Institute of Thermal Turbomachinery
and Machine Dynamics

INSTITUTE OF THERMAL TURBOMACHINERY
AND MACHINE DYNAMICS
GRAZ UNIVERSITY OF TECHNOLOGY

Master's Thesis in Mechanical Engineering

Model-based development of a condition monitoring system for turbomachinery application

Stefan Baueregger BSc

Student ID: 01130804

Summer Term 2018

Head of Institute: Univ.-Prof. Dr.-Ing. Franz HEITMEIR

Supervisor: Dipl.-Ing. Dr.techn. Andreas MARN

Graz, on October 1st, 2018

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen und Hilfsmitteln wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz am 1. Oktober 2018

. . . .

(Unterschrift des Verfassers)

Abstract

Title: Model-based development of a condition monitoring system for turbomachinery application

Author: Stefan Baueregger

1^{st} keyword: turbomachinery
2^{nd} keyword: clearance
3^{rd} keyword: measurement
4^{th} keyword: model-based

The everlasting goal to reduce fuel consumption by increasing efficiency of turbomachinery, paired with the possibilities given by the improvements in computational power and numerical methods in use, has led to the realization of compact machine designs. This progress was accompanied by the necessity of monitoring operational machines for performance and safety reasons. While the present thesis aims to develop said condition monitoring system, the focus is limited to the aspect of radial rotor-casing clearance detection and the resulting processing possibilities. Generally speaking, a monitoring system consists of two main components: data acquisition and data processing. Due to the lack of available hardware and unclear requirements for turbomachinery application, the former is replaced by a model implemented in Matlab, dealing with the simulation of signals. The latter is realized within LabView, with the goal to derive the information carried by the generated signals.

The underlying document initially presents the results of a fundamental research regarding the topic of clearance measurement, alongside the most important information that is necessary to comprehend the design decisions made throughout the development stages. This is followed by the introduction of the Matlab routine that uses the two-dimensional model of a turbomachine together with the capacitive sensing principle for the generation of clearance signals. Building up on this model, the next section shows the initially mentioned processing unit that is designed to be applicable in real-world monitoring, by verifying it based on simulated signals. The entire development environment is subsequently used to determine influential parameters and system requirements for the acquisition of a measurement chain. Apart from the sensing principle and its limitations, the necessary sensor setup, sample rates and signal enhancement tools are evaluated and presented.

The thesis ultimately concludes with the presentation of the full range of performance of the developed tip clearance monitoring system, which proves to be very situational due to the model-based approach. Nevertheless, the results of processed data show promising accuracy in determining tip clearance and approximating shaft movement, so that the software environment is deemed ready for real-world testing.

Table of Contents

List of Figures

List of Tables

1 Introduction and Problem Definition

Since the beginning of modern day turbomachinery design at the end of the 19^{th} century with the development of the steam turbine, the coherent purpose of industrial research has been to reduce fuel consumption as well as CO_2 emission, or in other words, to optimize stage performance. Initially, the approaches taken by various inventors were mainly based on alternating process sequences. While Sir Charles Parson had chosen to realize the reaction turbine, his college Karl Gustaf de Laval invented the impulse turbine. The main difference between the two variants are that the former makes use of pressure and kinetic energy within the rotor while the latter only converts kinetic energy in the impeller. All of these basic principles were refined throughout the first half of the 20^{th} century, using better understanding of fluid mechanics and enhanced mathematical solutions of the governing equations, which ultimately led to the successful design of turbo compressors. What remained was the everlasting strive for optimization of fluid flow and loss reduction of the entire turbomachinery domain. At the second half of the 20^{th} century this trend has been greatly accelerated by the invention of the computer that allowed for the solution of more sophisticated mathematical models. Progressively, the development of numerical solvers towards the end of the century has peaked in the modern day understanding of this machine type.

In recent years, the standards in accuracy of numerical simulations have allowed for more and more compact and three dimensional design, while still retaining the required operational stability. It has become possible to predict undisturbed flow fairly accurate, but the need to account for loss mechanisms has motivated extensive research being conducted in order to refine the problem areas. One of the main contributors to the overall loss was identified within the area of the end-wall. Due to the nature of the machine, the moving rotor contained within the stationary casing requires for a radial clearance between the two components, allowing for a fraction of the fluid to pass above the rotor instead of through the passage between the blades. This so called 'tip leakage flow' ultimately enters a mixing process with the main flow, which is accountable for an increase in entropy, translatable to loss. Caused by this phenomenon, manufacturers have come up with a variety of approaches to enhance performance, but there has not yet been established a coherent ideal design. The general idea has consistently been to reduce the magnitude of the radial clearance between the rotor and the casing, which helps to reduce the secondary flow and thereby the losses. Even though modern day manufacturing methods provide all the necessary tools for decreasing this gap to a fraction of a millimeter, the dynamics of the operational turbomachine are known to be responsible for the real limits. While the clearance at the stationary state of the machine can range to multiples of 0.1 mm that margin is reduced greatly in the operational condition. This is mainly caused by three effects. The first challenge becomes apparent when looking at rotordynamics. A variety of influential factors such as unbalanced masses, gravity or other external forces cause the displacement of the spinning rotor within the bearings and thereby change the clearance. Furthermore, the centrifugal forces cause extension of the blades, which subtracts from the already reduced margin. Ultimately, the remaining clearance is influenced by thermal expansion. Compared to the stationary, cold state the dimension of every machine component that is affected by the temperature changes its dimensions. Since rotor and casing are not always affected to the same extent, a further reduction of the radial clearance is possible. If all of these mechanisms add up to a value greater than the available clearance, the rotor contacts the casing, which might lead to the malfunction of the blade or even the entire machine and therefore has to be avoided.

The just mentioned trade off between improved performance through reduction of tip leakage flow at the risk of contact between rotor and casing has been subject to detailed research. The introduction of many design principles to address this problem, coupled with the strive to reduce the clearance, went alongside the necessity of constant monitoring of the actual condition. It has become state of the art to monitor operational machines in order to ensure early detection of failures and to improve decision making concerning safety of operation. The advantage of constant availability of information has outweighed the considerable costs of measurement systems, due to the fact that delaying remission in a safe manner potentially safes tremendous amounts of resources. Nevertheless, this trend has not been started in the industrial sector but rather in research facilities. There, the development and testing of various different machine design features required for the validation of conducted computational fluid dynamics simulations.

As of today, a variety of measurement systems are available for that purpose. One example is the 'CAPACISENSE' by Pentair that allows for the on-line measurement of a variety of machine parameters using capacitive sensing. The functionally usually depends on the purchased package. The options provided by Pentair within their '5 Series' of the system include channels for:

- Raw Blade Passing Signal

- Average Blade Clearance
- Blade by Blade Clearance
- Blade Tip Timing

measurement. The client has to decide which data is vital for the machine monitoring in the individual case and which packages are purchased. While this approach might be a viable solution for industrial use, where the system is installed on a turbomachine with a lifespan of 20 years or more, the main problems for research purposes are the costs, the flexibility and the accessibility. In a research facility testing cycles are usually limited to a few months and demands for data can vary from project to project. This is why purchasing an integrated module with specified measurement channels at a high price can be infeasible. Furthermore, the quality of measurement is often coupled with the understanding of the underlying mathematical models of the systems. In case of the mentioned devices, the client is only provided with the final results of the calculations.

