench for a long time to obtain data related to the progression of the phenomenon. For this purpose, the simulator proved to be very useful, allowing the generation of a large amount of simulated data in a short time. By taking advantage of the simulation's flexibility it was possible to examine various backlash situations and get a full understanding about the way the backlash signature shape changes as backlash grows. This made it easier to determine the relationship between the backlash's value and the disturbance's intensity.

Many simulations were performed by keeping the test condition as constant: the motor was run at a constant speed of 100 rps and a payload of 5 kg were used. The

only parameter that has been varied was backlash value. Ten different increasing values were considered spanning the interval $[0.0003; 0.0021]$rad with a step of 0.0002rad. The maximum and minimum values were selected as significant for the tested system. Data from the motor encoder were gathered at each simulation, so a final batch of ten dataset was collected.

The numerous datasets generated through simulation, each associated with different backlash values, were employed to explore the correlation between backlash variation and changes in the disturbance signal. It was observed that the amplitude of the disturbance oscillation is linked to the backlash value in the joint, consistently increasing as the backlash gap widens. This behavior is evident in Figure 4.7, where disturbance signals from the datasets are plotted together, illustrating the effects of a progressively expanding backlash. Each signal represents the vibration disturbance observable in the motor speed signal for a distinct backlash value. The vibration is triggered by small impacts within the mechanical transmission resulting from the associated backlash. The set of 10 signals corresponds to the 10 different backlash values considered in the simulations.

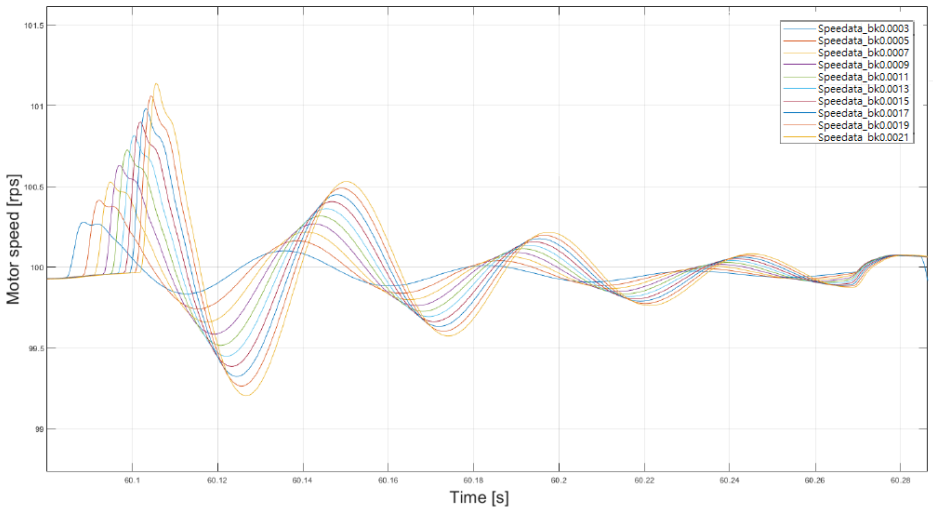


Fig. 4.7 Disturbance pattern appearance as backlash gap expands.

The parameter A in the disturbance model, being connected to the disturbance amplitude, was utilized as a reference for estimating backlash and treated as the monitored parameter for fault growth. Monitoring the value of A over time provides information about changes in the system's backlash.

For an absolute backlash measurement, calibration is necessary to associate the value of A with the specific backlash value in the system. This calibration step is crucial as the amplitude of A depends on the mechanical characteristics of the tested system. The mapping $\delta \mapsto A$ between the backlash value in the joint and the corresponding value of the A parameter is presented in Figure 4.8 for the simulated system. This mapping was established by plotting the maximum amplitude of signals in Figure 4.7 against the corresponding backlash values and fitting the points with a cubic polynomial function. Each point represents a unique simulation, with the δ coordinate indicating the specific backlash value used in the simulation and the A coordinate denoting the amplitude of the corresponding disturbance observed on the encoder signal. The 10 pairs $(A;\delta)$ correspond to the 10 datasets considered in the analysis. The interpolating cubic function, $\delta = f(A)$, is represented by the continuous dotted line. The resulting function was

$$\delta = f(A) = 1 \times 10^{-6} \cdot (-0.0018 \cdot A^3 + 0.4323 \cdot A^2 + 9.2524 \cdot A + 34.0313) \tag{4.7}$$

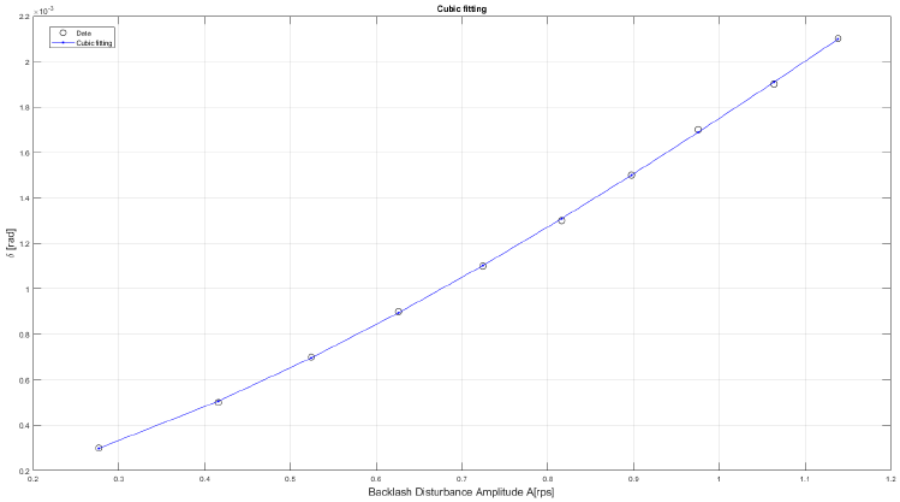


Fig. 4.8 The plot shows the association between the joint backlash (δ) and the amplitude of the speed disturbance (A) within the simulated system.

This function will be a key element of the PdM tool because it encompasses the relationship between the actual output fault indicator and the value for the backlash estimate. This relationship, in reverse form, can be used to obtain an estimate of the monitored system's state (i.e. the backlash δ) based on the value of the observed fault indicator (A). By collecting the output fault indicator change, the state of the

backlash in the system can be reconstructed so as to realize the rapid prediction and warning.

Chapter 5

Proposed approach

5.1 Backlash faults in robots

Backlash is not inherently harmful for a mechanical system, indeed a small amount of backlash is positive as it facilitates a smooth operation of the transmission (Ref to Chapter 1). The problem with backlash arises when the backlash value defined at the design stage does not remain the same over time, but increases due to the wear that occurs as the robot's operation progresses over time. The engaging teeth of the gears wear down as they slide against each other and the process results in an accumulation of excess space within the transmission, commonly known as the *backlash fault"*.

A *fault* is defined as alteration in the normal operation of the system that causes a deterioration in performance [9]. It can thus be understood as a malfunction of the system. A fault can be tolerated at its early stages and later became unacceptable as it progresses over time. Whereas, a *"failure"*, is defined as a complete breakdown of a system/component, and must be addressed as soon as possible.

The goal of this research work is to identify a method to recognize and measure backlash problems in a robot joint in order to plan maintenance optimally and avoid incurring in the machine failure. The final objective is to develop a monitoring system that fulfills three main tasks:

- detects the presence of an anomaly in the operation of the robotic joint (*fault detection*).
- connects this anomaly with the particular issue of backlash

- compute an estimate of the size of the backlash.

Fault detection strategies can be of different types. The selection of a specific strategy depends on the nature of the system, the available data, and the types of faults expected. Often, a combination of these strategies is employed for a comprehensive fault detection and diagnostics.

5.2 Fault detection method

The method presented in this thesis combines a model-based fault detection method with a Evolution Strategy (ES) algorithm and uses this integrated technique for diagnosing the fault of backlash in robotic manipulators.

The *model-based approach* uses a mathematical model of the system to derive a reference signal to be used to measure the difference between an healthy behavior of the system and a faulty one. Then the measured signal is compared with a reference pattern, a *"fault signature"*, and a parameter is derived from the comparison and used as *"fault indicator"*. The fault indicator will have a double value: it signals the presence of the fault, its value is zero when the system is operating normally while diverges from zero when a fault is occurring; and it indicates the level of fault severity, the larger the fault indicator the greater the fault severity. The comparison between the measured signal and reference pattern is performed through an Evolution Strategy (ES) algorithm, a search technique inspired by principles of evolution and natural selection. Figure 6.3 schematises the architecture of the functional blocks at the basis of the proposed method.

The choice of a model-based approach was dictated by:

- the lack of data on faults
- the lack of a second sensor for the measurements, which prevented the possibility of a direct measurement of the backlash
- the installation position of the sensor, that being far from the source of the phenomenon to be measured opens up to the presence of disturbances and interference of on the signal.

The use of a model compensates for the lack of knowledge arising from the circumstances mentioned above.

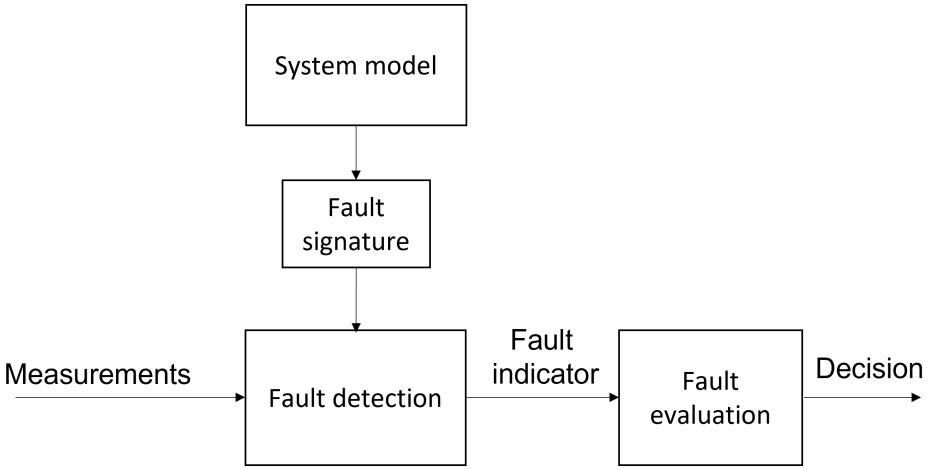


Fig. 5.1 Model-based fault detection approach

5.3 Estimation toolchain

The idea was then to perform specific test movements of the robotic joint, record the signal from the motor encoder, and get an estimate of the backlash affecting the robotic joint by looking at the characteristics of a peculiar disturbance signal that can be found in the motor speed signal. In particular, by measuring the disturbance amplitude A and then using the function $\delta(A)$, an estimate of the backlash in the joint can be obtained. To this intent, the speed signal of the motor is collected and compared with the backlash signature, if a match is found then backlash is present in the joint transmission and the value of backlash can be obtained by looking at the characteristic of the identified signature. The greater the amplitude of the signature, the larger the backlash in the joint.

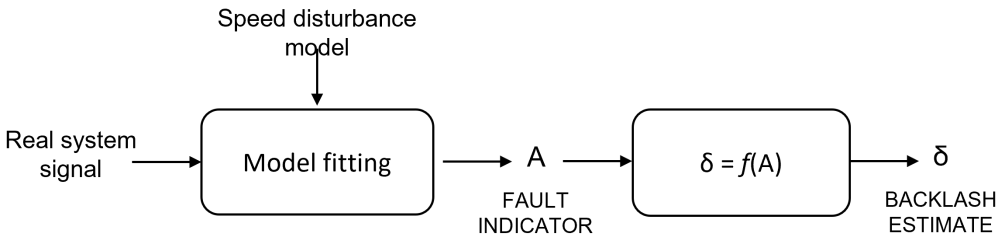


Fig. 5.2 Basic architecture of the proposed tool-chain.

5.3.1 The fitting problem

The goal is to fit the backlash signature on the motor speed signal. The fitting is performed by solving an optimization problem where the cost function to be minimized is RMSE error between the signature and the signal from robot.

It was mentioned that to estimate the backlash δ, the initial step involves calculating A. The value of A, along with other parameters in the model defined by Equation (3.7), is determined through the optimization process. The optimization problem aims to identify the optimal combination of parameter values which allows the fitting between the disturbance model $h(t)$ and the encoder signal. Due to the non-convex nature of the problem, traditional optimization methods are not directly applicable. Therefore, a stochastic optimization meta-heuristic is employed. The formulation of the problem and the characteristics of the algorithm will be elaborated below. In the initial phase, the difference signal $r(t)$, representing the variance between the measured speed and the theoretical speed (calculated under no backlash conditions), is computed. The resulting signal is

$$r(t) = v(t) - v_t \tag{5.1}$$

where v_t represents the prescribed motor speed as specified by the test conditions. Subsequently, the residual signal is examined to identify the backlash disturbance pattern $h(t)$. This identification is based on minimizing the error between $r(t)$ signal and the model signal $h(t)$. Specifically, the employed cost function is the Root Mean Square Error (RMSE) calculated between these two signals.

$$RMSE = \sqrt{\frac{\Sigma_{i=1}^{N}\Big(r(t) - h(t, A, t_0, \tau, \omega, T_1, T_2, T_d, T_f)\Big)^2}{N}}.$$

The parameters to be determined are

$$X = [A, t_0, \tau, \omega, T_1, T_2, T_d, T_f]$$

The algorithm used to implement the fitting is a stochastic optimization algorithm based on evolutionary strategies known as CMA-ES. The description of this algorithm, along with the results obtained from the fitting, is reported in the Chapter 7.

Chapter 6

Computational Intelligence

Based on the discussions in the previous chapters, the issue of measuring backlash in a robotic joint can be traced back to an optimization problem, more specifically a non-convex, multi-variable optimisation problem. Managing this kind of problems can be challenging, but various techniques and approaches can be employed.

Among the many strategies that are suitable for the task are Swarm Intelligence algorithms together with Evolutionary Strategies, both belonging to the field of *Computational Intelligence*. A field of study within artificial intelligence (AI) and computer science that involves the development and application of computational models to mimic and emulate intelligent behavior. To choose the most appropriate algorithm for addressing the specific optimization problem, five distinct algorithms were configured and assessed. A description of their working principle is given below, while the analysis and comparison of their results on the backlash detection problem is given Chapter 7.

6.1 Nature inspired intelligence

"Look deep into nature, and then you will understand everything better" — Albert Einstein.

From the earliest times, nature has been serving as an inspiration for humans. A plethora of brilliant inventions happened just through careful observation of nature. For example Velcro was inspired by the way burrs attach themselves to dog's fur, or

the ultra-fast Shinkansen Train's nose that was inspired by the aerodynamic shape of the beak of Kingfisher bird, Figure 6.1.

Fig. 6.1 Bullet train nose design compared to a Kingfisher

As a result of millions of years of successive improvement through trial and error, nature learned a solution for everything. The approach to human innovation, via emulating nature, is called *Biomimetic* and is also applied to Computer Science. Models, systems, and strategies of nature can be modeled within algorithms to be used to solve a broad variety of problems. The sub-branch of Artificial Intelligence, that contains the many nature-inspired computational methodologies is called Computational Intelligence.