The disadvantages of purchasing a closed system for the usage in a research facility has led to the underlying project. The Institute of Thermal Turbomachinery and Machine Dynamics requires a radial rotor-casing clearance monitoring system for their test stand in order to validate simulation results and to increase significance of conducted research. This need was the cause for the start of a project that has been laid out consisting of the following steps:

1. Create Tip Clearance Monitoring Software
2. Design of the Measurement Hardware
3. Acquisition of the Measurement Hardware
4. Installation of the Measurement Hardware on the Machine
5. Validation of Measurement Accuracy

This thesis is part of the project, dealing with the first step of the mentioned plan. The creation of a monitoring software leads the project in order to gather fundamental understanding of the domain and to derive basic hardware requirements as a result of machine behavior. For this purpose the software package LabView shall be used due to its abilities in data processing and availability. The resulting clearance monitoring system has to be able to process signals originating in measurements conducted on axial thermal turbomachinery and return information to a user. The desired measurement data is:

- Clearance between Sensor and Rotor
- Shaft Location within the Casing

- Minimum Clearance between Rotor and Casing

In order to derive exact system requirements as well as influential parameters on the monitoring software without needing to operate actual turbomachinery, the development shall be based on a mathematical model of the measurement signal. For the creation of this signal an additional piece of software shall be handled by Matlab and result in plausible clearance data that can be processed by the monitoring system. Therefore, the steps within this thesis serving the purpose of completing step one in the project are:

1. Creation of a Matlab model of the machine and necessary measurement devices with the goal of simulating measurement data
2. Development of the monitoring routine using LabView based on the simulated data
3. Analysis of influential parameters using this development environment

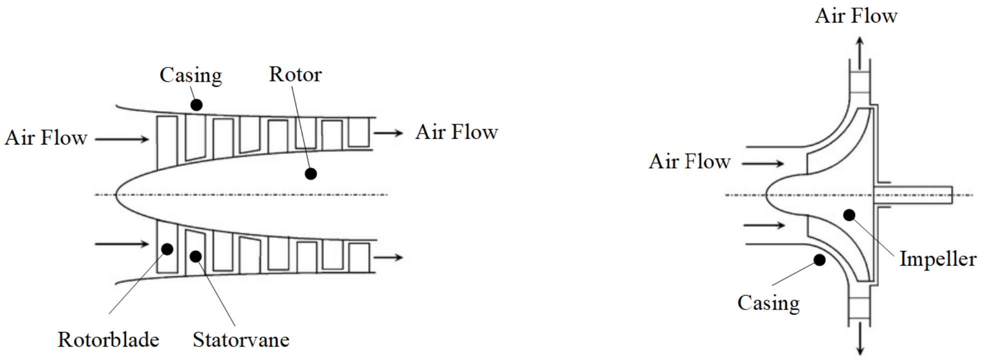
In order to guarantee applicability of the LabView routines, the models are developed using real-world measurement hardware and machine as a reference. First, a simplified, geometric abstraction of a rotor spinning within a casing is implemented. This includes the two dimensional positioning of the shaft and various geometries. Second, the interaction of these geometric replications with chosen key elements of the measurement chain is computed and the resulting data used as signal for the LabView routine. By basing all models on real-world principles, the simulated signal shape and behavior resembles actual measurement data so that future hardware can be implemented without major adaptation of the routines. Due to this model-based approach, the initial project costs are kept low, testing the clearance monitoring system is easy and gradual expansion of the LabView routines according to the institutes needs is possible.

2 Turbomachinery

The introduction states the intention of developing a Matlab model of machine and measurement chain that simulates radial rotor-casing clearance data. This is supposed to be followed by the creation of the clearance monitoring routine within LabView. In order to guarantee the applicability of the monitoring unit, the signal it is based on has to be created using comprehensive consideration of the underlying machine and the measurement elements involved. Therefore, this chapter starts by delimiting the machine type that shall be monitored. This is followed by the analysis of the problem area itself - the rotor-casing clearance - and the description of mechanisms that influence this parameter.

2.1 Machine Characterization

Generally speaking, a turbomachine can be classified by the operation of a spinning rotor within a stationary casing. Nevertheless, this statement is not conclusive for the machines the clearance measurement shall be conducted on, which is why a variety of attributes can be used to further categorize the target machine. First, the regime is divided into turbomachinery that uses energy transfered by a fluid flowing through blade cascades and turbomachinery that uses alternative means of energy conversion. For the underlying thesis, the former shall be treated exclusively. Second, within the fluid based working principle, the type of fluid and direction of energy transfer allows for additional distinctions. The primary attribute is the working fluid. On one side, if the fluid can be described as compressible, the machine classifies as a thermal turbomachine. On the other side, if the fluid can be seen as incompressible, a hydraulic turbomachine is described. The general design of thermal and hydraulic turbomachinery is comparable. The fluid enters either machine type on one side, passes one or more blade/vane rows and exits on the other side. Note, that in case of thermal turbomachinery the compressible nature of the fluid has to be respected by accounting for thermodynamics while solving the governing fluid mechanical problem. This step can be simplified when fluid behavior is near incompressible as is in the hydraulic case. Dependent on the direction of energy conversion another subdivision can be made. If the energy is delivered by the rotor and transfered to the fluid, hydraulic and thermal turbomachine are both called turbine. When the energy is provided by the fluid and transported to the rotor, either a pump (hydraulic) or a compressor (thermal) is at work. The classification process for enclosed machinery is displayed in figure 2.2 on the left side. Additionally, another characteristic can be provided that is based on the direction of the fluid flow and not on the type of fluid or the direction of energy conversion. If the fluid passes the machine following the axis of the shaft, it is categorized as axial turbomachinery. On the contrary, if the fluid flows into radial direction, it is categorized as a radial turbomachinery. Since this thesis is created in cooperation with the Institute of Thermal Turbomachinery and Machine Dynamics at the University of Technology in Graz and its results shall be applicable on axial, thermal turbomachinery, an example of this subtype is presented in figure 2.1. The images show the difference between radial and axial turbomachinery. Both machines are thermal turbo-compressors. This fact can be derived by the divergent fluid canals in flow direction that hint the compression of the working fluid.



(a) Axial compressor

(b) Radial (centrifugal) compressor

Figure 2.1: Comparison of axial and radial compressors (Source: Quora Website)

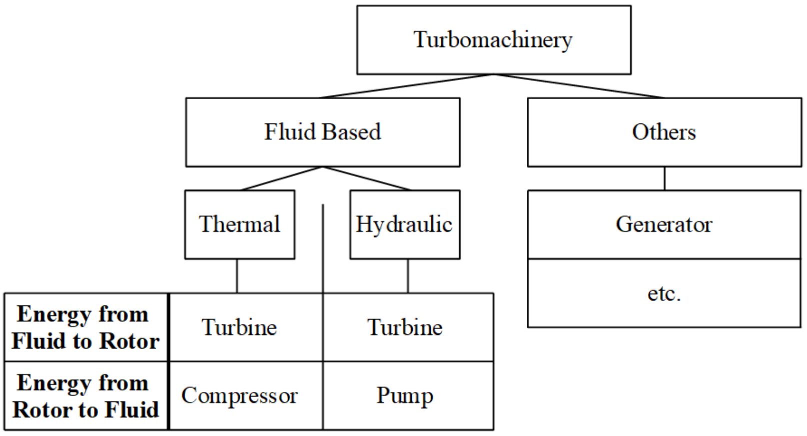


Figure 2.2: Classification of turbomachinery

2.2 Working Principle

The components of axial thermal turbomachinery are equal for both turbines and compressors. They consist of following main parts:

- stationary casing
- rotating shaft containing the rotor blades
- stator vanes that are connected to the casing
- bearings supporting the shaft within the casing

Figure 2.3 shows the simplified model of a stage of an axial thermal turbomachine. In case of turbines, the fluid flows from A to B and in case of compressors, the fluid flows from B to A. Note the conical shape of the duct. This is based on the fact that the fluid is either expanding or contracting, effectively changing the required volume. In case of hydraulic turbomachinery this design feature is absent in most cases since the density can be considered to be constant.

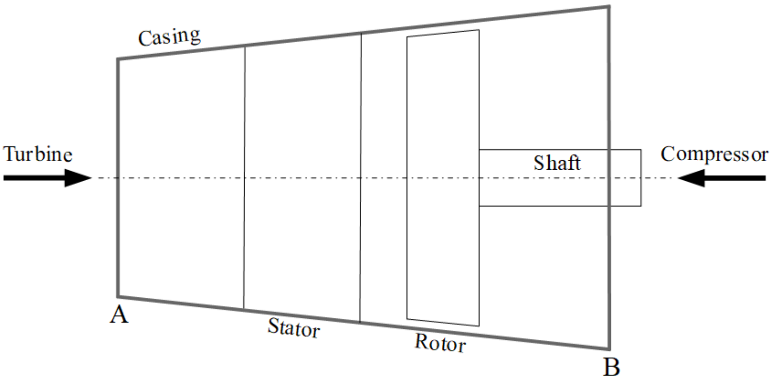


Figure 2.3: Simplified model of an axial turbomachine

Usually both rotor and stator are composed of blades. The stator's vanes are fixed within the casing while the rotor's blades are rooted to the shaft. This can happen either directly or by using a disk that is connected to the rotating shaft and thereby enabling the entire blade row to spin. A rotor made out of a shaft, disk and corresponding blades can be seen in figure 2.4.

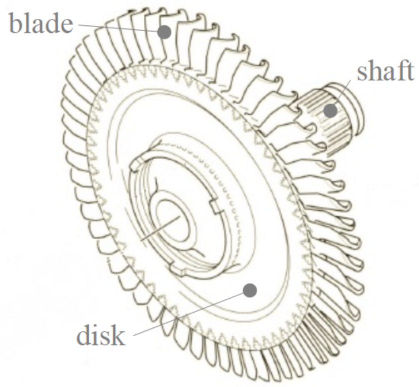


Figure 2.4: Model of a rotor composed of a disk and attached blades (Source: Aerossurance Website)

The basic idea that governs the energy transport between the rotor and the fluid in a thermal turbomachine can be derived from the lift that is present at an airfoil of infinite span. Imagine a large control volume according to figure 2.5. The fluid enters the control volume on the left and leaves it on the right. Within the entire control volume the net circulation is zero. When the fluid approaches the airfoil the flow is deflected upwards and downwards. The fluid proceeds to follow the contours which leads to it being directed downwards towards the end of the airfoil.

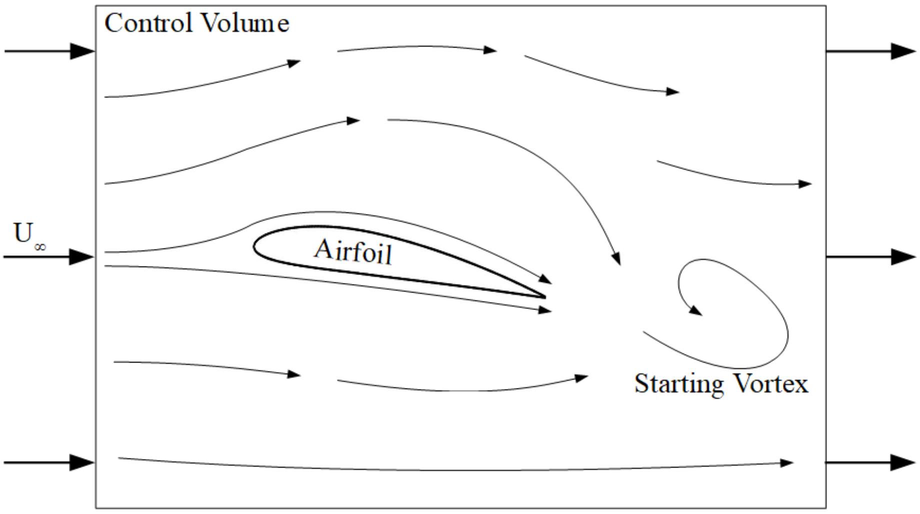


Figure 2.5: Airfoil within a control volume

The sum of all upward and downward motion around the object results in a clockwise circulation Γ, as is indicated in figure 2.6. This, in order for the net circulation within the control volume to be zero, leads to the formation of the so called 'starting vortex' behind the airfoil, which dissipates over time due to viscous losses. The phenomenon has been described by Kutta and Joukowsky, who linked a force F' on the airfoil that is oriented perpendicular to the flow direction to the clockwise circulation according to

$$\oint_C \vec{v} \cdot d\vec{s} = \Gamma \tag{2.1}$$

$$F' = \Gamma U_\infty \rho \tag{2.2}$$

Here, F' represents the resulting force per unit span, U_∞ the velocity of the fluid far away from any disturbance and ρ the density of the fluid. An explanation for the resulting force can be given using Bernoulli's stream line theory. In the portrayed case, the circulation Γ is oriented in the same direction as the fluid flow on the upper side of the airfoil and in the opposite direction on the lower side. This leads to the initial fluid velocity being increased above and decreased below the structure. When computing Bernoulli's equation for the streamline above and the streamline below, this leads to a higher pressure beneath and a lower pressure above the object. The pressure differences add up, together with the effective surface of the airfoil, to the upward force.

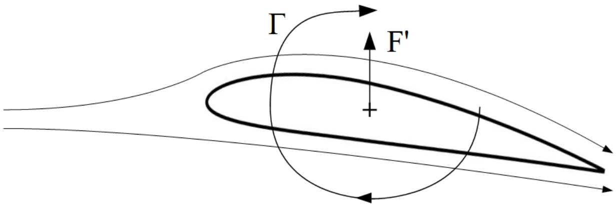


Figure 2.6: Circulation around an airfoil

This explanation can be adapted for describing energy transformation in turbomachinery. In case of a turbine rotor, a large number of these airfoils are mounted on the circumference of a shaft (compare figure 2.4). Note, that a blade is a finite variant of the just described airfoil. Therefore the same mechanisms apply. The lift on every blade results in torque on the shaft, which can then be transfered on to a generator for example. This is the case for turbines, where the fluid exerts power on the blades and spins the rotor. In a compressor the circumstances are reverted and therefore the direction of energy conversion changes. Note as well, that the finite blade length and compact machine design causes additional flow phenomena, especially in the area of boundaries like the tip and root of the blade. Additional information about fluid mechanics in turbomachinery can be found in respective literature, since this thesis will only deals with basic ideas that serve explanatory reasons.

Generally speaking, the energy can only be transfered if the fluid flow behaves in the way it has been described above. If the flow is disturbed and does not circulate around the profiles, no lift can exist. This is the reason for careful fluid dynamic computation of the entire machine during the design. As has been mentioned already, one of the areas where the flow does not follow the desired principles is in the vicinity of the blade's tip. The fluid does not just pass the intended passage but leaks over the tip equally. On the one hand, this leakage flow does not contribute to the creation of lift and on the other hand worsens the flow in the following regions. Thus the discussion of loss arises. Generally speaking, loss in thermal turbomachinery is defined by an increase in entropy due to certain fluid dynamic effects that can not be reversed or used for the intended energy conversion. According to Denton (1993), who has studied loss-mechanisms in this type of machine in detail, all of these entropy increasing mechanisms can be broken down to the following categories:

- Friction due to shear layers: boundary layer, wake flow, jet flow
- Heat transfer with the surrounding
- Shock waves

Analysis has lead to the identification of end-wall losses, profile losses itself and the tip leakage as main contributors to the overall loss. Most of the times the effects are highly interdependent, which is why they often have to be studied as a whole. As of today, the percentage of the energy that has actually been transfered between fluid and rotor within the rotor's cascade and the energy that could have been transfered ideally lies around 90%. Since a turbomachine is usually designed to be operational for multiple decades - sometimes even constantly - every fraction of a percent added to the energy conversion efficiencies of the rotor's cascade contributes to the economical and ecological win.

2.3 Rotor Clearance Leakage

As has been highlighted at various occasions throughout the last pages, the effects linked to leakage flow contribute considerably to the overall performance loss of the machine. Within this section the flow in the gap region will be analyzed and the loss mechanisms explained. This goes alongside with the introduction of various blade geometries that are a result of researchers trying to reduce the negative effects of the secondary flow.
First, leaving aside viscid fluid behavior, it can be said that the leakage flow of an unshrouded blade occurs from pressure side to suction side. This generally reduces the mass flow through the blade's passage, which leads to less fluid contributing to the intended energy transfer on the blade and a part of the realized pressure difference between the two sides of one blade being equalized. Additionally, the leakage flow over the blade causes a disturbance of the flow regime in the vicinity of the tip. This leads to a reduction in created lift in the area of the tip of the blade. Second, the modeling of viscid flow behavior leads to the detection of an increase in entropy, what can be explained using figure 2.7.