Computational Intelligence (CI) encompasses an interdisciplinary field of study and problem-solving techniques. It takes inspiration from the principles of natural intelligence to create algorithms and computational models capable of undertaking tasks that have traditionally been associated with human intelligence. CI involves a wide range of methodologies, including but not limited to *artificial neural networks, fuzzy logic, evolutionary algorithms, swarm intelligence, and expert systems.*

The key characteristics of computational intelligence are the ability to adapt to changing environments, learn from experience, and handle complex and uncertain information. CI techniques are particularly suited for solving problems that may be too complex or ill-defined for traditional rule-based programming. Computational intelligence seeks to replicate aspects of human cognition and problem-solving capabilities in machines, allowing them to autonomously learn and make decisions in diverse and dynamic contexts. The application of CI spans various fields, including data analysis, optimization, pattern recognition, robotics, and decision support systems.

6.2 Meta-heuristics

Meta-heuristic algorithms (MHs) form a subset of Computational Intelligence, they refer to high-level strategies or frameworks designed to solve complex optimization problems. Meta-heuristics are viewed as a more advanced form of heuristics, operating at a higher level. Unlike simple trial-and-error methods, meta-heuristic algorithms are designed to learn from past solutions, prioritize better moves, select optimal solutions, and construct sophisticated search strategies. This distinguishes them from heuristic algorithms, making meta-heuristics more effective and efficient than random, exploratory approaches. The ability of meta-heuristics to leverage learned knowledge, bias toward improved solutions, and employ advanced search techniques contributes to their superiority over traditional heuristics and random search methods. One of the reasons why they are so appreciated is that meta-heuristic algorithms excel in global search strategies. Unlike traditional optimization methods that may get stuck in local optima, meta-heuristics efficiently explore and exploit the entire solution space. This attribute is crucial for tackling optimization problems with intricate landscapes, where finding the global optimum is challenging. The ability to conduct a comprehensive search ensures that meta-heuristics are adept at uncovering optimal solutions even in highly complex scenarios. Another reason for the use of meta-heuristics lies in their flexibility. These algorithms are designed to be versatile and can be applied to a wide variety of optimization problems without requiring problem-specific information. This adaptability is particularly advantageous in situations where the problem structure is dynamic, and the algorithm needs to adjust its strategy accordingly. The flexibility of meta-heuristics allows them to tackle diverse problem types, ranging from continuous optimization to combinatorial problems involving discrete decision variables. Moreover, meta-heuristics are well-suited for non-convex optimization problems. Traditional optimization methods may struggle with objective functions that are non-convex or discontinuous. Meta-heuristics, on the other hand, are robust in handling such complexities. By employing innovative search and exploration strategies, meta-heuristics can navigate through challenging solution spaces, overcoming the limitations of traditional approaches. The inherent parallelism of meta-heuristic algorithms further contributes to their effectiveness. These algorithms often explore multiple solutions concurrently, leading to faster convergence and efficient utilization of computational resources.

A readily accessible collection that encompasses traditional and modern sets of meta-heuristic algorithms is provided in Figure 6.2.

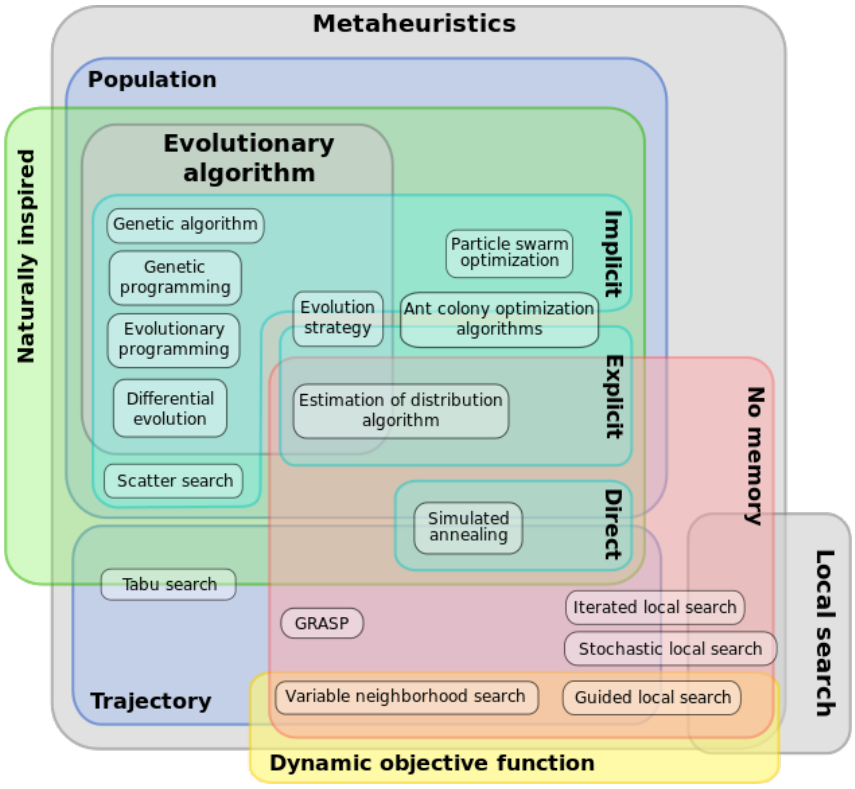


Fig. 6.2 Taxonomy of the most common meta-heuristic algorithms [1].

6.2.1 CMA-ES

The initial implementation experiments of the method proposed in Chapter 5 for estimating backlash were conducted using the CMA-ES algorithm, a particular tool for numerical optimization belonging to the class of Evolution Strategy (ES). Evolution strategies are optimization algorithms inspired by the process of natural selection and evolution. They involve maintaining a population of candidate solutions, subjecting them to variation and selection operations, and iteratively improving them over successive generations to find optimal or near-optimal solutions to optimization problems.

[1] Johann "nojhan" Dréo, Caner Candan, CC BY-SA 3.0 <http://creativecommons.org/licenses/by-sa/3.0/>, via Wikimedia Commons.

CMA-ES (Covariance Matrix Adaptation Evolution Strategy) is a well-recognized evolutionary strategy known for its strong performance in solving black-box optimization problems. Initially introduced by Hansen in 1995 [21] was further refined in the subsequent works [19, 20]. It is an ES that leverages a *covariance matrix*, which is closely associated with the inverse Hessian in convex-quadratic functions. The method demonstrates its strength in addressing challenging problems characterized by non-linearity, non-convexity, and non-separability, particularly in cases with moderate to high dimensionality (i.e., n in the range [10, 100]). Notably, the CMA-ES distinguishes itself by not relying on gradients, whether exact or approximated, and operates successfully in scenarios where derivative-based methods such as *Broyden-Fletcher-Goldfarb-Shanno* [14] or *conjugate gradient* may encounter difficulties, such as in the presence of discontinuities, sharp bends, noise, or local optima.

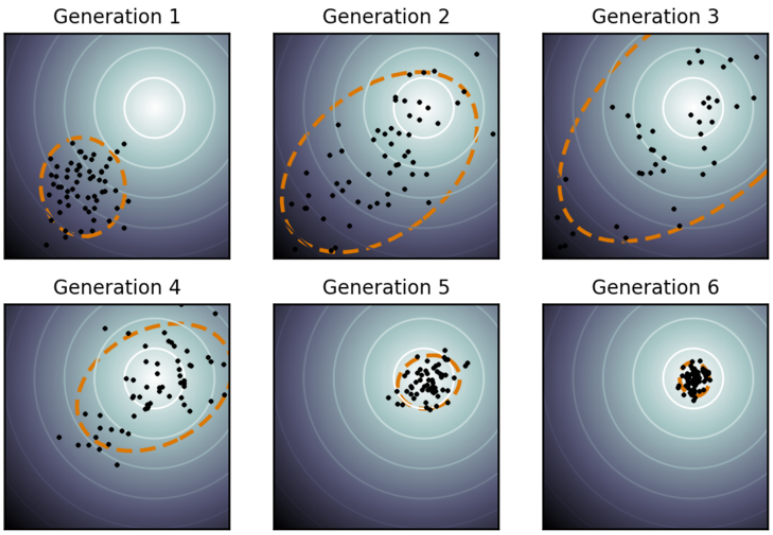


Fig. 6.3 CMA-ES generation evolution on a two dimensional problem[2].

In the context of CMA-ES, the term *generations* is employed to refer to iteration steps, a nomenclature inspired by its biological foundations. For any generic algorithm parameter y, its value during generation g is denoted as $y^{(g)}$. The mean vector $\mathbf{m}^{(g)} \in \mathbb{R}^n$ represents the currently favored and most promising solution. The *step size* $\sigma^{(g)}$, a positive real number in $\mathbb{R}_+$, serves to control the length of the steps taken during the iteration. Simultaneously, the *covariance matrix* $\mathbf{C}^{(g)} \in \mathbb{R}^{n \times n}$

[2]Sentewolf, Public domain, via Wikimedia Commons

plays a crucial role in determining the shape of the distribution ellipsoid within the search space. The primary objective of $\mathbf{C}^{(g)}$ is, in broad terms, to align the search distribution with the contour lines of the objective function f that is being minimized. It strives to adapt the distribution to fit the characteristics of the function. The initial setting of the covariance matrix is denoted as $\mathbf{C}^{(0)}$, which is initialized to the identity matrix I, $\mathbf{C}^{(0)} = \mathbf{I}$.

During each iteration or generation g, a set of λ new solutions, denoted as $\mathbf{x}_i^{(g+1)} \in \mathbb{R}^n$, is generated. This generation process involves sampling from a multivariate normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{C})$ with a mean vector of $\mathbf{0}$ (refer to equation 6.1).

$$\mathbf{x}_k^{(g+1)} \sim \mathcal{N}\left(\mathbf{m}^{(g)}, \left(\sigma^{(g)}\right)^2 \mathbf{C}^{(g)}\right), k = 1, \ldots, \lambda \tag{6.1}$$

The symbol $\cdot \sim \cdot$ indicates that the distribution on the left side is the same as the one on the right side.

After the sampling phase, the generated solutions undergo evaluation and ranking. $\mathbf{x}_{i:\lambda}$ denotes the i^{th} ranked solution point, such that $f(\mathbf{x}_{1:\lambda}) \leq \ldots \leq f(\mathbf{x}_{\lambda:\lambda})$. The top μ solutions from the set of λ are chosen to guide the subsequent generation $g+1$. Initially, the update process involves modifying the distribution mean, as illustrated in Equation (6.2).

$$\mathbf{m}^{(g+1)} = \sum_{i=1}^{\mu} w_i \mathbf{x}_i^{(g)}, w_1 \geq \ldots \geq w_\mu > 0, \sum_{i=1}^{\mu} w_i = 1 \tag{6.2}$$

To refine its internal parameters, the CMA-ES monitors the *evolution paths*, which are sequences of consecutive normalized steps observed across multiple generations.

The *conjugate* evolution path is $\mathbf{p}_\sigma^{(g)} \in \mathbb{R}^n$, and $\mathbf{p}_\sigma^{(0)} = \mathbf{0}$. To normalize paths the $\sqrt{2}\frac{\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{n}{2}\right)} \approx \sqrt{n} + \mathcal{O}\left(\frac{1}{n}\right)$ expectation of the Euclidean norm of a $\mathcal{N}(\mathbf{0}, \mathbf{I})$ distributed random vector is used. While the *variance effective selection mass* is usually denoted as $\mu_{\text{eff}} = \left(\sum_{1=1}^{\mu} w_i^2\right)^{-1}$. By having $c_\sigma < 1$ as the learning rate for cumulation for the rank-one update of the covariance matrix, and $d_\sigma \approx 1$ as the damping parameter for step size update, the paths can be updated through the equations:

$$\mathbf{p}_{\sigma}^{(g+1)} = (1-c_{\sigma})\mathbf{p}_{\sigma}^{(g)} + \sqrt{c_{\sigma}(2-c_{\sigma})\mu_{\text{eff}}}\mathbf{C}^{(g)-\frac{1}{2}}\frac{\mathbf{m}^{(g+1)}-\mathbf{m}^{(g)}}{\sigma^{(g)}} \tag{6.3}$$

$$\sigma^{(g+1)} = \sigma^{(g)}\exp\left(\frac{c_{\sigma}}{d_{\sigma}}\left(\frac{\left\|\mathbf{p}_{\sigma}^{(g+1)}\right\|}{\sqrt{2}\frac{\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{n}{2}\right)}}-1\right)\right) \tag{6.4}$$

The evolution path is $\mathbf{p}_c^{(g)} \in \mathbb{R}^n$, and $\mathbf{p}_c^{(0)} = \mathbf{0}$. By having $c_c < 1$ as the learning rate for cumulation for the rank-one update of the covariance matrix, μ_{cov} as the parameter for weighting between rank-one and rank-μ update, and $c_{\text{cov}} \leq 1$ as the learning rate for the covariance matrix update then the covariance matrix $\mathbf{C}$ is updated through the two equations:

$$\mathbf{p}_c^{(g+1)} = (1-c_c)\mathbf{p}_c^{(g)} + \sqrt{c_c(2-c_c)\mu_{\text{eff}}}\frac{\mathbf{m}^{(g+1)}-\mathbf{m}^{(g)}}{\sigma^{(g)}} \tag{6.5}$$

$$\begin{aligned}\mathbf{C}^{(g+1)} &= (1-c_{\text{cov}})\mathbf{C}^{(g)} + \frac{c_{\text{cov}}}{\mu_{\text{cov}}} \\ &\quad \times \left(\mathbf{p}_c^{(g+1)}\,\mathbf{p}_c^{(g+1)^T} + \delta\left(h_{\sigma}^{(g+1)}\right)\mathbf{C}^{(g)}\right) \\ &\quad + c_{\text{cov}}\left(1-\frac{1}{\mu_{\text{cov}}}\right)\sum_{i=1}^{\mu} w_i\,\text{OP}\left(\frac{\mathbf{x}_{i:\lambda}^{(g+1)}-\mathbf{m}^{(g)}}{\sigma^{(g)}}\right)\end{aligned} \tag{6.6}$$

where $\text{OP}(\mathbf{X}) = \mathbf{X}\mathbf{X}^{\mathbf{T}} = \text{OP}(-\mathbf{X})$.

Notably, the application of CMA-ES is characterized by minimal necessity for parameter tuning. Users are not asked with determining the internal parameters of the strategy. Additionally, the default population size λ is intentionally set relatively small to facilitate rapid convergence. It has been observed that incorporating restarts with an escalating population size proves beneficial for enhancing the global search performance [2], and this practice is now integrated as a standard option in the algorithm.

It was decided to explore CMA-ES performances on the backlash detection problem as an initial literature research suggested CMA-ES as promising in various scenarios more than other well-known algorithms as PSO (Particle Swarm Optimization) or ACO (Ant Colony Optimization). In particular, the two works [55], [57] were used as references. In the first CMA-ES and PSO were tested on a set of problems called the Black-box Optimization Benchmarking (BBOB) suite. The suite is composed of several problems that vary in complexity and dimensions, mimicking real-world scenarios. The study reveals that CMA-ES shows superior performances, achieving a higher success rate and uncovering the absolute best solutions (global optima) more frequently. This is particularly true for problems that are poorly defined (ill-conditioned) and involve many variables (high-dimensional). The second study examines how CMA-ES and PSO algorithms perform on problems with multiple potential solutions (multimodal problems). Although PSO sometimes finds solutions faster, CMA-ES excels at discovering various optimal solutions and avoiding getting stuck in dead ends (local traps) thanks to its superior exploration methods. Finally, the work [10] provides a literature survey on CMA-ES comparing CMA-ES to other algorithms and emphasizing how it stands out in three key areas:

1. Finding the absolute best solution (global optimization): CMA-ES excels at navigating complex landscapes to reach the truly optimal outcome, not just settling for a local good one.

2. Handling noisy data: Even when information is uncertain or misleading, CMA-ES remains reliable, making it suitable for real-world problems with messy data.