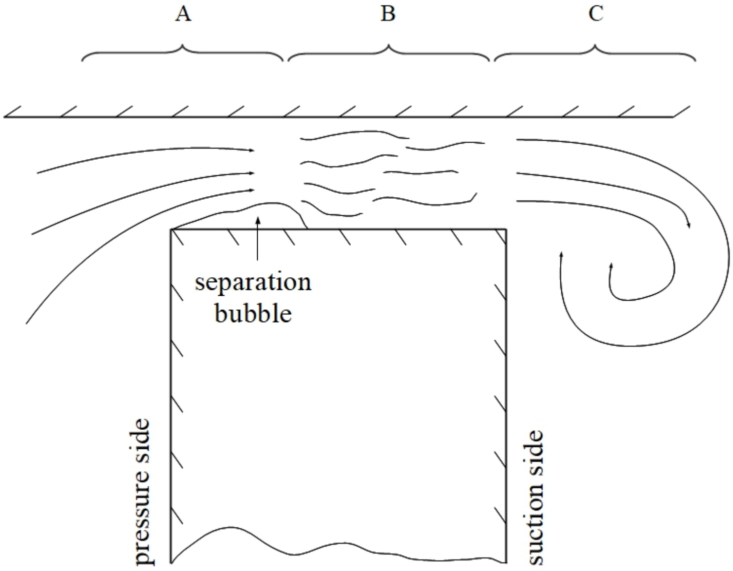


Figure 2.7: Leakage flow over an unshrouded blade by Denton (1993)

When entering the gap between rotor and casing, the flow is contracted to a jet and a separation bubble might form at the edge to the pressure side (figure 2.7 A). When traveling through the gap (figure 2.7 B) the entropy increases, caused by viscous shear between the separated flow layers of the casing and the blade. On the suction side of the blade the jet encounters the main flow and the entropy rises further, this time due to mixing processes based on the jet flow theory (figure 2.7 C). Note, that up to this point, the increase in entropy has not been seen as loss. Here, due to the insignificant mass flow through the gap compared to the main flow, the mixing of the two flows equalizes the temperature difference and thereby renders the process irreversible.

Due to the obvious downsides of tip leakage flow over unshrouded blades, significant effort has been made to reduce - or rather control - the effects, resulting in different designs for the cascade. One approach places additional tips at the top of the blades. These so called squealer tips reduce the radial clearance, without risking full blades impacting the casing in case of failure conditions. The additional tips are thin enough, so that contact with the casing might not lead to critical damage on neither the casing nor the blade and yet achieve the desired reduction of tip leakage flow by reducing the gap. The advantages of this design feature have been studied in a wide variety by authors like Ameri et al. (1998) or Azad et al. (2002). The ideas for location and amount of additional tips depends mainly on personal design philosophies of the engineer, as can be taken from the three exemplary geometries in figure 2.8, showing the work of Azad et al. (2002).

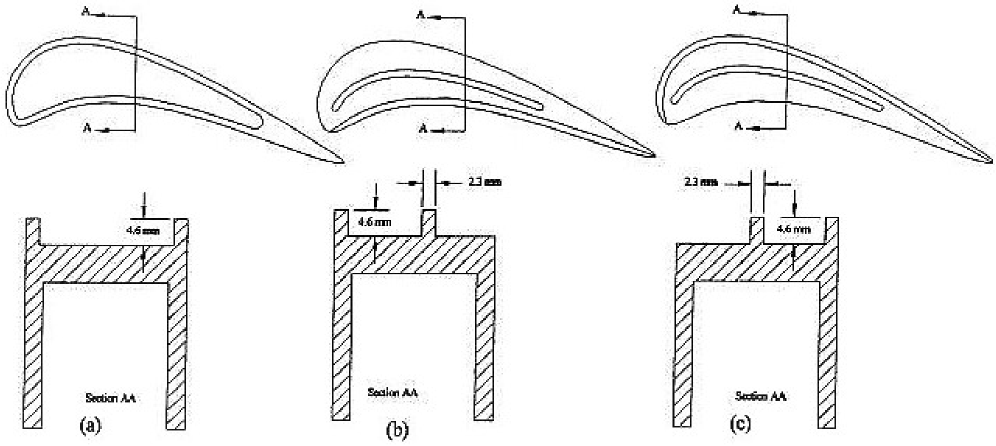


Figure 2.8: Different squealer tip geometries

Apart from a variety of different blade tip geometries being used, the presence of a tip shroud changes the flow behavior in the radial clearance area drastically since shroud leakage and tip leakage are fundamentally different. While the fluid passes the blade from pressure to suction side at an unshrouded blade, the presence of a shroud blocks this path entirely. Instead, the fluid takes another route. A separation flow forms, passing the top of the shroud and reentering the main flow after the blade row. This means that the leakage flow passes the entire blade row instead of one blade. Figure 2.9 displays the circumstances.

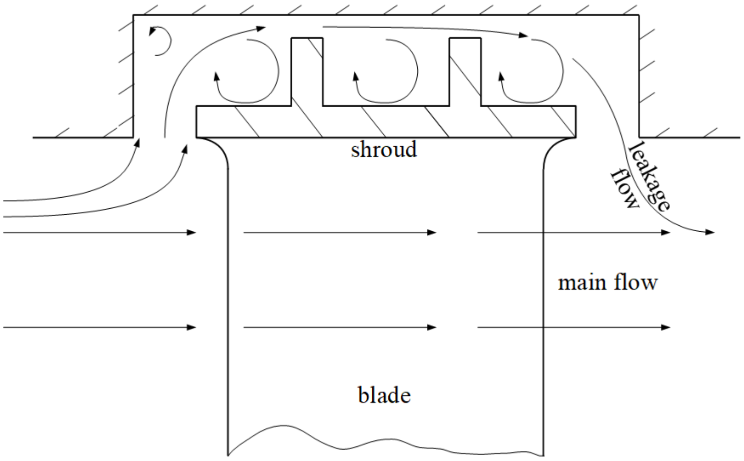


Figure 2.9: Leakage flow over shrouded tip geometry

Even though the location of separation and the flow path change, the previously mentioned downsides of mixing processes and reduction of mass flow through the blade passages still remain. Since the option is equally viable and is often realized to dampen long blades, measures have been taken in order to reduce the shroud leakage flow in case of this geometry. The approach is comparable to the squealer tips. The shroud's top is not modeled as a plane, but rather contains a random amount of additional fins, as displayed in figure 2.10. This again improves the radial clearance locally and thereby reduces the leakage flow, without risking large parts of the shroud impacting the casing in extreme cases.

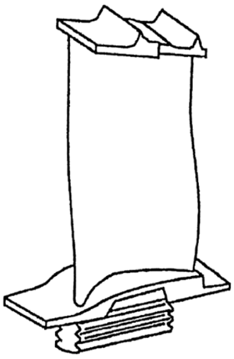


Figure 2.10: Shrouded tip geometry (Source: Earl and Ramendra (1995))

The variety of solutions for problems caused by tip leakage flows are highly versatile and depend on design philosophies of companies and engineers. The presented geometries can be seen as a selection of basic principles used in the design of turbomachinery. When working within this field, these geometries can be encountered with possible minor variations.

2.4 Radial Clearance

As has been made apparent in the last section, a goal of machine design is the minimization of the operational rotor clearance. This has to be done while accounting for various dynamic effects that are changing according to the operational condition. At rotational speeds of multiple 1000 Revolutions per Minute (RPM) and fluid temperatures of more than 1500 °C, dynamic forces and temperatures lead to a change in radial clearance. The effects can be broken down to three major contributors:

- Rotordynamics
- Centrifugal Force
- Thermal Expansion

The consideration of every aspect leads to the prediction of an operational clearance, according to which the machine is designed. Unfortunately, this process is afflicted by certain uncertainties (oil temperatures, stiffnesses, etc.), which is why a margin has to be defined for operational machines, usually depending on the machine's field of application. In order to give a glimpse over the dynamics that are present between operational conditions, these phenomena will be discussed within this section section. This will be concluded by a brief outlook over the consequences of contact between rotor and casing.

2.4.1 Rotordynamics

The field of rotordynamics covers every aspect of a rotating machine, including prediction of shaft behavior for unbalanced rotors, orthotropic stiffnesses, instabilities within the bearings or aerodynamic forces on the blades. To mathematically model rotor behavior of an axial turbomachine, a system in accordance with the Jeffcott rotor model (or Laval rotor) discussed by Marn (2014, page 132) can be introduced. This model allows for basic elastic shaft behavior to be modeled and can therefore be used as foundation for problem analysis.

> The Jeffcott rotor model consists of a massless shaft with a single disk in the middle of that shaft. The shaft is supported by rigid bearings on either side of the disk. The mass of the disk is concentrated in its center of gravity.

In order to acquire simple statements, only the effect of an unbalanced rotor shall be discussed. Therefore, the rotor is oriented vertically as shown in figure 2.11, so that the influence of gravity can be neglected.

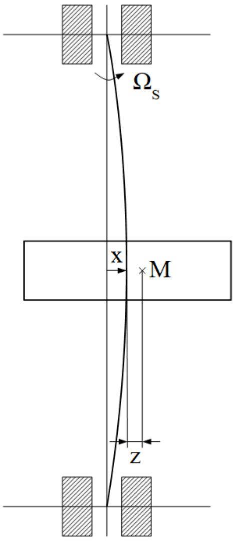


Figure 2.11: Jeffcott rotor model (Source: Marn (2014))

The momentary horizontal displacement amplitude of the rotor's centerline is given by the value x. The unbalance is modeled by horizontally shifting the rotor's center of mass M away from the center line. The distance between the shaft's centerline and the center of mass M shall be the value z. The shaft is assumed to rotate with the angular velocity Ω_s. Starting with the definition of the centrifugal force F_C, affecting the mass m with

$$F_C = m(x + z)\Omega_s^2 \tag{2.3}$$

The resulting restoring force F_S, due to the spring characteristic of the shaft, can be given by

$$F_S = -cx \tag{2.4}$$

with c being the spring's stiffness. According to Newtons third axiom, where actio equals reactio, the two forces have to cancel each other out, which is made apparent in figure 2.12 by reducing the entire model to the rotor.

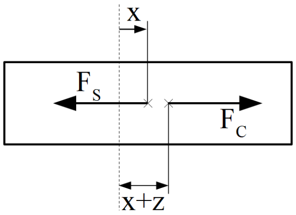


Figure 2.12: Sum of forces on the rotor caused by an unbalanced mass

$$F_C = F_S \tag{2.5}$$

$$m(x+z)\Omega_s^2 = -cx \tag{2.6}$$

After transforming this equation to explicitly display x, the expression

$$x = \frac{mz\Omega_s^2}{c - m\Omega_s^2} \tag{2.7}$$

is acquired. Introducing the natural angular frequency for a system moving in the x-plane with $\omega_0 = \sqrt{\frac{c}{m}}$, the final statement is

$$x = \frac{\Omega_s^2}{\omega_0^2 - \Omega_s^2} z \tag{2.8}$$

This equation describes the rotors horizontal displacement x, dependent on the angular velocity Ω_s. Starting at the stationary rotor $\Omega_s = 0$ and increasing its velocity, a few characteristic movement states can be highlighted. Table 2.1 shows the result for this parameter variation.

Table 2.1: Rotor angular velocity variation

Ω_s^2 $[\frac{rad}{s}]$	x [m]
0	$\frac{0}{\omega_0^2 - 0} z = 0$
ω_0^2	$\lim\limits_{\Omega_s^2 \to \omega_0^2} \frac{\Omega_s^2}{\omega_0^2 - \Omega_s^2} z = \frac{\omega_0^2}{\omega_0^2 - \omega_0^2} z = \frac{\omega_0^2}{0} z = \infty$
∞	$\lim\limits_{\Omega_s^2 \to \infty} \frac{mz\Omega_s^2}{c - m\Omega_s^2} = \frac{mz}{\frac{c}{\Omega_s^2} - m} = \frac{mz}{0 - m} = -z$

The same variation can be seen in figure 2.13, where statements become more obvious. On the left side the amplitude of the rotor for an undamped system (D = 0) and a damped system (D > 0) at changing angular velocities is visualized. On the right side, the phase angle between the rotors displacement direction (response) and the direction of the force due to the unbalance (cause) is made apparent. For the stationary rotor, the horizontal displacement x of the the shaft is zero and cause and response are aligned (0 deg). From this point, while increasing the angular velocity Ω_s, amplitude and phase start rising. As soon as Ω_s approaches the nature angular frequency ω_0, the displacement amplitude rises towards infinity at the undamped rotor (D = 0), entering the so called resonant state. Fortunately, dampening in real machinery limits amplitudes to finite magnitudes (D > 0). At this state, the phase difference equals exactly 90 deg. After exceeding this angular velocity, the amplitude starts decreasing since the force (cause) is now counteracting the displacement x (response) due to a phase difference of more than 90 deg. This means that increasing the angular velocity even further after passing ω_0 leads to the displacement amplitude x approaching the value of z (180 deg). This phenomenon, where the displacement of a rotor spinning at angular velocities above its nature angular frequency strives towards the distance between the shaft's centerline and the rotor's center of mass, is called self centering of the supercritical rotor.

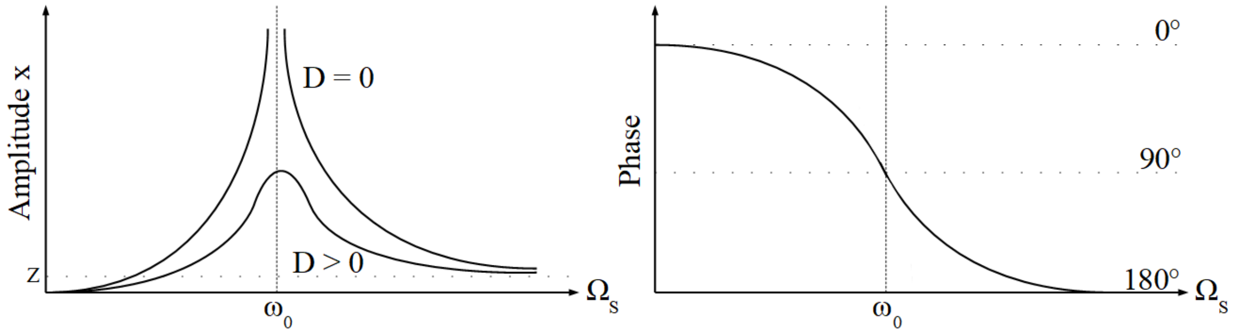
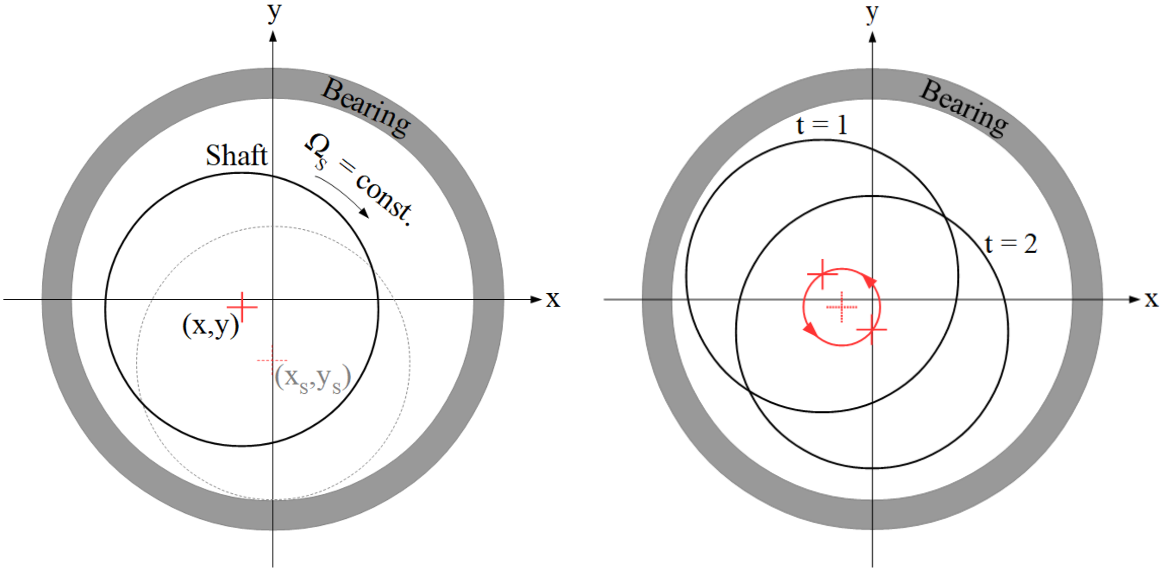


Figure 2.13: Frequency response of the Jeffcott rotor model (Source: Marn (2014))

Since the bearings, that are holding the shaft of a turbomachine, are attached to its casing, this displacement caused by the unbalanced rotor leads to a localized change in radial rotor-casing clearance. Therefore, in order to create a foundation for safe machine design, regulations concerning permitted maximum displacement amplitudes have been established in cooperation with other design disciplines. For example, this includes instructions concerning the mathematical estimation of unbalanced rotors, the balancing of the rotor masses or the limit for remaining unbalances. Note, that even though these regulations exist, perfect balancing of rotor masses is not realistic in practical application. This is why the effects described throughout this chapter are constantly relevant in clearance behavior.