3. Working efficiently on multiple processors: When tackling complex problems, CMA-ES can leverage multiple computing cores to solve them faster, saving you time and resources.

Overall, this survey paints a positive picture of CMA-ES, showcasing its strengths in various optimization scenarios. In conjunction with the above, it is also important to note that references provide insights but don't guarantee definitive superiority. Each problem and optimization task have unique characteristics and its effectiveness strictly depends on the specific needs and the problem's characteristics.

6.2.2 Artificial swarms

Swarm intelligence represents another class of metaheuristics, characterized by the coordinated and self-organized actions of decentralized agents within a population or swarm. Numerous algorithms fall within this category, all drawing inspiration from the observation of animals and their social behavior. All these algorithms base their effectiveness on a two-stage approach that integrates a broad exploratory stage with a focused exploitative stage in an iterative fashion. The choice of the best strategy depends on the specific characteristics of the optimization problem at hand. Experimenting with different approaches and combinations can help find effective solutions for non-convex multivariable optimization problems. To this intent, four cutting-edge nature-inspired algorithms were tested to determine their suitability for the backlash signature recognition problem on the motor signal. The considered algorithms are: the Dragonfly Algorithm, the AntLion Optimizer, the Grasshopper Optimization Algorithm, and the Grey Wolf Optimizer. All are developed by the same author, Seyedali Mirjalili, and available for free download from the internet. The algorithm and the principle underlying their operation are described here below.

The Ant Lion Optimizer

The Ant Lion Optimizer (ALO) was introduced in 2015 [37], inspired by the predatory behavior of the antlion insect, particularly its hunting strategy for capturing ants. The antlion constructs cone-shaped traps in the sand, hiding beneath the pit and employing sand-throwing to capture prey. Once an ant approaches the area where an antlion is hidden, it ends up being 'trapped' in the proximity of the predator and gradually converges towards the bottom of the trap where the antlion captures it. The cone-shaped pit of the antlion and the walk of the ant in the area around the antlion is reported in Figure 6.4.

In the ALO algorithm, four main steps are involved: the random positioning and movement of ants, the establishment of traps, the entrapment of ants within these traps, and the subsequent capture of victims. Both ants and antlions are associated with potential solutions, and their fitness is evaluated at each iteration.

Like real ants, these virtual ones wander randomly across the search space, exploring possible solutions. Unlike ants, antlions remains stationary at specific points, representing the best solutions found so far. Each antlion's "trap" size reflects

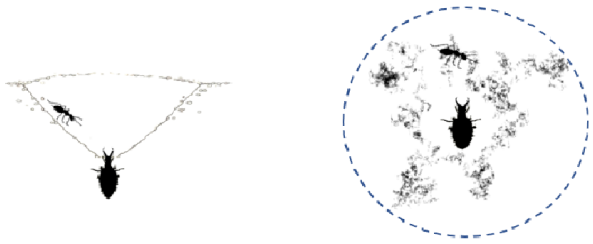


Fig. 6.4 Antlion cone-shaped pit and ant random walk of the ant in the area around the antlion (on the left)

its fitness score, with bigger traps for better solutions. Each ant is linked to a specific antlion. The ant explores the area around its assigned antlion's trap, searching for even better solutions. If an ant finds a solution better than its antlion's current best, the antlion "moves" to that new location and its trap expands to reflect the improved fitness. Once the exploration stops, the antlion with the highest fitness ("best trap") is designated as *elite*. Its final position represents the best solution found for the optimization problem. The algorithm flowchart is represented in Figure 6.5.

In the ALO, the random walk undertaken by ants, x(t), is defined from the vector of random shifts

$$s(t) = [0, (2r(t_1) - 1), (2r(t_2) - 1), ..., (2r(T) - 1)] \tag{6.7}$$

as

$$x(t) = cumsum(s(t)). \tag{6.8}$$

Here, *cumsum* is the sequence of partial sums of elements in the vector, t_i counts for iterations (and correspond to steps of random walk), T is the maximum number of iteration, and $r(t)$ is a stochastic function

$$r(t) = \begin{cases} 1, & \text{if } rand > 0.5 \\ 0, & \text{if } rand > 0.5 \end{cases}$$

yielding values of 0 or 1 with equal probability. Lastly, *rand* is a random number generated with uniform distribution in the interval of [0,1].

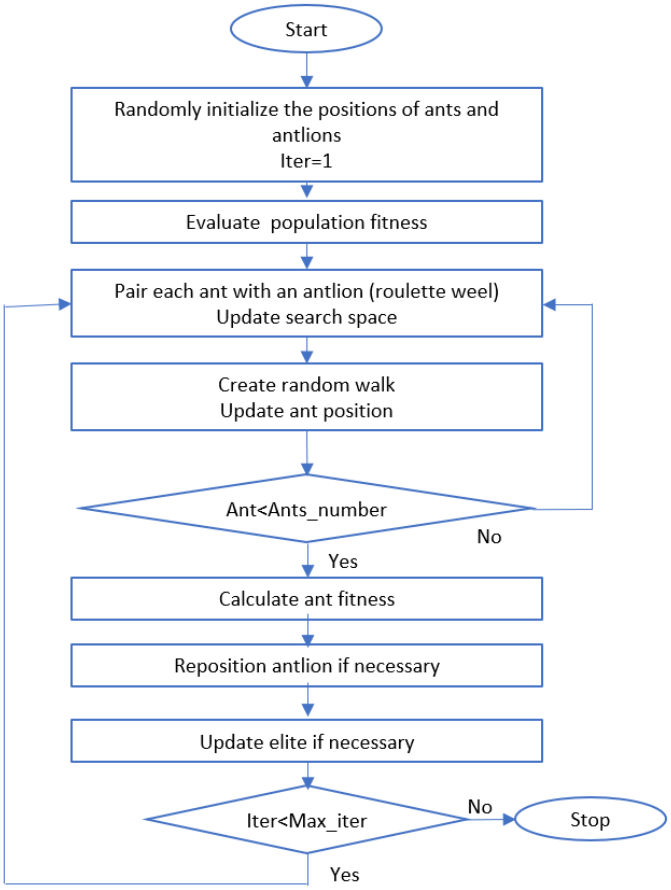


Fig. 6.5 ALO algorithm flowchart

The positions of ants are stored in the matrix

$$\mathscr{M}_A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1d} \\ A_{21} & A_{22} & \dots & A_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nd} \end{bmatrix} \tag{6.9}$$

with dimensions n (number of ants) by d(problem dimension). At every iteration each ant is evaluated, $f(\cdot)$ is the fitness function and the fitness values of ants are

stored in the matrix

$$\mathscr{F}_A = \begin{bmatrix} f([A_{11}, A_{12}, ..., A_{1d}]) \\ f([A_{21}, A_{22}, ..., A_{2d}]) \\ \vdots \\ f([A_{n1}, A_{n2}, ..., A_{nd}]) \end{bmatrix} \tag{6.10}$$

and both matrices are updated at each optimization step. Similar matrices $\mathscr{M}_{AL}$ and $\mathscr{F}_{AL}$, are maintained for position and fitness of antlions.

Ants position is updated at every step of optimization, to keep the walk inside the search space the position are normalized using the formula

$$X_i^t = \frac{(X_i^t - a_i) \times (d_i^t - c_i^t)}{(b_i - a_i)} + c_i \tag{6.11}$$

where a_i, b_i is the minimum and maximum random walk of i-th variable respectively, c_i^t, d_i^t is the minimum and maximum of i-th variable at iteration t.

The effect of antlion attraction on ant movement is modeled by adjusting c^t and d^t at every iteration through the formulas

$$\begin{aligned} c_i^t &= Antlion_j^t + c^t \\ d_i^t &= Antlion_j^t + d^t \end{aligned} \tag{6.12}$$

where $Antlion_j^t$ is the position of antlion j at iteration t. The algorithm models the prey falling into the cone-shaped trap by decreasing the radius of the random walk using specific equations:

$$\begin{aligned} c^t &= \frac{c^t}{I} \\ d^t &= \frac{d^t}{I} \end{aligned} \tag{6.13}$$

The variables c^t and d^t represent the minimum and maximum, respectively, of all variables at each iteration t, and I is a decreasing ratio calculated to control the random walk as $I = 10^w \frac{t}{T}$. In the equation, T represents the maximum number of iterations, and w is a constant determined by the current iteration. The value of w incrementally rises as iterations advance, regulating the precision level of the exploitation phase.

In this way, the ant gradually converges to the antlion and is captured. The ant becomes fitter than its corresponding antlion, and the antlion has to update its position to the latest position of the hunted ant.

In the algorithm each ant is associated to a corresponding antlion. the selection is based on a roulette wheel selection. The random movement of an ant depends on the position of the corresponding antlion and also by the position of the elite. This dependency is expressed as

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \tag{6.14}$$

where R_A^t and R_E^t are random walks around the antlion and elite at the same iteration.

At the end of every iteration step the best antlion is compared to the best antlion found during optimization (elite) and substituted if it is necessary.

Grasshopper Optimization Algorithm

The Grasshopper Optimization Algorithm (GOA) [39] is a swarm-based algorithm inspired by the social behavior of grasshoppers searching for food. Mimicking the two main phases of a grasshopper's life, nymph and adulthood, the algorithm incorporates swarming behavior during both phases. In the nymph stage, grasshoppers move slowly on the ground, consuming vegetation, while in adulthood, they develop wings for faster movement and long-distance flight. The algorithm utilizes these phases to represent exploitation and exploration in optimization.

In GOA, grasshoppers symbolize potential solutions, moving within the search space. The fittest grasshopper at each optimization step becomes the *"food position"* or the best solution so far. This position influences other grasshoppers, prompting them to move towards it, potentially discovering fitter positions. As the optimization progresses, better solutions are continuously found, and the algorithm converges towards the optimum.

In GOA, the position X of the i-th grasshopper is determined by three forces: social interaction (S_i), gravitational attraction (G_i), and wind flow (A_i). The formula is:

$$X_i = S_i + G_i + A_i \tag{6.15}$$

Gravitational attraction, G_i is calculated as follow

$$G_i = -g\widehat{e_g} \tag{6.16}$$

given by the gravitational constant and the unity vector $\widehat{e_g}$ in the direction of the Earth's center. Wind drift, A_i, is computed as

$$A_i = u\widehat{e_w} \tag{6.17}$$

where u is a constant drift and $\widehat{e_w}$ a normalized vector in the wind direction.

The social interaction depends by the distance between grasshoppers and is provided by the formula

$$S_i = \sum_{j=1, j\neq i}^{N} s(d_{ij})\widehat{d_{ij}}. \tag{6.18}$$

where d_{ij} is the distance between grasshopper i and grasshopper j, computed as $d_{ij} = |X_j - X_i|$, N is the number of grasshoppers. While $\widehat{d_{ij}} = \frac{X_j - X_i}{d_{ij}}$ is the unitary vector between i-th and j-th grasshoppers. The Equation 6.15 then becomes

$$X_i = \left(\sum_{j=1, j\neq i}^{N} s(|X_j - X_i|)\frac{X_j - X_i}{d_{ij}} \right) - g\widehat{e_g} + u\widehat{e_w}. \tag{6.19}$$

Attraction or repulsion among grasshoppers depends on the power of social interaction $s(d)$ that is given by the formula

$$s(d) = fe^{-\frac{d}{l}} - e^{-d} \tag{6.20}$$

with l as the scale of attractive length and f as the intensity of attraction. As a result, the space around a grasshopper is divided into 3 zones: repulsion, attraction, and comfort zone, see Figure 6.6.

However, the mathematical model of Equation 6.19 is not used directly for the optimisation problems while a modified version of the equation is proposed. Gravity (G_i) is disregarded, and wind direction (A_i) is considered as always directed towards the food (i.e., the target $\widehat{T}$). Moreover, as the swarm quickly reach the comfort zone and does not converge to a given point as grasshoppers, a coefficient that decreases over iterations to prevent grasshoppers from being stuck in an equilibrium position

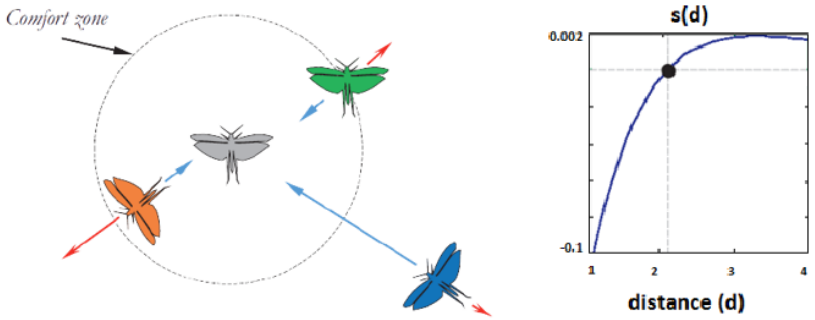


Fig. 6.6 The hyper-sphere of social attraction around a grasshopper and the resulting 3 zones of attraction, repulsion, equilibrium (on the left). The function s(d) that describes the power of social interaction for grasshoppers (on the right).

id added:

$$X_i = c\left(\sum_{j=1, j\neq i}^{N} c\frac{ub-lb}{2}s(|X_j - X_i|)\frac{X_j - X_i}{d_{ij}}\right) + \widehat{T}. \tag{6.21}$$

In this context, c is a diminishing coefficient that decreases as the iterations progress, leading to a reduction in the repulsion region between grasshoppers, ub and lb denote the upper and lower bounds of the search space, respectively. $\widehat{T}$ represents the best value discovered thus far, and $s(.)$ is the function defined in Equation (6.20). The presence of c in the formula twice is deliberate, as it serves to influence both exploration and exploitation behaviors. As c decreases, it limits the movements of grasshoppers around the target, emphasizing exploitation over exploration. The coefficient is updated based on the current iteration l, and a maximum number of iterations L, thought the formula

$$c = c_{max} - l\frac{c_{max} - c_{min}}{L} \tag{6.22}$$

where c_{max} and c_{min} are the maximum and the minimum value, respectively. The main steps of the algorithm are summarized in Figure 6.7.

Grey Wolf Optimizer

The Grey Wolf Optimizer (GWO) is an optimization algorithm based on the social and hunting behavior of grey wolves [38]. Grey wolves live in hierarchical packs with distinct roles for alpha, beta, delta, and omega wolves. The alpha wolf leads

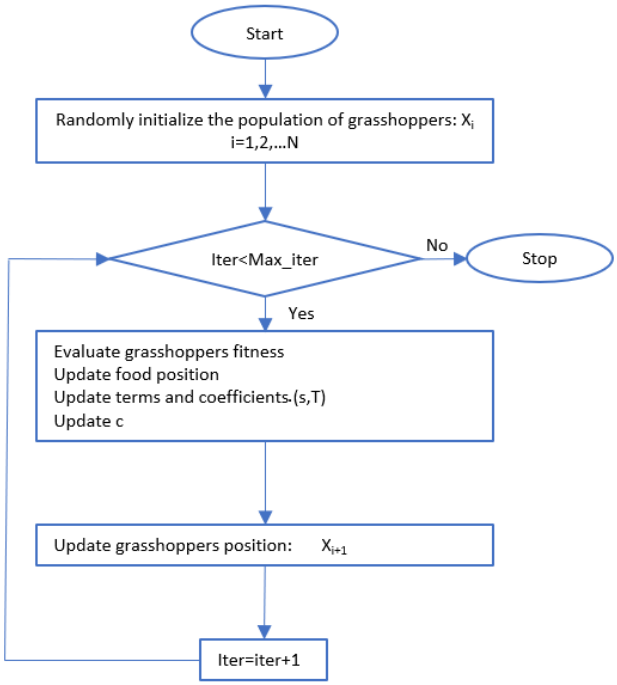


Fig. 6.7 The flowchart of the GOA algorithm

the pack, assisted by beta and delta wolves in decision-making and essential tasks such as hunting. Omega wolves are subordinate members. The hunting technique of Grey Wolves involves searching for prey, surrounding it, and then attacking it. The GWO algorithm mimics these behaviors with each wolf representing a potential solution. The prey is the best solution of the problem. Hierarchy depends on fitness. During each iteration, the solutions are evaluated, and the fittest one becomes the alpha wolf, with the second and third best solutions as beta and delta wolves. It is these wolves that determine the position of the prey, which will be within the area that these wolves circumscribe. For this reason these 3 wolves influence the next iteration's wolf positions. The algorithm aims for convergence to the prey while ensuring divergence among wolves. The exploration or exploitation behavior of wolves can be adjusted by manipulating specific coefficients in their position formula.