In order to establish a common ground for frequently used expressions and to introduce vocabulary used throughout the rest of this thesis, the discussion of transient shaft behavior for a shaft held in place by fluid bearings will be discussed. For that purpose the axial view of the shaft shall be used. The plane is parallel to the radial direction and perpendicular to the direction of the axis connecting the two bearings. The entire plane is intersecting one of the two bearings. This idea is presented in figure 2.14 (a), where two states of a shaft are indicated by the the shaft's circumference (black) and it's center position (red). The coordinates (x,y) serve to reference the shaft's center position within the bearing. In dashed, the position (x_S, y_S) of the mentioned shaft can be seen at a angular velocity of $\Omega_S = 0$. It is resting at the lowest point of the fluid bearing until an increased angular velocity induces the required hydrodynamic lift to shift the centerline to a general location (x,y).



(a) Increase in angular velocity

(b) Modeling of rotordynamics

Figure 2.14: Axial view of transient shaft states

The displayed position (x,y) in figure 2.14 (a) for a shaft spinning at constant angular velocity neglects every aspect of rotordynamics that has been mentioned above. Therefore, the figure has to be extended. For the present argument the path of the centerline will be shown for the previously described unbalanced rotor at constant Ω_S. As has been proven, an unbalance causes the displacement of the shaft's centerline. The position of the shaft in the bearing for two points in time can be seen in figure 2.14 (b). The sum of one full revolution of the shaft describes the indicated circle in red which is commonly called 'orbit' of the centerline. Note, that the center of this orbit (dashed red cross) is still the stationary offset of the shaft caused by the hydrodynamic lift of the bearing (x,y).

This form of referencing a shafts location is commonly used for vibration monitoring at turbomachinery, which is why it will be adapted by this thesis. Therefore, the plane that has been placed parallel to the bearing - intersecting them - and perpendicular to the axis connecting the two bearings is now moved in axial direction until it intersects the rotor. The result can be seen in figure 2.15.

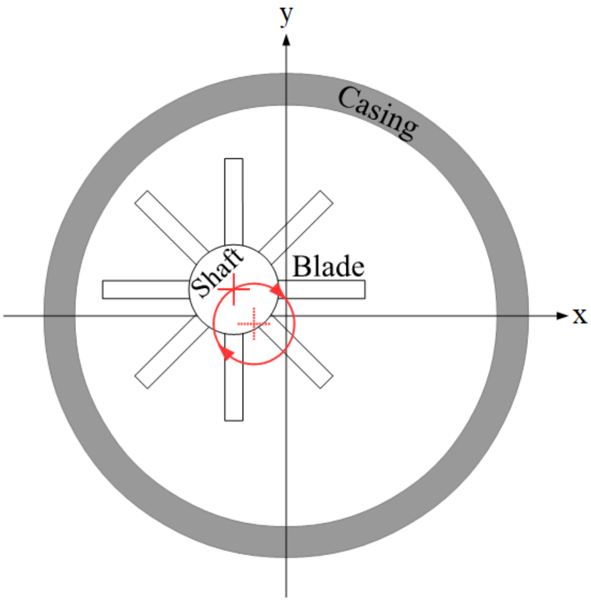


Figure 2.15: Display of rotor position within an axial turbomachine

Note, that an unbalanced mass is only one of many reasons causing orbits. Stiffnesses of the shaft and bearings, aerodynamic forces or other external factors can influence the path the rotor takes. Therefore, the circular path in figure 2.14 (b) can be elliptic, contain loops or be entirely chaotic - only dependent on the rotordynamic state of the shaft. Nevertheless, two expression can be used to adequately describe the shaft's centerline position within the casing:

- Offset: The stationary component of the shaft's centerline displacement within the casing described by (x,y) in figure 2.14 (a).
- Eccentricity: The current distance between the stationary offset and the absolute position. In figure 2.14 (b) the eccentricity is a constant value, describing the radius of the red, circular orbit.

2.4.2 Centrifugal Force

While the centrifugal force has already contributed to the change in radial clearance in combination with an unbalance in section 2.4.1, another physical process can be linked to this dynamic force. When the shaft is spinning, the disk and the attached blades are affected increasingly by their own weight. The force for a mass centered at a single point can be given using equation 2.3

$$F_C = \Omega_S^2 x m$$

Using a differential approach, both the blade's and disk's strain, can be described. For the purpose of this argument it shall suffice to prove an extension of the blade as a result of centrifugal force. Therefore, assuming elastic deformation of the material, Hooke's law can be applied:

$$\sigma = E\epsilon \tag{2.9}$$

To simplify calculations, the blade will be regarded as a square beam with a length much greater than it's thickness. The beam displayed in figure 2.16 is observed while affected by it's differential mass $Adx\rho$ during stationary rotation with angular velocity ω. Note, that the only respected force is the radial centrifugal force. Any other radial or tangential force is neglected and neither bending nor torsion of the beam is allowed.

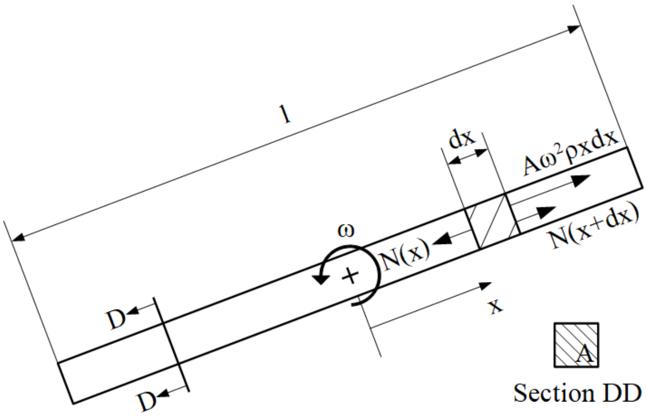


Figure 2.16: Model of rotating beam

Starting by using Newton's first law due to the stationary nature of the forces, the sum of all forces that are active on the infinitesimal mass element $Adx\rho$ in direction of x are:

$$\sum F_x = N(x) - N(x+dx) - A\omega^2 \rho x dx = 0 \tag{2.10}$$

Making use of the linear term of a Taylor expansion under the assumption of small deformations, the expression $N(x+dx)$ can be rewritten as

$$N(x+dx) = N(x) + \frac{dN(x)}{dx}dx \tag{2.11}$$

This leads to

$$N(x) - N(x) - \frac{dN(x)}{dx}dx - A\omega^2\rho x dx = 0 \tag{2.12}$$

$$-\frac{dN(x)}{dx} - A\omega^2\rho x = 0 \tag{2.13}$$

$$\frac{dN(x)}{dx} = - A\omega^2\rho x \tag{2.14}$$

$$dN(x) = - A\omega^2\rho x dx \tag{2.15}$$

By integrating indefinite the expression

$$N(x) = -A\omega^2\rho\frac{x^2}{2} + C \tag{2.16}$$

is found, where the unknown constant C can be found when observing the normal $N(x)$ force at the free end of the beam.

$$N(\frac{l}{2}) = 0 \tag{2.17}$$

$$C = A\omega^2\rho\frac{l^2}{8} \tag{2.18}$$

$$N(x) = A\omega^2\rho(\frac{l^2}{8} - \frac{x^2}{2}) \tag{2.19}$$

The strain in direction of x caused by stress in the same direction can be defined as

$$\epsilon_{xx} = \frac{\triangle L}{L} = \frac{du}{dx} \tag{2.20}$$

and rewritten using Hooke's law

$$\sigma_{xx}(x) = E\epsilon_{xx}(x) = E\frac{du(x)}{dx} = \frac{N(x)}{A} \tag{2.21}$$