The generic position of a wolf around prey is represented by two equations:

$$\vec{D} = |\vec{C}\vec{X}_p(t) - \vec{X}(t)| \tag{6.23}$$

$$\vec{X}(t+1) = |\vec{X}_p(t) - \vec{A}\vec{D}|. \tag{6.24}$$

Here, t is the current iteration, $\vec{X}_p(t)$ and $\vec{X}(t)$ the position vectors of the prey and the grey wolf at iteration t respectively. In the formulas are also some coefficient vectors, namely

$$\vec{A} = 2a\vec{r}_1 \tag{6.25}$$

$$\vec{C} = 2\vec{r}_2 \tag{6.26}$$

where $\vec{r}_1$, and $\vec{r}_2$ are vectors of random elements in [0,1] and a is a coefficient that linearly reduces from 2 to 0 as the optimization proceeds. The position of an omega wolf is updated by considering the positions of alpha, beta, and delta wolves, as it depends on the positions of these wolves:

$$\vec{D}_\alpha = |\vec{C}_1\vec{X}_\alpha - \vec{X}| \tag{6.27}$$

$$\vec{D}_\beta = |\vec{C}_2\vec{X}_\beta - \vec{X}| \tag{6.28}$$

$$\vec{D}_\delta = |\vec{C}_3\vec{X}_\delta - \vec{X}|. \tag{6.29}$$

where $\vec{X}_\alpha$, $\vec{X}_\beta$, $\vec{X}_\delta$ indicates the position of alpha, beta and delta wolves respectively, and $\vec{C}_1, \vec{C}_2, \vec{C}_3$ are vectors of random coefficients. The $\vec{A}$ and $\vec{C}$ vectors are randomly generated and employed to adjust the balance between the exploration and exploitation phases in the GWO algorithm. The main elements for wolves position update and the flowchart of the algorithm are reported in Figure 6.8.

Dragonfly Algorithm

The Dragonfly Algorithm (DA) [36] is an optimization technique inspired by the collective behavior of dragonflies, capturing elements from both their static feeding and dynamic migratory swarming. These two behaviours directly correspond to the two phases of an optimisation process: exploration and exploitation. The algorithm is grounded in swarm principles encompassing separation, alignment, and cohesion,

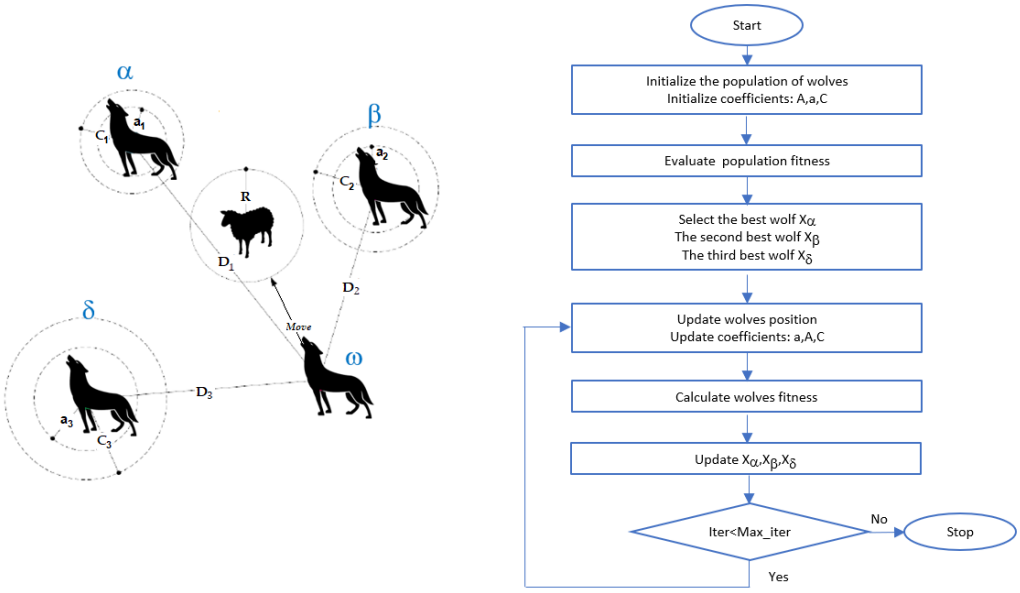


Fig. 6.8 Main elements of position updating in GWO (on the left) and algorithm flowchart (on the right).

as identified by Reynolds in [48] and [49], and integrates attraction towards a food source and distraction from enemies. The two main behaviors of dragonfly swarms, and the 5 basic principles underlying their behaviour are reported in Figure 6.9. The separation term prevents collisions, alignment ensures consistent velocity, and cohesion drives individuals to converge towards the swarm's center.

The separation term,

$$S_i = -\sum_{j=1}^{N} X_i - X_j \tag{6.30}$$

defines individual avoidance within a visibility radius. Alignment,

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \tag{6.31}$$

regulates velocity consistency; while cohesion

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X_i. \tag{6.32}$$

encourages convergence. The algorithm also incorporates attraction towards food

$$F_i = X^+ - X_i \tag{6.33}$$

and distraction from enemies

$$E_i = X^- + X_i \tag{6.34}$$

with $X+$ and $X-$ denoting the positions of the food source and enemy, respectively. These principles collectively steer swarm movement, mirroring the swarming behavior of dragonflies during different flight phases. Individuals navigate through the search space, and their positions are updated at each iteration following the rules:

$$X_{t+1} = X_t + \Delta X_{t+1} \tag{6.35}$$

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t. \tag{6.36}$$

The velocity vector ΔX of the dragonfly, dictating its direction and speed, is influenced by the inertia weight w and iteration counter t. Coefficients s, a, c, f, e play a role in various behaviors, steering dynamic swarm behavior in the initial optimization stages characterized by high coordination, aligned speeds, and robust cohesion and separation. With optimization progress, the swarm shifts to static swarming, featuring increased cohesion and reduced alignment and separation, enhancing convergence toward the prey. This transition is regulated by adjusting the radius r and coefficients s, a, c, f, e based on the iteration count. Notably, food and enemy positions are updated at the beginning of each iteration, with the fittest individual determining the food location and the worst individual indicating the enemy's position. The steps of the algorithm are reported in Figure 6.10.

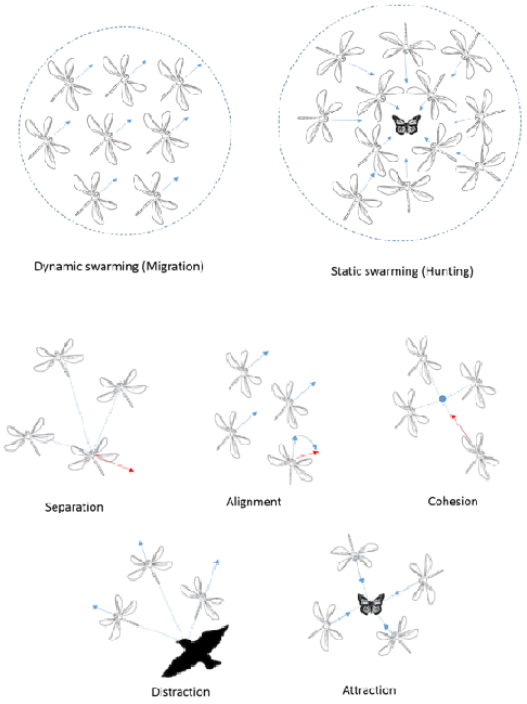


Fig. 6.9 The two main behaviors of dragonfly swarms, and the 5 basic principles of swarms

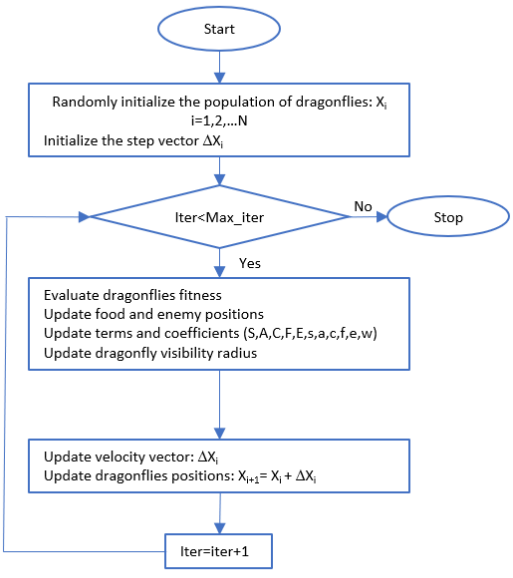


Fig. 6.10 The flow diagram of the DA algorithm

Chapter 7

Implementation and testing

7.1 First test of the method

The first implementation of the approach introduced in Chapter 5 to estimate backlash were carried out by employing the CMA-ES optimization algorithm to solve the signature fitting problem. This choice was made based on the algorithm's suitability and effectiveness in addressing black-box optimization problems where the mathematical form of the objective function is unknown, complex, or computationally expensive to evaluate. This is the exact case of our fitting problem where the function to be minimized is a complex multi-variable formula. CMA-ES excels in such scenarios because it doesn't rely on explicit knowledge of the objective function or its derivatives.

The test for the assessments of the complete analysis pipeline involved utilizing two signals acquired from the test bench. These signals were obtained at two distinct time points, corresponding to different and increasing levels of backlash in the joint. The goal of the test was to verify the method's ability to recognize the backlash disturbance in the signal, provide an estimate of its magnitude, and also distinguish between two different and progressive states of the problem.

The CMA-ES algorithm was the tool selected to process the signals, detect the oscillations induced by backlash and deriving the parameters of the disturbance model. Through the estimation of these parameters, particularly the parameter A, which we know is closely linked to the amount of backlash in the joint, it is possible to gain insight into the internal state of the joint and monitor changes over time, as

explained in Paragraph 5.3. Python was selected as the implementation language due to its user-friendly syntax, extensive ecosystem, open-source nature, compatibility with libraries, and cross-platform capabilities.

7.1.1 Setting of initial conditions

A very important aspect has been the definition of the initial values for variables and parameters for the optimization algorithms, as these values have a substantial impact on both algorithm's performance and its convergence.

In an optimization algorithm, the choice of initial values act as the starting point for the optimization process, determining the trajectory of the algorithm's search through the solution space. A well-chosen set of initial conditions can lead convergence towards optimal solutions, enabling the algorithm to explore promising regions efficiently. Conversely, inadequate initial conditions may lead the algorithm wandering, resulting in prolonged convergence times or the risk of converging towards suboptimal solutions. Carefully select initial conditions has a direct impact on the efficiency and effectiveness of the optimization algorithm. Moreover, prioritizing the establishment of suitable initial conditions helps alleviate algorithmic instability and ensures consistent convergence towards high-quality solutions across diverse instances of the same problem.

The initial phase of implementing CMA-ES for the optimization problem defined in the preceding chapters, involved setting up the initial conditions for the algorithm. Preliminary trials uncovered a noteworthy sensitivity of the algorithm to the initial conditions, as even slight variations in the starting point could lead to different outcomes. Consequently, this triggered the quest for a systematic definition of the initial conditions to be used.

Automatic procedure

To elevate the algorithm's capacity to consistently converge towards high-quality solutions across diverse instances of the same problem and to eliminate the need for manual selection of initial conditions, an automatic procedure to calculate the initial conditions was developed. The procedure leverages the a priori knowledge of the test system and the characteristics of the velocity signal v(t) sampled from the

system, which is used as input for the optimization algorithm. The process begins by analyzing the signal provided as input to the algorithm and extracting from it the range of variability for each parameter (see Table 7.1). Subsequently, the starting values for the mean x_0 and the variance σ_0 of each parameter, which are the required inputs of CMA-ES, are estimated using the following formulas:

$$x_0 = \frac{Max\_value + Min\_value}{2}$$
$$\sigma_0 = \frac{Max\_value - Min\_value}{4}$$

The possible solution is in the form:

$$X = [A, t_0, \tau, \omega, T_1, T_2, T_d, T_f]$$

and the vector dimension is 9. Given the physical significance of all parameters and the availability of a priori information about the system under test, it is feasible to leverage this information to establish upper and lower bounds for each parameters. Moreover, as test condition are known and constant, T_d and T_f have a fixed and predetermined values and the effective problem dimension reduces to 7.

The table below reports parameters upper and lower bounds for the identification. Benchmark values are obtained from the system specifications assessed in the experiments.

Table 7.1 Ranges for the model's parameters. In the formulas, t is the time vector of the measured signal $v(t)$

Symbol	Min_value	Max_value	Units
A	$-\frac{\max v(t) - \min v(t)}{2}$	$\frac{\max v(t) - \min v(t)}{2}$	rps
t_0	$\min t$	$\max t$	s
τ	5	30	-
ω	2π	$2\pi \cdot 40$	rad/s
v_t	$\min v(t)$	$\max v(t)$	rps
T_1	0	$\frac{0.204}{2}$	s
T_2	0	0.204	s
T_d	0.204	0.204	s
T_f	0.408	0.408	s

Using initial conditions defined in this way and setting a population size of 3500 individuals, it has been observed that the CMA-ES algorithm converges to a satisfactory solution (i.e., error = 0.14) in approximately 100 iterations.

7.1.2 Test of the full toolchain

The initial tests on the entire analysis pipeline were conducted using two signals, $v_1(t)$ and $v_2(t)$, collected from the test bench. The two signals were gathered at two moments distant in time, corresponding to two different, and growing, values of backlash in the joint. These datasets were collected from the test bench approximately 4 months apart during a continuous operation cycle of the device.

The tests were successful as CMA-ES algorithm proved able to identify the increasing backlash gap. The algorithm adeptly detected the known disturbance pattern in the speed signal and provided estimates for its parameter values.

The two separate identifications yielded two increasing values for the oscillation amplitude A:

$$A_1 = 0.515055\,rps\,, A_2 = 0,653661\,rps.$$

The results are shown in Figure 7.1. The figure demonstrates that CMA-ES successfully derived an average value for the model parameters for both datasets, enabling an accurate approximation of the initial signal despite high noise.