After transforming this equation to display the infinitesimal displacement $du(x)$

$$du(x) = \frac{\omega^2\rho}{E}(\frac{l^2}{8} - \frac{x^2}{2})dx \tag{2.22}$$

and indefinitely integrating, the expression showing the deformation of the beam becomes

$$u(x) = \frac{\omega^2 \rho}{E}(\frac{l^2 x}{8} - \frac{x^3}{6}) + D \tag{2.23}$$

To find the unknown constant D the displacement at the root of the blade can be used. Since the root is the center of the rotating beam, the displacement at this position is $u(0) = 0$ which leads to $D = 0$. This results in the final deformation of the beam according to

$$u(x) = \frac{\omega^2 \rho}{E}(\frac{l^2 x}{8} - \frac{x^3}{6}) \tag{2.24}$$

$$u(\frac{l}{2}) = \frac{\omega^2 \rho}{E}(\frac{l^3}{16} - \frac{l^3}{48}) \tag{2.25}$$

$$\triangle l = 2u(\frac{l}{2}) = \frac{\omega^2 \rho}{E}\frac{l^3}{12} \tag{2.26}$$

Assuming this beam is spinning at the center of a casing, the extension of $\triangle l$ leads to the mentioned reduction of clearance. Note, that the disk can be treated equally. Nevertheless, since the mathematical description is much more complex and the extension of the blade by the centrifugal force has been proven, the section will not elaborate the extension of the disk.

2.4.3 Thermal Expansion

Aside from deformations caused by dynamic loads, the effect of the temperature difference between stationary, cold machine and operational condition has to be considered. This is especially important for gas turbines, where the gas is initially compressed in a compressor which is followed by additionally heating within a combustion chamber and concluded by the expansion of the gas within a turbine. In the area of the exit plane of the combustion chamber and the entry plane to the first stage of the turbine, gas temperatures can exceed 1500 °C. While the gas temperatures can change fast, the material takes time to heat up or cool down. A possible stationary temperature profile of the material is displayed in figure 2.17 for an operational axial gas turbine.

During transient heating phases, a phenomenon called thermal expansion changes the dimensions of all heated components according to the equation

$$\epsilon_{th} = \frac{\triangle L}{L} = \alpha_{th} \triangle T \tag{2.27}$$

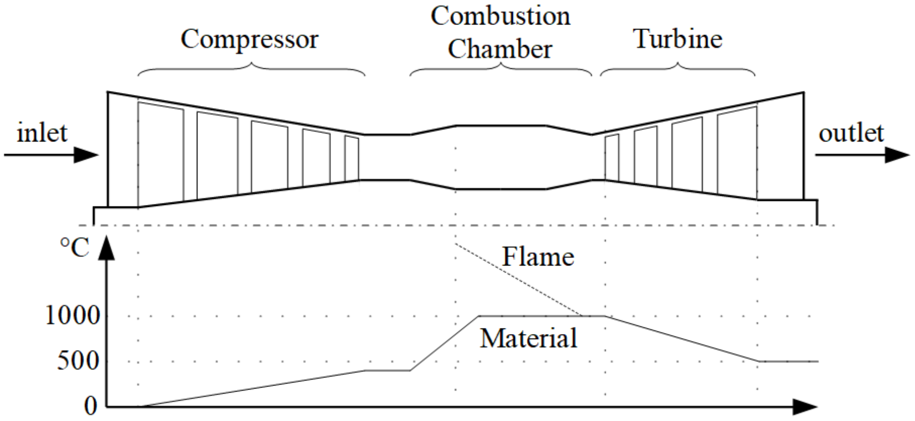


Figure 2.17: Temperature distribution of the material in a combustion turbine

Note, that this is a simplified, one-dimensional way of looking at expansion. This equation states a strain that shows a linear correlation to the temperature change $\triangle T$. The non-linear character of the expansion can be accounted for using the thermal expansion coefficient α_{th}. This parameter is dependent on the material itself and the temperature. In literature like Böge (2013), the value for α_{th} for chrome steel is given by $11 \cdot 10^{-6} K^{-1}$ within a temperature range from 0 to 100 °C. For aluminum this value is stated to be $23.5 \cdot 10^{-6} K^{-1}$ within the same temperature range. The variance of the coefficient for two different metals and the temperature dependency give an idea about the difficulties predicting thermal expansion within a thermal turbomachine.

Apart from final deformation states, the heating and cooling processes between machine states pose an additional threat. Given the fact the operational conditions change, transient and localized temperature has to be expected. When not in use, after a sufficient amount of time, all the machine components show the same temperature as the machine's surrounding. As soon as the machine is started and hot working fluid passes the turbomachine, the heating up of the different components varies in time. In order to achieve thermal equilibrium between gas and casing, much more time has to pass, than for the blades stationary temperature to be reached. This is caused by the usage of different materials (different thermal expansion coefficients α_{th}) and differences in mass.

To summarize, the thermal expansion of a component is mainly affected by following aspects:

- Material: thermal expansion coefficient α_{th}
- Mass: time to reach an equilibrium with the gas temperature
- Gas Temperature: changing gas temperatures throughout the entire machine (compare figure 2.17)

All of these factors combined result in different deformation magnitudes for every part, resulting in a huge challenge for engineers, especially during transient operational conditions. The complexity of the process makes apparent that the radial clearance is affected greatly by thermal expansion. Most importantly casing, disk and blade cause the change of radial clearance since they are usually directly exposed to the working fluid's temperature.

2.4.4 Contact

The three mechanisms described throughout the past pages combined make up for the expected radial rotor-casing clearance behavior. While stationary turbomachinery shows sufficient magnitude, operational effects alter the available margin and pose a possible threat if the design fails to provide sufficient room for movement during transient states. The trade-off between the benefits gained due to reduction of radial clearance and reduction in margin movement climaxes in the threat of contact between rotating and stationary parts. This is especially relevant for transient turbomachinery application, like for example in airplane combustion turbines. Figure 2.18 shows a possible profile of radial rotor-casing clearance from the start-up until in-flight equilibrium. To simplify the explanation, the effects of rotor dynamics are neglected and only thermal expansion and centrifugal force are accounted for.

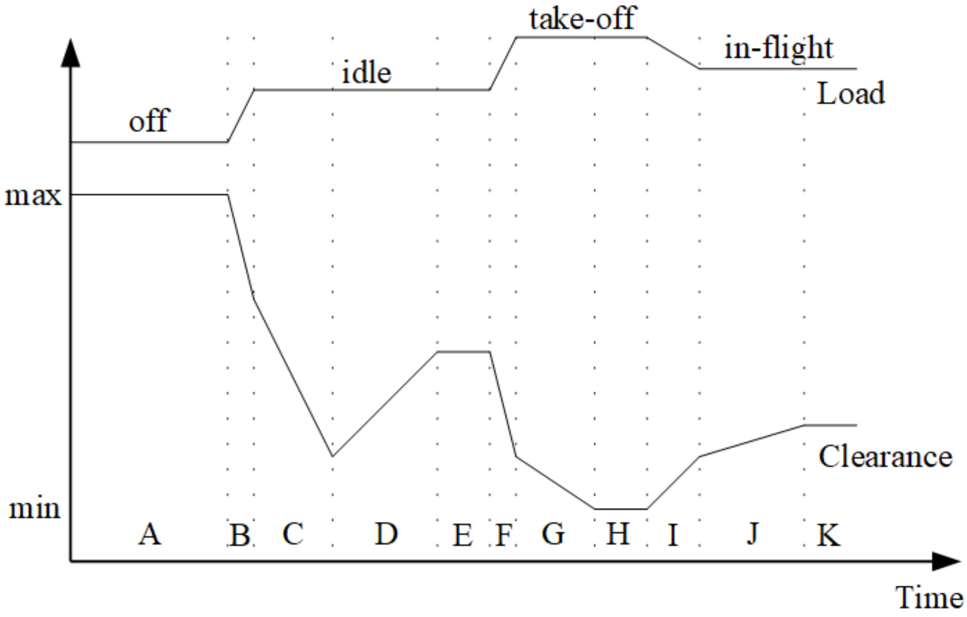


Figure 2.18: Possible radial rotor-casing clearance over time for an airplane combustion turbine

- A: off - The clearance is at its maximum. → clearance stationary
- B: transition from off to idle state - An increase in angular velocity of the rotor and increase in working fluid temperature cause centrifugal force and thermal expansion to extend the blade. → clearance decreases
- C: idle state - Centrifugal force is not increased anymore since angular velocity is stationary. Differences in heating up of components cause blades to extend faster than the casing. The process is heading towards a local minimum. → clearance decreases
- D: idle state - The blades reach their final temperature and extension stops, while casing keeps on heating up. The local clearance minimum has been passed. → clearance increases;
- E: idle state - The casing has reached temperature equilibrium with the working fluid as well and stops extending. → clearance stationary
- F-H: transition from idle state to take-off: The additional increase in angular velocity and fluid temperature due to increased load leads to further extension (the transition is comparable to steps B to E). → clearance decreases
- I-K: transition from take-off to in-flight: The reduction of angular velocity and fluid temperature reverses a part of the component extension → clearance increases

If at any point in time the clearance falls short of zero, the rotor collides with the casing. Note, that this is the reason for the extended waiting period in the idle state during heat up in the displayed case. If the turbine would have been taken into take-off mode in section C, the differences in thermal expansion speeds of the various components might have led to excessive clearance reduction. This can equally be caused by externally induced forces that displace the shaft, as for example during maneuvers of the plane.