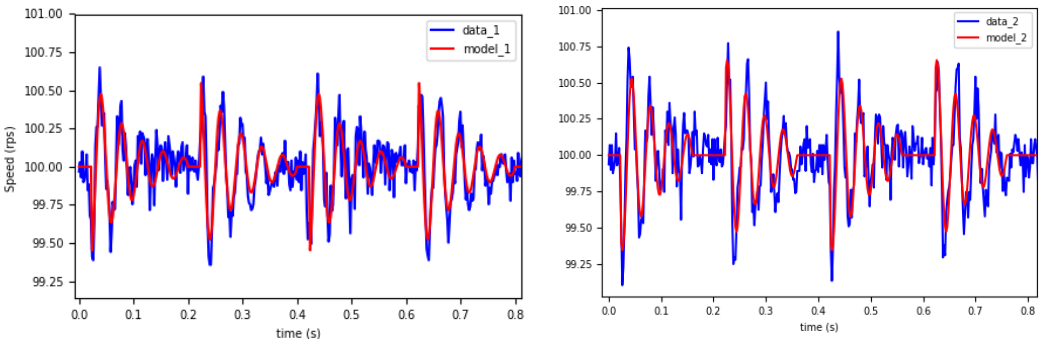


Fig. 7.1 Identification results for Dataset1 (on the left) and Dataset2 (on the right) using CMA-ES. The blue plots represent the signals from the test bench, the red plots represent the model reconstruction based on the parameters identified using CMA-ES.

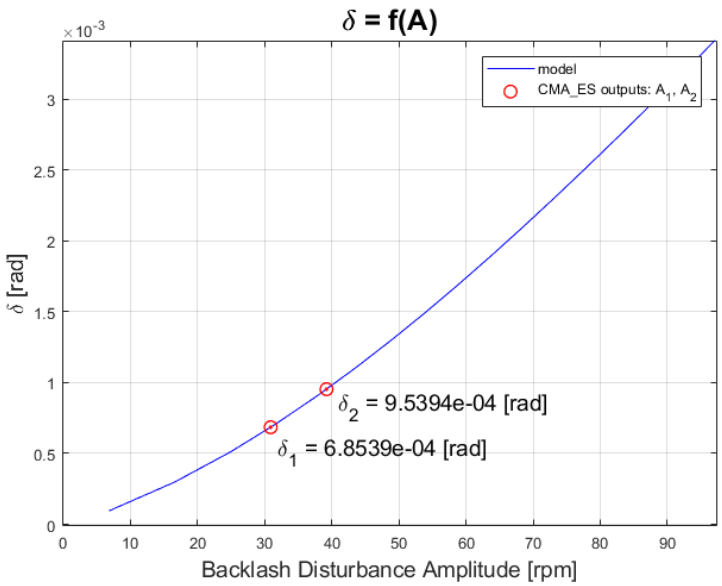


Fig. 7.2 Final backlash estimate evaluation.

Values become

$$A_1 = 30.9033\,rpm\,, A_2 = 39.2197\,rpm.$$

when considering revolute per minute instead of revolute per seconds.

At this point, the values of fault indicator, A, were processed through the relationship between the fault indicator and the corresponding backlash value in the joint, $\delta = f(A)$, provided by the formula (4.7).

Through this formula, the corresponding backlash values obtained were:

$$\delta_1 = 6.8539 \times 10^{-4}\text{rad}\,,\ \delta_2 = 9.5394 \times 10^{-4}\text{rad}.$$

The function $\delta = f(A)$ and the calculated values for A and δ are reported in Figure 7.2. The obtained results highlights the algorithm's ability to sensitively detect changes in parameter A, allowing for the characterization of variations in the backlash gap over time.

Unfortunately it was not possible to validate the quality of the estimate of the backlash value in absolute terms as no direct measure of backlash were available. Due to limited access to the test bench installation area, only automatic signal

collection from the system was feasible, precluding manual measurements on the system.

7.2 Optimization tool benchmarking

A benchmarking process for multiple optimization strategies was then performed to select the most suitable meta-heuristic for the problem addressed. To this intent CMA-ES was compared to 4 different swarm-based algorithms with the goal of examining crucial factors such as ease of implementation, resources consumption, and accuracy of the proposed solutions. Out of the numerous meta-heuristics available, swarm algorithms were chosen due to their straightforward implementation and their minimal demands on computational and memory resources in comparison to other meta-heuristics.

7.2.1 Dataset creation

The first step was to have a comprehensive dataset for the analysis. This dataset was created through simulation using the Matlab/Simulink model of the robotic joint. The set consists of 20 signals, each affected by a different level of backlash, recorded on the motor while the transmission model was parameterized with a progressively increasing value of backlash. Dataset1 encompasses the signal corresponding to the lowest backlash level, while Dataset20 contains data associated with the highest value of backlash. The datasets along with their respective backlash values (in blue) and disturbance amplitudes (in red) are depicted in Figure 7.3. These datasets cover the 20 backlash values within the interval $[\delta_{min}; \delta_{max}] = [0.0001; 0.004]$rad, with a constant step size of $\Delta\delta = 0.0002$rad. As it denotes the smallest interval between two potential measurements, the step size was also adopted as the required accuracy for the backlash estimate method.

7.2.2 Optimization parameters setting

Differently form CMA-ES, the swarm-based algorithms required a further step of constraint definition. It was necessary to implement the interdependence of some

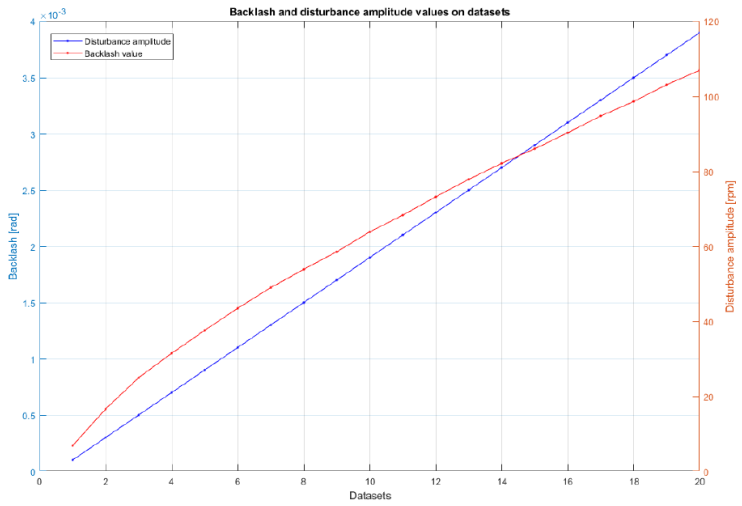


Fig. 7.3 Backlash values and corresponding disturbance amplitude on the considered interval of analysis.

parameters to ensure that the final resulting signature aspect aligned with the expected outcome. Specifically, constraints were established for t_0, T_1, and T_2 as follows:

$$0 < t_0 < (max(t) - T_1),$$

$$T_2 < T_1 < T_d,$$

$$0 < T_2 < \frac{pi}{2\omega}.$$

A constraint check was added at the end of every iteration, when the agents' positions are updated for the next cycle and after verifying whether the new values fall within the defined domains. The rule to manage a parameter's value falling outside the interval specified by the constraints was capping the value at the extreme limit of the allowable range. For the sake of comparison, the algorithms were executed under identical conditions: the same number of agents and iterations, identical constraints, and same input data. The optimal configuration was identified as 300 agents and 200 iterations. This selection was made with the DA algorithm as a reference, employing a trial-and-error approach. The same setting was also used for CMA-ES for the sake of comparison.

7.2.3 Experiments execution

Each algorithm was executed on the same dataset defined in Paragraph 7.2.1. As initial tests indicated that the backlash disturbance in the first two datasets was too minimal to be accurately identified, these datasets were excluded from the analysis. The focus was directed towards the assessment of the remaining 18 datasets.

Thirty independent experiments were conducted for each dataset, maintaining consistent conditions across all trials. The testing protocol, including the setting and algorithms, remained uniform for all experiments. Additionally, to ensure equal opportunities for each algorithm, the starting solution was consistently chosen at random. The computations were conducted on a workstation featuring a 2.70 GHz processor, 16 GB of RAM, and running on the Windows7 operating system.

Figure 7.4 displays a collection of all the results.

The graphs provides a summary of the backlash disturbance amplitude values obtained from the many tested optimization methods. In plots in the upper part of the figure the outcome of each performed experiment is reported as a dot. The vertical columns of stacked dots correspond to the 30 outcomes of experiments on the same dataset. The red line in the graph connects the expected values for the disturbance amplitude, while the red dotted lines indicate the maximum allowable error for the measurement, i.e., the defined accuracy on δ reported on the value of A.

$$\Delta\delta = 0.0002\text{rad} \rightarrow \Delta A = 3.8672 rpm$$

$$A = A_{expected} \pm \frac{\Delta A}{2}$$

Given the nonlinear relationship between A and δ, the accuracy ΔA, is defined as the minimum distance between two consecutive values of A plotted in Figure7.3. The figure clearly shows as CMA-ES is the most suited meta-heuristic for the problem at hand.

Swarm algorithms comparison

By analysing the results in more detail, some further consideration can be done about the comparison of the 4 swarm-based algorithms.

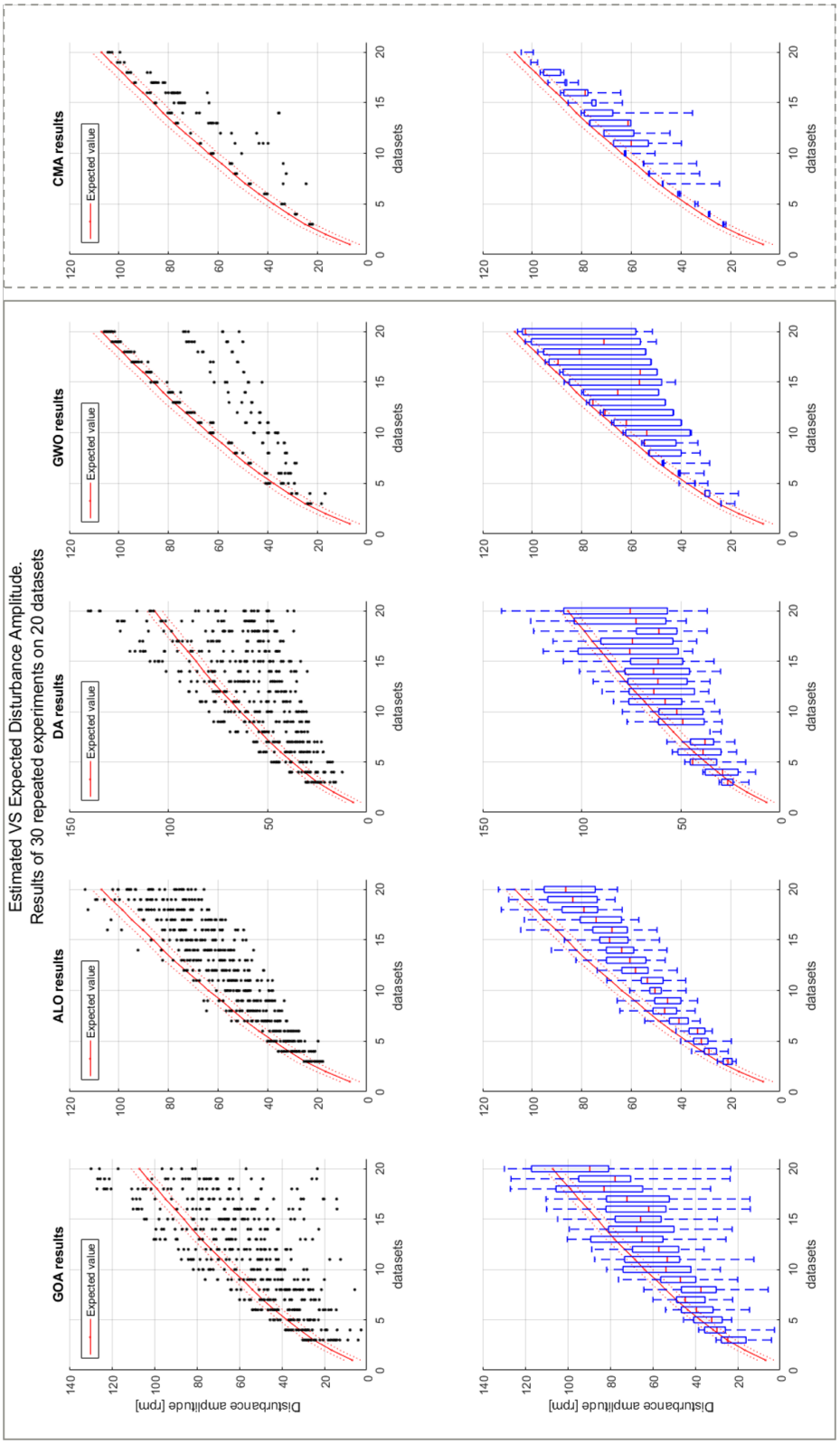


Fig. 7.4 Compilation of the test results presented using two different visualization modes: as points or with box and whisker plots.

Observing the plots it becomes evident that the distribution of results widens as we move towards larger values of backlash, progressing from left to right in the plots. This widening is not a consequence of increased problem complexity but is influenced by the way the problem search space is defined. Specifically, the amplitude-related parameter of the model, A, has boundaries that expand as we move through the datasets on the right, leading to an enlargement of its search space, as discussed in Section 7.1.1. Looking at the same plots, it also appears that the solution is more often underestimated than overestimated in the case of swarm algorithms. This tendency can be attributed to fitting errors. In cases of fitting failure, only the second and smaller half of the decreasing oscillation is fitted, while the first and larger part is lost, resulting in an underestimated solution. A different perspective on the same data is presented through the box and whisker diagrams in Figure 7.4. These diagrams provide a quantitative representation of the dispersion of results around the expected value. While the diagrams offer clear insights into the performance of GOA, ALO, and DA, they are not as informative for GWO. This limitation arises from the unique shape of the GWO results' dispersion, with the majority closely aligned with the red expected value and a wide empty band separating the remaining data. Consequently, the box plot may be misleading for GWO. Therefore, even though the box charts indicate ALO as the best in terms of precision, a more in-depth analysis reveals that GWO outperforms all others.

A final observation can be made regarding the execution time of a single experiment. Figure 7.5 shows the average execution time for a single experiment in the case of the different algorithms.

To summarize, the 4 swarm algorithms under consideration were deemed inadequate for solving the backlash signature fitting problem. The issue stems from the presence of numerous local minima, and none of the algorithms demonstrated sufficient exploration capabilities to avoid becoming trapped in one of these local solutions. Despite certain advantages such as reduced memory usage and ease of implementation, results confirmed that bio-inspired swarm algorithms are unsuitable for the problem addressed.

A statistical test procedure was also employed, as relying solely on mean performance assessment is insufficient to compare stochastic algorithms effectively. Therefore, the results were evaluated using a statistical test procedure. The series of 30 results obtained from experiments on a single dataset was treated as a set

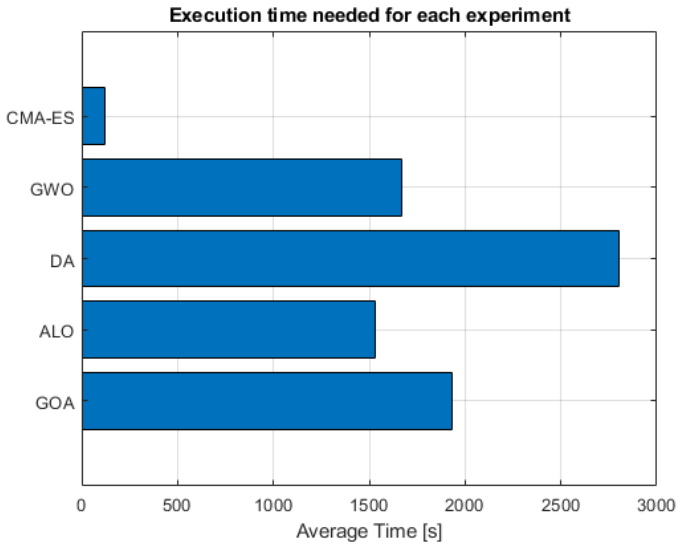


Fig. 7.5 Execution time comparison, average time for a single experiment.

of sampled values from a continuous distribution. The two-sample Kolmogorov-Smirnov test [51] was employed to assess the hypothesis that two distinct series of results originate from the same distribution. Specifically, the test is conducted with a significance threshold α of 0.05. The results are presented in Table 7.2, where the reported value is the mean of the fitting error across the 30 repeated experiments. Italics are used for results that likely stem from the same distribution, indicating their lack of significance for comparison. However, in the majority of cases, it is possible to accept the hypothesis that the collected data come from different distributions. In these instances, the best result is highlighted in bold. Once again, GWO emerges as the most favorable among the four algorithms.

Based on the considerations above, it can be concluded that among the 4 analyzed algorithms the GWO algorithm demonstrates superior performance in addressing the backlash problem, while the DA algorithm performs the least effectively.

7.3 Testing noise sensitivity

Once CMA-ES was defined as the best optimisation algorithm to be used within the backlash estimation proposed method, the performance of the full method was tested more thoroughly by considering conditions more similar to those found in reality. To

Table 7.2 Kolmogorov-Smirnov test results

Datasets	ALO	GOA	DA	GWO
3	2.777	4.781	3.227	**2.637**
4	3.331	6.169	4.558	**3.278**
5	4.560	7.214	6.039	**3.327**
6	5.752	9.511	7.422	**3.630**
7	6.728	*8.946*	*9.277*	**4.250**
8	6.622	11.628	9.363	**5.400**
9	7.580	*12.163*	*11.553*	**5.817**
10	8.816	13.065	11.412	**7.310**
11	9.456	*14.159*	*13.399*	**7.957**
12	8.647	16.212	12.846	**8.268**
13	10.218	16.571	14.409	**8.844**
14	10.729	18.311	14.321	**9.289**
15	10.957	17.419	16.261	**10.411**
16	**10.427**	19.766	16.709	10.509
17	11.301	20.982	16.921	**10.521**
18	11.932	20.055	18.920	**11.017**
19	13.628	20.69	17.799	**12.576**
20	13.660	22.126	19.150	**11.214**

this intent a new set of experiments was conducted by considering noise-corrupted data and a random starting point for the input signal.

In real-world scenarios, encountering noise in recorded signals is unavoidable. Noise has the potential to distort and obscure the informative content of a signal, adversely affecting the detection of the backlash signal. The datasets utilized in the preceding set of experiments, as outlined in Section 4.3, were generated through simulation, making them free from noise and solely containing the effects of the phenomena considered in the model. To mimic conditions closely resembling those of a real robot, random noise was introduced to the clean data. Subsequently, the noisy datasets were tested, and the impact of noise on the performance of the fitting algorithm was assessed.

An additive noise model was taken into consideration, and the noisy signal was defined as the sum of the original clean signal, denoted as $v(t)$, and the noise signal $n(t)$.:

$$v_n(t) = v(t) + n(t) \tag{7.1}$$

where:

$$n(t) \sim \frac{A_n - A_{min}}{\Delta A} \cdot \mathscr{U}(-0.5, 0.5) \tag{7.2}$$

The signal $n(t)$ represents uniform white noise with a mean value of 0. The noise amplitude was varied for each dataset and was chosen as directly proportional to the expected amplitude of the corresponding backlash disturbance. This decision was made to prevent situations where a fixed noise amplitude might obscure the backlash information in the signal. Consequently, datasets with minimal backlash disturbance exhibit a lower level of noise, while datasets associated with a more substantial backlash disturbance experience a higher level of noise. The expectation is that systems more affected by backlash will also display larger amounts of noise. Ten datasets were analyzed to gauge the algorithm's capability to detect backlash in its early stages. Given that the chosen optimization algorithm for the approach is stochastic, the identification procedure was executed multiple times on the same dataset to assess result repeatability. So, tests were iterated 30 times on each dataset to derive a statistical distribution of the outcomes.

To better replicate real-world scenarios, an enhancement was introduced for the simulated data sets. Since simulated data always began at a fixed time point, unlike real-world data with an unknown starting point, a random sample was chosen as the starting point for data analysis. This random selection, following a uniform probability distribution, was made from the first 100 samples of the dataset.

The results were compared by evaluating the error between the expected and estimated values of parameter A. All experiments, along with the source code of the tools and the data, are accessible under the *European Union Public Licence*[1], version 1.2 or later, from a public repository on GitHub[2]. A summary of the experimental results is presented in Table 7.3.

In the table above, each row represents 30 runs of CMA-ES on the same dataset. The datasets are arranged based on the corresponding expected value of A, and the amplitude of the noise added to the original clean signal is provided in the second column. The third and fourth columns display the estimate of A produced by the algorithm, expressed as the mean value and the variance of the results over 30 repeated experiments. Finally, the last column presents the relative error between the estimated value of A and its theoretical value.

[1] https://eupl.eu/
[2] https://github.com/ElianaGiovannitti/Backlash-estimation

Table 7.3 Results of the experimental evaluation on artificial datasets with noise and various values of A. The last line illustrates the outcome of the proposed approach applied to a dataset containing only random noise, without backlash disturbance. The values are measured in revolutions per second (rps).

Theoretical value of A (rps)	**Noise amplitude**	**Experimental value of A (rps)**		**Mean relative error (%)**
		mean	**stdev**	
0.2769	0.1079	0.2641	0.0024	4.63%
0.4159	0.1158	0.3870	0.0009	6.94%
0.5245	0.1237	0.4925	0.0138	6.10%
0.6263	0.1316	0.5709	0.0124	8.84%
0.7246	0.1394	0.6731	0.0223	7.11%
0.8162	0.1474	0.8074	0.0127	1.07%
0.8976	0.1552	0.8960	0.0011	0.18%
0.9757	0.1631	0.9269	0.0276	5.00%
1.0639	0.1711	1.0328	0.0570	2.92%
1.1385	0.1789	1.0696	0.0194	6.05%
(no backlash) 0.0	0.1000	0.0099	0.0178	-

Upon examining the data, it is evident that the error consistently remains below the 10% threshold. In terms of backlash value, this implies a maximum absolute error of 0.0004 rad in the estimate of the backlash value for the dataset with the highest backlash. Additionally, the absolute error is lower than 0.0002 rad in the estimate of the backlash value for the dataset with the smallest backlash. This indicates that the proposed approach delivers reliable estimates of the backlash value in the early stages of its manifestation. However, the quality of the estimate tends to diminish as the backlash increases. Another noteworthy result stems from the analysis of a dataset devoid of backlash, containing only noise. The very small estimated value confirms that the algorithm correctly identifies the absence of backlash. Although further refinement of the method could be considered to broaden the range of validity of the estimate, the results obtained thus far are already adequate for the initial objective of this work. They demonstrate that the method can distinguish "healthy" working conditions from the early stages of backlash presence and identify the escalation of backlash in the robot joint.

7.4 Test on real-world data

The concluding segment of the experimental assessment focused on evaluating the performance of the method using data gathered from real-world systems. Specifically, signals were collected on the test bench. Three distinct datasets, representing successive states of the system over time, were employed to assess the algorithm. An overview of the experimental outcomes is presented in Table 3.

Table 7.4 Results of the experimental evaluation on real-world datasets derived from experiments conducted on a physical system with varying degrees of backlash. Each row represents 30 runs of the proposed approach under identical conditions.

Theoretical value of A (rps)	**Experimental value of A (rps)**		**Mean relative error (%)**
	mean	**stdev**	
0.6263	0.5704	0.0206	8.93%
0.6755	0.6678	0.0879	1.14%
0.7246	0.7615	0.0010	5.09%

Chapter 8

From single joint to complete robot

8.1 Tests a complete robotic arm

A final required step is to investigate whether the same methodology developed for the single joint of the test bench can be generalized to joints of a complete industrial robot. The heterogeneity in joint sizes, load capacities, mechanical components, and configurations in a robotic arm can have a significant impact on the performance of the devised method.

In a robotic arm the joints have different load capacities, are made of different mechanical components, and are arranged in different configurations meaning different working conditions for each joint. The variations in joint characteristics are purposefully designed to accommodate the specific needs and functionalities of different parts of the robotic arm. Firstly, the joints differ in size, reflecting their role and function within the arm. Larger joints are capable of bearing heavier loads, and are usually placed at the base of the robot. The smaller joints, with lower load capacity, are instead positioned near the free end of the robotic arm. Different materials, gears, and mechanisms are employed based on the joint's intended use. Additionally, the arrangement of joints within the robotic arm contributes to diverse working conditions. Some joints may have a wider range of motion, allowing for more versatile applications, while others might be optimized for specific tasks. Finally, as the joints are mounted sequentially on the same mechanical structure, there is a tendency for them to exert mutual influence on each other. Each joint's motion within the robotic arm is not isolated; rather, it influences and is influenced by the

movement of other joints due to their mounting on a shared mechanical structure. Consequently, any motion or force applied to one joint has a consequential impact on the dynamics of the neighboring joints. See Figure 8.1 as a reference. For all these reasons it was then fundamental to verify whether the developed method and the results obtained on the test bench, which comprises a single joint, remain valid when applied to different joints of a complete robot arm.

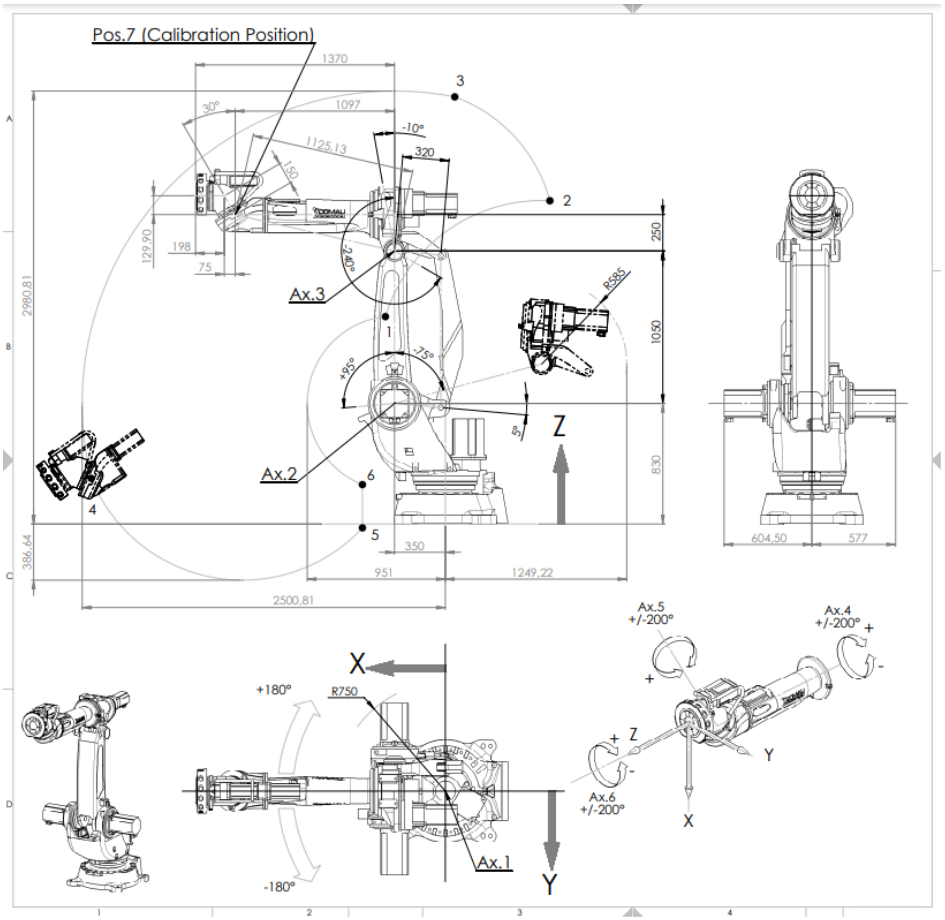


Fig. 8.1 Mechanical details about the articulated robot. The revolute joints and the corresponding range of motion are indicated by curved arrows. Link lengths are also specified. Image from COMAU website

8.2 Data acquisition

Unfortunately, no historical data from a robot affected by backlash were available, and collecting new data was not feasible, as there was no worn-out robot for testing

at the COMAU facility in Grugliasco. Therefore, the decision was made to take a robot in good condition and tamper with the gears of the joint transmissions to induce backlash in the robot. Subsequently, a test campaign was conducted on the tampered manipulator to ensure the ongoing validity of the test conditions identified on the test bench, the backlash observability and the features of the backlash signature.

The robotic arm taken as a reference was the NJ4 170 2.5 (Figure 8.2). It is a medium size robot with $170Kg$ maximum payload and $2.5m$ maximum horizontal reach, mostly used in spot welding applications for car body framing. The test were conducted at the COMAU facility in Grugliasco. On the robot flange a cast iron mass equivalent to the maximum allowed payload was installed.

Fig. 8.2 The 6 axis COMAU robotic manipualtor NJ4 170 2.5 used for the tests. Image from COMAU website

Data were collected by conducting tests on the robot under pristine conditions and on the same robot with induced backlash in the joints. The backlash in the joints was introduced through mechanical tampering. Specifically, the positioning of the motor shaft and gears was altered to increase the space between the teeth of mating parts. Tampering and corresponding data acquisition were performed proceeding one joint at a time, starting from joint 1 and going sequentially up to joint 6. During the execution of the data collection tests, specific joint movements were carried out to isolate the joint under analysis, with all other robot joints held stationary. This approach aimed to minimize influences between the different joints.

Various types of movements were considered for each joint to highlight the backlash effects. Similar to the test bench scenario, these included long movements, short and impulsive movements, motion reversals, and sinusoidal-type movements. However, adjustments were made to the tests programs to address the limited range of motion of the joints due by the complex mechanical structure of the robotic arm. During the tests, both motor position and motor current, were recorded thanks to specific functionalities provided by the robot controller. The velocity signal of the motor was obtained by calculating the derivative of the motor position.

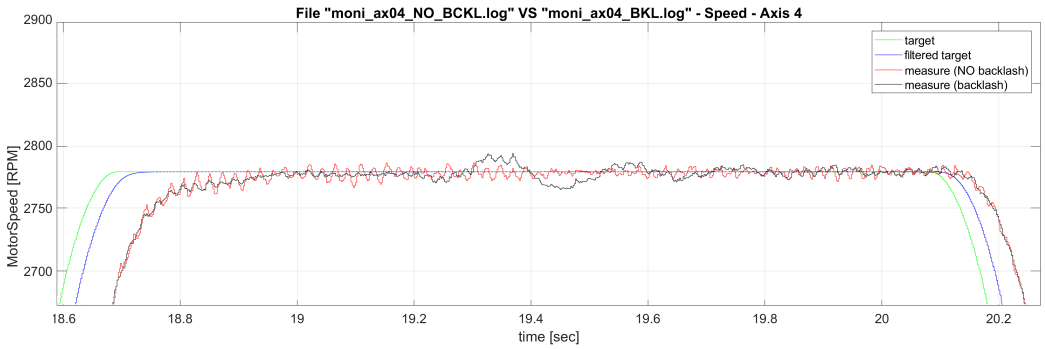
8.3 Test results

The tests have shown that in joints $4, 5$, and 6 of the robot, it is still possible to observe the backlash through the motor encoder, and it is possible to identify and estimate its value automatically by using the method identified in this research work. The speed signal for the motor of joint 4 before and after tampering is reported in Figure 8.3a, where the two signals have been overlaid to facilitate comparison. A damped oscillation induced by backlash is visible on the motor speed signal recorded after tampering. The second expected oscillation, the one with the inverted sign, is not present due to mechanical constraints restricting the axis movement to less than a full 360 deg rotation.

The speed signal was windowed in the interval $t = [18.99; 20.11]s$ and processed with the estimation algorithm to obtain a value for the amplitude of the backlash disturbance. The test was run 30 times to collect a statistic of the outcome, mean value and standard deviation for the A parameter are reported in the table in Figure 8.3b.

A similar behaviour was found for joint5, refer to Figure 8.4a. Here the backlash disturbance was less clear as the speed signal was more noisy. However also in this case, the algorithm succeeded in estimating the backlash gap increase.

Finally, backlash was observable also on joint 6, see Figure 8.5a. In this case the effect of backlash is most noticeable. Results of the A amplitude estimation provided by the algorithm for joint6 are listed in the table under the figure.



(a) Motor speed signal before and after the tampering. Backlash effect is visible.

Data	Expected value for A [rps]	Experimental value of A [rps]	
		mean	stdev
NO backlash	0.07	0.101	0.025
With backlash	0.17	0.209	0.030

(b) Results provided by the estimation algorithm on 30 runs.

Fig. 8.3 Analysis for joint 4 of the robot. Signal comparison (a) and backlash estimation results (b).

Unfortunately, in joints 2 and 3 of the robot the backlash in the encoder signal was not observable, even by changing the type of joint movement. Signals from Joint 2 are shown in Figure 8.6 as an example.

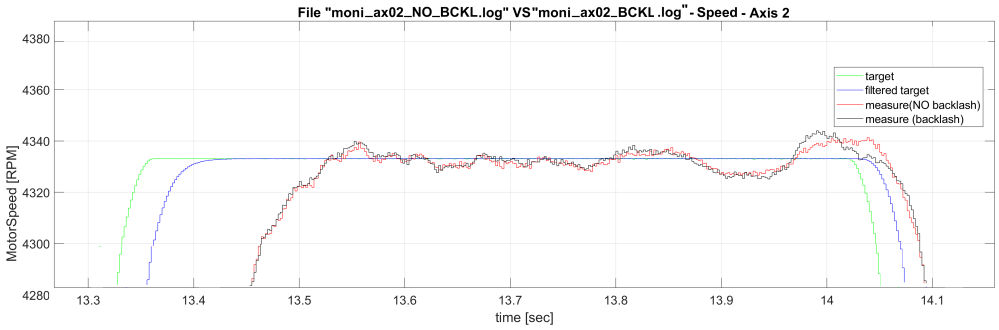
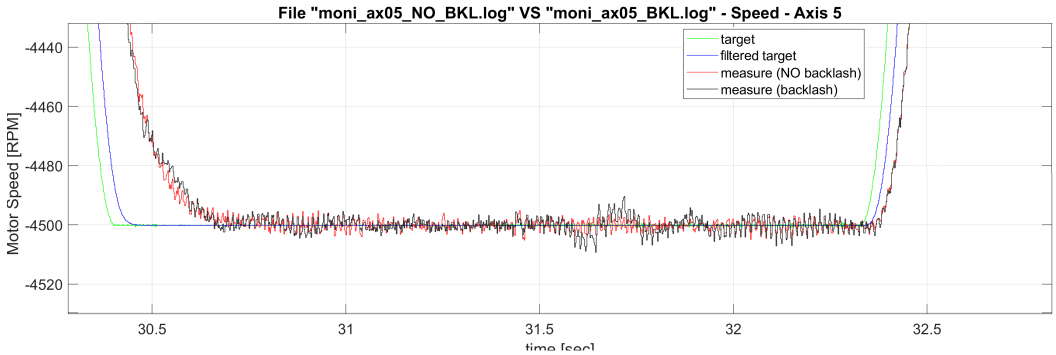


Fig. 8.6 Speed signal of joint 2 before and after the tampering. No backlash effect is visible.

The un-observability of the phenomenon on joint 2 and 3 can likely be attributed to two main factors:



(a) Motor speed signal before and after the tampering. Backlash effect is visible.

Data	Expected value for A [rps]	Experimental value of A [rps]	
		mean	stdev
NO backlash	0.05	0.063	0.026
With backlash	0.1	0.101	0.020

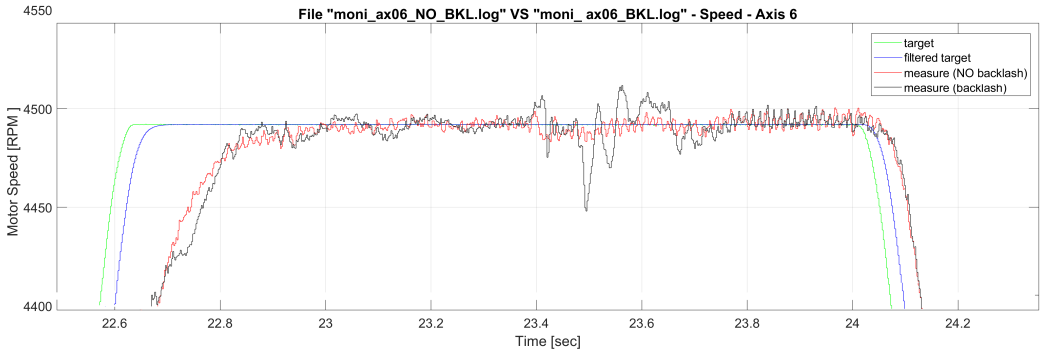
(b) Results provided by the estimation algorithm on 30 runs.

Fig. 8.4 Analysis for joint 5 of the robot. Signal comparison (a) and backlash estimation results (b).

- The impossibility of stimulating the joint in the most effective manner due to constraints imposed by the robot's mechanical structure (it is not possible to replicate the same movements performed on the test bench on robot joints 2 and 3),
- The inability to induce an actual backlash disturbance through tampering, given the particular mechanical structure of the reduction block.

Finally, a separate consideration must be made for joint 1. Specifically, the movement of joint 1 is unaffected by gravity since the rotation axis of the joint is parallel to the direction of gravity. Consequently, the stimulus that typically triggers the manifestation of backlash in the joint, as identified in this research work, is absent.

In summary, and for an easier comparison, an overview of the backlash state in the various analyzed signals is provided in the Figure 8.7



(a) Motor speed signal before and after the tampering. Backlash effect is visible.

Data	Expected value for A [rps]	Experimental value of A [rps]	
		mean	stdev
NO backlash	0.1	0.095	0.026
With backlash	0.7	0.520	0.028

(b) Results provided by the estimation algorithm on 30 runs.

Fig. 8.5 Analysis for joint 6 of the robot. Signal comparison (a) and backlash estimation results (b).

8.4 Automatic test procedure

Once the test cycles were assessed and the method has been proven applicable on the robot, an automatic procedure was developed to make data acquisition phase as easy and fast as possible. It was designed as a guided procedure for a non-expert user, displaying the necessary steps on the teach panel of the robot. The idea was to create a user-friendly tool for collecting data from robots, suitable for any robot user without requiring the help of expert technicians.

Moreover, the procedure was designed to minimize the time required for data acquisition. The shorter the time a working robot in a manufacturing plant is stopped, the lower the economic losses for the company. Therefore, the procedure includes two different test modalities: the fast modality and the full modality. The former only takes a few minutes to be performed and collects the minimum amount of data necessary for the backlash analysis; the latter takes longer and collects a wider set of data. At the end of the procedure, one or more moni.log files for each joint under test are recorded. These files are the input for the analysis algorithm that is performed offline at a later.

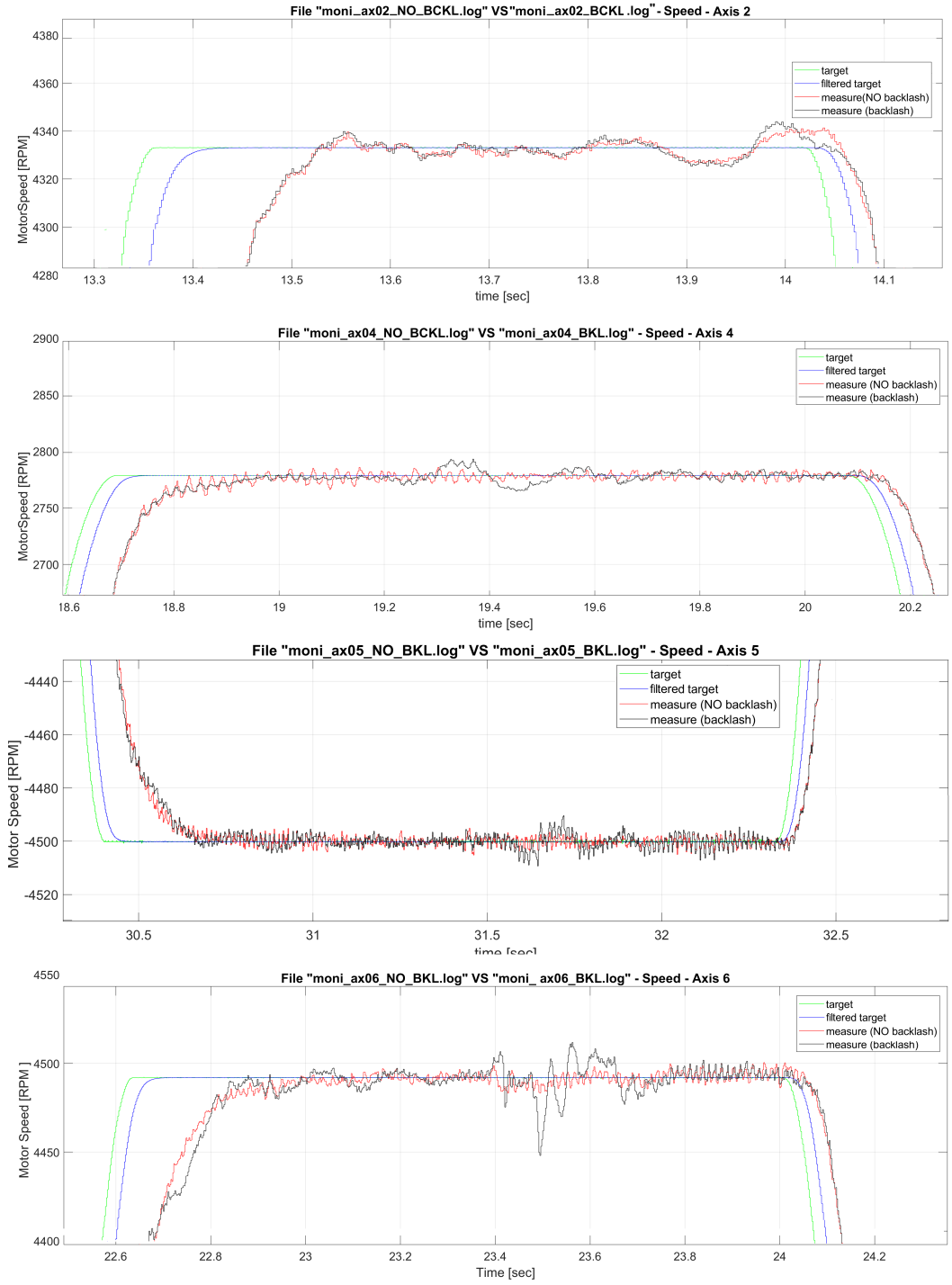


Fig. 8.7 Comparative view of the backlash state before and after tampering on the analyzed robot axes.

Chapter 9

Conclusion

This study aimed to investigate the possibility of measuring backlash in robotic joints by using the motor encoder only. This request arose from an industrial need to perform automatic evaluations of the backlash state in both newly manufactured and already deployed industrial manipulators. This was envisioned as part of an initiative to develop a tool for predictive maintenance of robots. To address this demand, the research activity provided an opportunity to gain a deeper understanding of backlash in the joints of an industrial manipulator and laid the foundation for designing a framework for automatic backlash analysis to be implemented in COMAU's IIoT platform.

Beginning with the analysis of a simple real system, a test bench with a single robotic joint specifically designed for this purpose, it was possible to identify test conditions that accentuate the effects of backlash in the transmission and make them visible on the motor encoder. This outcome, achievable in literature only on very simple test systems, demonstrated the potential applicability of the developed method to more complex kinematic structures, such as those found in an industrial robot. Among the various test movements used, one was identified as suitable for signal acquisition and analysis. This movement represents an "unusual" circumstance for observing backlash, entirely new compared to the classical observation made during direction reversals. In the proposed methodology, the backlash effect is triggered during continuous and constant-speed motion, with the rotation axis of the link arranged orthogonally to the force of gravity. The effects of gravity precisely trigger the opening and closing of the backlash gap.

In such circumstances, it was observed that the backlash effect is akin to a percussive phenomenon that propagates as an oscillation along the mechanical structure of the transmission until it reaches the motor encoder. Here, on the motor speed signal, it appears as a characteristic-shaped disturbance superimposed on the main signal. Since the amplitude of this disturbance is very small, and as disturbances of various natures, including noise, appear on the same signal, confidently recognizing the backlash effect becomes challenging. It was therefore useful to resort to defining a mathematical model that describes the peculiar characteristics of the backlash disturbance, which could be used as a "backlash signature." By searching for and recognizing this signature in the motor speed signal, backlash in the joint can be identified.

Through a specially developed Matlab/Simulink simulator, it was possible to characterize the evolution of this signature with varying backlash amplitude. It was observed that the amplitude of the oscillation is directly proportional to the amplitude of the backlash. By appropriately parameterizing the mathematical model of the backlash disturbance, the backlash estimation problem could be reduced to fitting a model to a signal. Using the fitting error as a cost function to minimize, the backlash estimation problem was transformed into an optimization problem. The tool for solving this problem was identified in an algorithm based on an evolutionary strategy, which proved more suitable than other meta-heuristics tested for solving the problem. The use of a meta-heuristic was necessary due to the non-convex nature of the problem. Several tests conducted on both simulated and real data recorded on the single-joint test bench have allowed verifying the effectiveness of the method. A final validation step was carried out during the generalization of the method from the single-joint case to the case of a complete industrial manipulator. In this latter case, it was observed that the method is still valid, albeit with some restrictions. The conducted tests demonstrated its applicability to the robot's wrist joints (joints 4, 5, and 6) but not to joints 1, 2, and 3. The reasons for this are multiple and involve both the working conditions of the joints, constraints of the robot's kinematic structure preventing the implementation of necessary test conditions, and the mechanical characteristics of the components used.

There are many aspects of the method that can and should be improved for effective industrial implementation. Firstly, the method requires a broader testing activity, including the comparison of results obtained with a directly measured backlash value on the robot. The tests on real data conducted during the research

remained at the level of estimating the "fault indicator" value (the *A* parameter of the model) without reaching the estimation of the corresponding backlash value. This is due to the lack of direct measurements on real systems.

Even the model of the backlash signature could be revised in light of the results obtained on the robot. The experience gained during the tests has shown that in the case of a complete robot, it is difficult to implement test movements that are wide enough to fully replicate the test conditions and, therefore, the characteristics of the disturbance model identified on the test bench. In robotic joints, it is more likely to detect only a part of this model, which could be simplified by reducing the number of considered damped oscillations.

As a consequence, the optimization algorithm could also be reconsidered, considering that simplifying the model would reduce the number of parameters to identify and, consequently, the size of the optimization problem.

Furthermore, the method should be made more robust because, as demonstrated by tests on a complete robot, the traces indicating the presence of backlash are lighter in the robot compared to those observed on the bench with a single joint.

Despite these aspects that still need improvement, the identified procedure has proven effective in detecting a progression of the backlash level in the joint. Through subsequent comparisons of evaluations obtained over time on the same robot, it is possible to derive information about the magnitude of the backlash change in the joint. Starting from simple motion tests on the robot, the method is indeed capable of recognizing and quantifying the increase in backlash over time. However, to provide an absolute estimate of the backlash value in the joint, an additional calibration phase is required, which has not yet been executed due to the limited amount of available data. This phase can be completed when a database of historical robot data, collected on robots “in the field ”or eventually generated through simulation, is available. The idea is to start making the current version of the tool accessible through the IIoT platform to begin monitoring robots at production sites and collecting data. COMAU collaborates with several Stellantis plants worldwide, where robots are employed in car production lines. In some of these plants, COMAU’s IIoT platform, named InGrid, is installed to collect, analyze, and visualize data from various industrial assets, including robots. With data collected on working robots a dedicated database can be created and used to refine the method providing a more detailed backlash assessments. The entire phase related to the automation of the data input collection

procedure for the test has already been completed, a guided procedure was developed and is currently available to allows the user to easily perform test motion cycles on the robot.

References

[1] Rosmaini Ahmad and Shahrul Kamaruddin. "An overview of time-based and condition-based maintenance in industrial application". In: *Computers & Industrial Engineering* 63.1 (2012), pp. 135–149. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2012.02.002. URL: https://www.sciencedirect.com/science/article/pii/S0360835212000484.

[2] A. Auger and N. Hansen. "A restart CMA evolution strategy with increasing population size". In: *Proc. IEEE Congress Evolutionary Computation*. Vol. 2. 2005, pp. 1769–1776. DOI: 10.1109/CEC.2005.1554902.

[3] Piotr Barosz, Grzegorz Gołda, and Adrian Kampa. "Efficiency Analysis of Manufacturing Line with Industrial Robots and Human Operators". In: *Applied Sciences* 10 (Apr. 2020), p. 2862. DOI: 10.3390/app10082862.

[4] S. Beinke et al. "Identification of nonlinear two-mass systems for self-commissioning speed control of electrical drives". In: *IECON '98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.98CH36200)*. Vol. 4. 1998, 2251–2256 vol.4. DOI: 10.1109/IECON.1998.724071.

[5] André Bittencourt, Kari Saarinen, and Shiva Sander. "A Data-Driven Method for Monitoring Systems that Operate Repetitively - Applications to Wear Monitoring in an Industrial Robot Joint". In: Jan. 2012.

[6] Tawfik Borgi et al. "Data analytics for predictive maintenance of industrial robots". In: *2017 International Conference on Advanced Systems and Electric Technologies (IC ASET)*. 2017, pp. 412–417. DOI: 10.1109/ASET.2017.7983729.

[7] Alexandros Bousdekis, Dimitris Apostolou, and Gregoris Mentzas. "Predictive Maintenance in the 4th Industrial Revolution: Benefits, Business Opportunities, and Managerial Implications". In: *IEEE Engineering Management Review* 48 (Mar. 2020), pp. 57–62. DOI: 10.1109/EMR.2019.2958037.

[8] R. V. Canfield. "Cost Optimization of Periodic Preventive Maintenance". In: *IEEE Transactions on Reliability* 35.1 (1986), pp. 78–81. DOI: 10.1109/TR.1986.4335355.

[9] Jie Chen. "Robust residual generation for model-based fault diagnosis of dynamic systems." In: 1995.

[10] Krzysztof M Choromanski et al. "From complexity to simplicity: Adaptive es-active subspaces for blackbox optimization". In: *Advances in Neural Information Processing Systems* 32 (2019).

[11] Nicholas G. Dagalakis and Donald R. Myers. "Adjustment of Robot Joint Gear Backlash Using the Robot Joint Test Excitation Technique". In: *The International Journal of Robotics Research* 4.2 (1985), pp. 65–79. DOI: 10.1177/027836498500400206.

[12] Salih O. Duffuaa et al. "A generic conceptual simulation model for maintenance systems". In: *Journal of Quality in Maintenance Engineering* 7 (2001), pp. 207–219.

[13] Farzam Farbiz, Yuan Miaolong, and Zhou Yu. "A Cognitive Analytics based Approach for Machine Health Monitoring, Anomaly Detection, and Predictive Maintenance". In: *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. 2020, pp. 1104–1109. DOI: 10.1109/ICIEA48937.2020.9248409.

[14] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

[15] W.E. Forsthoffer. "11 - Preventive and Predictive Maintenance Best Practices". In: *Forsthoffer's Best Practice Handbook for Rotating Machinery*. Ed. by W.E. Forsthoffer. Boston: Butterworth-Heinemann, 2011, pp. 563–576. ISBN: 978-0-08-096676-2. DOI: https://doi.org/10.1016/B978-0-08-096676-2.10011-6. URL: https://www.sciencedirect.com/science/article/pii/B9780080966762100116.

[16] Roy Fritzsche and Rainer Lasch. "An Integrated Logistics Model of Spare Parts Maintenance Planning within the Aviation Industry". In: *International Journal of Economics and Management Engineering* 6.8 (2012), pp. 1958–1967. ISSN: eISSN: 1307-6892. URL: https://publications.waset.org/vol/68.

[17] Dirk Gebler and Joachim Holtz. "Identification and compensation of gear backlash without output position sensor in high-precision servo systems". In: *IECON'98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (Cat. No. 98CH36200)*. Vol. 2. IEEE. 1998, pp. 662–666.

[18] Maria Holgado Granados, Marco Macchi, and Luca Fumagalli. "Maintenance Business Model: a concept for driving performance improvement". In: 2015.

[19] Nikolaus Hansen, Sibylle D. Müller, and Petros Koumoutsakos. "Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES)". In: *Evolutionary Computation* 11 (2003), pp. 1–18. URL: https://api.semanticscholar.org/CorpusID:261944074.

[20] Nikolaus Hansen and Andreas Ostermeier. "Completely Derandomized Self-Adaptation in Evolution Strategies". In: *Evolutionary Computation* 9 (2001), pp. 159–195.

[21] Nikolaus Hansen, Andreas Ostermeier, and Andreas Gawelczyk. "On the Adaptation of Arbitrary Normal Mutation Distributions in Evolution Strategies: The Generating Set Adaptation". In: *Proceedings of the Sixth International Conference on Genetic Algorithms* (Jan. 1997).

[22] Geir Hovland et al. “Nonlinear identification of backlash in robot transmissions”. In: *Proceedings of the 33rd ISR (International Symposium on Robotics)*. Citeseer. 2002, pp. 1–6.

[23] British Standards Institution. *Maintenance Terminology*. British Standards Institution, 2001.

[24] Alaa Abdulhady Jaber and Robert Bicker. “Development of a Condition Monitoring Algorithm for Industrial Robots based on Artificial Intelligence and Signal Processing Techniques.” In: *International Journal of Electrical & Computer Engineering (2088-8708)* 8.2 (2018).

[25] Alaa Abdulhady Jaber and Robert Bicker. “Industrial robot backlash fault diagnosis based on discrete wavelet transform and artificial neural network”. In: *American Journal of Mechanical Engineering* 4.1 (2016), pp. 21–31.

[26] Japanese-Standards-Association. *JIS-B-1705:2013 Backlash For Bevel Gears, Japanese Industrial Standard*. en, ja. Standard JIS B 1705:2013. International Organization for Standardization, 2013.

[27] R. Jiang. “Optimization of alarm threshold and sequential inspection scheme”. In: *Reliability Engineering & System Safety* 95.3 (2010), pp. 208–215. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2009.09.012. URL: https://www.sciencedirect.com/science/article/pii/S0951832009002324.

[28] Tomas Kubela, Ales Pochyly, and Vladislav Singule. “Assessment of industrial robots accuracy in relation to accuracy improvement in machining processes”. In: *2016 IEEE International Power Electronics and Motion Control Conference (PEMC)*. 2016, pp. 720–725. DOI: 10.1109/EPEPEMC.2016.7752083.

[29] Adam Lagerberg and Bo Egardt. “Backlash Estimation With Application to Automotive Powertrains”. In: *IEEE Transactions on Control Systems Technology* 15.3 (2007), pp. 483–493. DOI: 10.1109/TCST.2007.894643.

[30] Ming Li et al. “On-line measurement method of transmission backlash based on angular velocity and double-end angular position information”. In: *Measurement and Control* 54.1-2 (2021), pp. 65–72. DOI: 10.1177/0020294020973252.

[31] Miguel F.M. Lima, J.A. Tenreiro Machado, and Manuel Crisóstomo. “Filtering method in backlash phenomena analysis”. In: *Mathematical and Computer Modelling* 49.7 (2009), pp. 1494–1503. ISSN: 0895-7177. DOI: https://doi.org/10.1016/j.mcm.2008.08.015. URL: https://www.sciencedirect.com/science/article/pii/S0895717708003063.

[32] Marco Macchi, Irene Roda, and Luca Fumagalli. “On the Advancement of Maintenance Management Towards Smart Maintenance in Manufacturing”. In: *APMS*. 2017.

[33] Muhamad Mansor, A. Ohsato, and Shamsuddin Sulaiman. “Knowledge Management for Maintenance Activities in the Manufacturing Sector”. In: *International Journal of Automotive and Mechanical Engineering* 5 (June 2012), pp. 612–621. DOI: 10.15282/ijame.5.2012.7.0048.

[34] Lőrinc Márton and Béla Lantos. "Friction and backlash measurement and identification method for robotic arms". In: *2009 International Conference on Advanced Robotics* (2009), pp. 1–6.

[35] R. Merzouki, J. C. Cadiou, and N. K M'Sirdi. "Compensation of backlash effects in an electrical actuator". In: *2003 European Control Conference (ECC)*. 2003, pp. 2511–2516. DOI: 10.23919/ECC.2003.7085343.

[36] Seyed Mohammad Mirjalili. "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems". In: *Neural Computing and Applications* 27 (2016), pp. 1053–1073. URL: https://api.semanticscholar.org/CorpusID:207018626.

[37] Seyedali Mirjalili. "The Ant Lion Optimizer". In: *Advances in Engineering Software* 83 (2015), pp. 80–98. ISSN: 0965-9978. DOI: https://doi.org/10.1016/j.advengsoft.2015.01.010. URL: https://www.sciencedirect.com/science/article/pii/S0965997815000113.

[38] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey Wolf Optimizer". In: *Advances in Engineering Software* 69 (2014), pp. 46–61. ISSN: 0965-9978. DOI: https://doi.org/10.1016/j.advengsoft.2013.12.007. URL: https://www.sciencedirect.com/science/article/pii/S0965997813001853.

[39] Seyedeh Zahra Mirjalili et al. "Grasshopper optimization algorithm for multi-objective optimization problems". In: *Applied Intelligence* 48 (Apr. 2018). DOI: 10.1007/s10489-017-1019-8.

[40] R Keith Mobley. "39 - Gears and Gearboxes". In: *Plant Engineers Handbook*. Ed. by R. Keith Mobley. Woburn: Butterworth-Heinemann, 2001, pp. 629–637. ISBN: 978-0-7506-7328-0. DOI: https://doi.org/10.1016/B978-075067328-0/50041-0.

[41] R Keith Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.

[42] R.K. Mobley. *Maintenance fundamental*. Burlington, MA, USA: Elsevier Butterworth–Heinemann, 2004.

[43] M. Nordin, J. Galic, and P. Gutman. "New models for backlash and gear play". In: *International Journal of Adaptive Control and Signal Processing* 11 (1997), pp. 49–63.

[44] Mattias Nordin and Per-Olof Gutman. "Controlling mechanical systems with backlash - A survey". In: *Automatica* 38 (Oct. 2002), pp. 1633–1649. DOI: 10.1016/S0005-1098(02)00047-X.

[45] Dimitrios Papageorgiou et al. "Backlash estimation for industrial drive-train systems". In: *IFAC-PapersOnLine* 50.1 (2017). 20th IFAC World Congress, pp. 3281–3286. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2017.08.621. URL: https://www.sciencedirect.com/science/article/pii/S2405896317310005.

[46] Dimitrios Papageorgiou et al. "Robust Backlash Estimation for Industrial Drive-Train Systems—Theory and Validation". In: *IEEE Transactions on Control Systems Technology* 27.5 (2019), pp. 1847–1861. DOI: 10.1109/TCST.2018.2837642.

[47] Yongyi Ran et al. “A Survey of Predictive Maintenance: Systems, Purposes and Approaches”. In: *ArXiv* abs/1912.07383 (2019).

[48] C. Raynolds. “Flocks, Herds, and Schools: A Distributed Behavioral Model”. In: *Computer Graphic- ACM SIGGRAPH ’87 Conference Proceedings* 21.4 (1987), pp. 25–34.

[49] C. Raynolds. “Steering behaviour for autonomous characters”. In: *http://www.red3d.com/cwr/steer/, first version* (1999).

[50] N. Sarkar, R.E. Ellis, and T.N. Moore. “BACKLASH DETECTION IN GEARED MECHANISMS: MODELING, SIMULATION, AND EXPERIMENTATION”. In: *Mechanical Systems and Signal Processing* 11.3 (1997), pp. 391–408. ISSN: 0888-3270. DOI: https://doi.org/10.1006/mssp.1996.0082.

[51] Nikolai V Smirnov. “On the estimation of the discrepancy between empirical curves of distribution for two independent samples”. In: *Bull. Math. Univ. Moscou* 2.2 (1939), pp. 3–14.

[52] J. Stein and Churn-Hway Wang. “Automatic detection of clearance in mechanical systems: theory and simulation”. In: *Proceedings of 1995 American Control Conference - ACC’95* 3 (1995), 1737–1745 vol.3.

[53] Jeffrey L. Stein and Churn-Hway Wang. “Automatic detection of clearance in mechanical systems: experimental validation”. In: *Mechanical Systems and Signal Processing* 10.4 (1996), pp. 395–412. ISSN: 0888-3270. DOI: https://doi.org/10.1006/mssp.1996.0028. URL: https://www.sciencedirect.com/science/article/pii/S088832709690028X.

[54] Jeffrey L. Stein and Churn-Hway Wang. “Estimation of Gear Backlash: Theory and Simulation”. In: *Journal of Dynamic Systems, Measurement, and Control* 120.1 (Mar. 1998), pp. 74–82. ISSN: 1528-9028. DOI: 10.1115/1.2801324. URL: http://dx.doi.org/10.1115/1.2801324.

[55] Paweł Szynkiewicz. “Comparative study of PSO and CMA-ES algorithms on black-box optimization benchmarks”. In: *Journal of Telecommunications and Information Technology* (2018).

[56] Sebastian Villwock and Mario Pacas. “Time-domain identification method for detecting mechanical backlash in electrical drives”. In: *IEEE Transactions on Industrial Electronics* 56.2 (2008), pp. 568–573.

[57] Stephen J. Walsh and John J. Borkowski. *Fast Computation of Highly G-optimal Exact Designs via Particle Swarm Optimization*. 2022. arXiv: 2206.06498 [stat.CO].

[58] Shota Yamada, Hiroshi Fujimoto, and Yuki Terada. “Joint torque control for backlash compensation in two-inertia system”. In: *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*. 2016, pp. 1138–1143. DOI: 10.1109/ISIE.2016.7745054.

[59] Ming Yang et al. “Study of on-line backlash identification for PMSM servo system”. In: *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*. 2012, pp. 2036–2042. DOI: 10.1109/IECON.2012.6388745.

[60] Ge Zhang. “Speed control of two-inertia system by PI/PID control”. In: *Industrial Electronics, IEEE Transactions on* 47 (July 2000), pp. 603–609. DOI: 10.1109/41.847901.