The unavoidable downside of decreasing radial clearance to increase performance is the necessity of more and more precise rotordynamics and heat transfer analysis. Due to the complexity of real-world machinery and limitations concerning computational power, the precision for these calculations has an upper limit, leading to a remaining uncertainty when predicting shaft behavior. If this uncertainty gets paired with aggressive clearance design, the outcome can be drastic. Under certain operational conditions, like insufficient dampening, radial displacement spikes can occur that exceed the clearance and result in contact between rotational parts and the casing. The outcome is a loss of the blade and/or an impulse on the entire shaft, causing a dampened natural oscillation of the machine which can potentially cause follow up impacts. In some cases the first impact already leads to the malfunction of the material, therefore to the destruction of the blade and in some cases even the entire machine. In an ideal scenario, the thermal energy that is created due to friction between the relatively moving components only ends up burning parts of the blade. In this case, even if the material does not fail initially, the loss of mass on the burnt blade might result in an alteration of balance and therefore overall shaft dynamics. In either case the rotor-casing contact causes degeneration of the machine to some extent, making it a threat that has to be avoided.

3 Measurement Chain

As has been stated throughout the problem definition, the goal of this thesis is the model based development of a clearance monitoring software using LabView. While LabView shall handle the signal processing, the data shall be created side-by-side using Matlab. The proposed model shall simulate basic machine behavior that has been outlined in the previous chapter, while being based on a real world measurement principle that can be realized later on in the project. Since the importance of sufficient radial rotor-casing clearance margin has been outlined and the influencing mechanisms have been discussed, this chapter aims to provide an overview over measurement techniques used to track this parameter in on-line axial thermal turbomachinery. The goal is to establish which elements are important for the modeling process and how they influence the resulting signal.

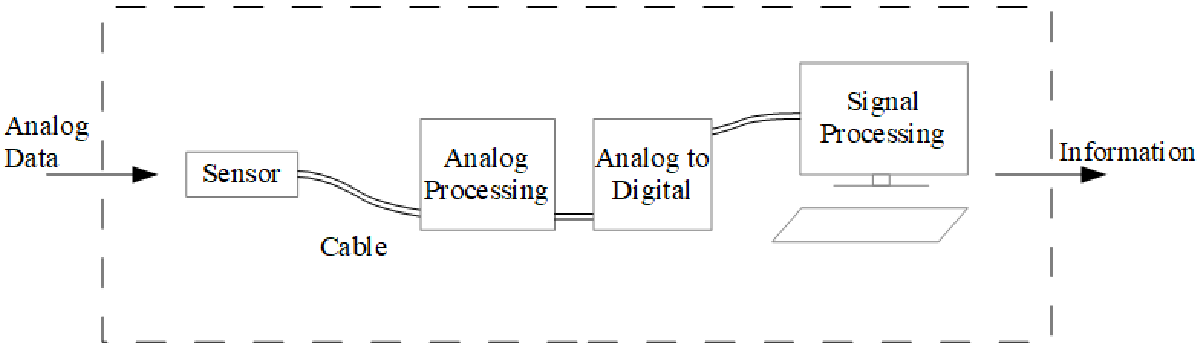


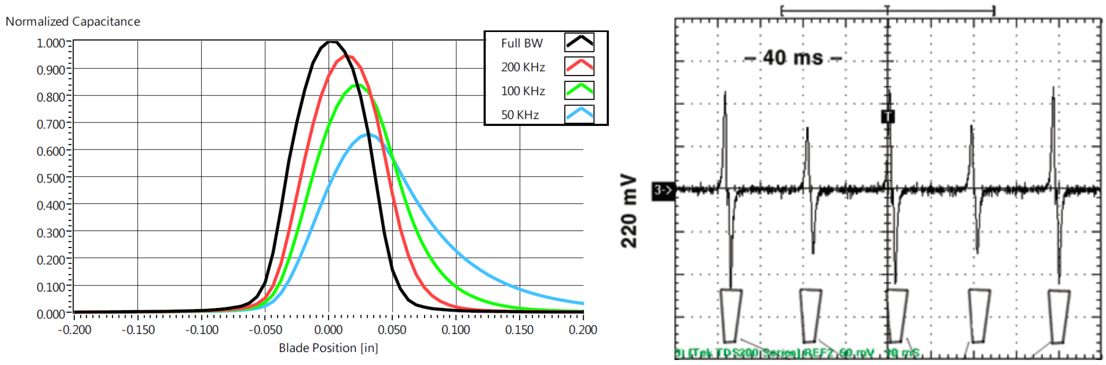
Figure 3.1: Measurement chain

Starting with the bigger picture and focusing on component after component afterwards, figure 3.1 displays the basic layout of a measurement chain. The goal of every type of measurement is to gather information that is available in form of analog data at any time. In order to be able to pick up a portion of this information, it is vital to realize what kind of physical principle the desired information is tied to. Taking sound as an example, the data is present in form of fluctuation of pressure over time. As soon as the information carrier is identified, a device that interacts in a predictable manner with said physical value - a so called sensor - can be used in order to extract the analog data from the entirety of data available. This is done by placing the sensor in proximity to the area of interest and

designing a circuit that transform the physical value into an electrical signal. In the present example of sound, the sensor usually consists of a membrane and elements that transform the oscillation of this membrane into either current or voltage.
The electrical signal can then be passed through a cable to an analog processing unit, where signal enhancing can be performed. This means the desired information is amplified, while unwanted additional components can be suppressed. After that, the signal proceeds to the Analog-to-Digital (AD) converter, where the analog signal is digitalized and made available for further processing and displaying on a computer. At this point, a connection between the signal and the initial physical value has to be defined. By precisely adjusting this connection and verifying the measurement results with known data - so called calibration - the desired information can be output to the user.

3.1 Influential Hardware

In order to be able to simulate tip clearance measurement data the most influential decision is connected to the sensing principle. As will become apparent within the next pages, the sensor type and the circuit method determine the general structure of a signal. This statement can be proven by looking at figure 3.2. On the left, actual measurement results of a blade passing by a capacitive sensor can be seen. The authors Haase and Haase (2013) have conducted research concerning the influence of different system bandwidths on the signal. The results all show a single peak caused by the interaction with the blade. When comparing this signal to the signal of multiple blades passing an eddy current sensor on the right, the differences between the measurement methods become apparent.



(a) Capacitive Signal

(b) Eddy Current Signal

Figure 3.2: Comparison of different sensor signals (Source: Flotow and Drumm)

While the capacitive signal shows only a single peak, the eddy current signal consists of a positive and a negative amplitude spike caused by a blade passing the sensor. This is due to the different dependencies of the sensors. While the capacitive sensor is mainly influenced by the target geometry, the eddy current sensor is sensitive to the passing velocity, direction and temperature.
As will be discussed later on, the other components of the measurement chain do influence the quality of the signal, but do not fundamentally alter the structure. Therefore, the modeling process is limited to the sensing principle and a corresponding circuit method. This ensures a relevant signal, while still being relatively simple to analyze due to less influential parameters, and basing the development of the monitoring routine on realistic signals.

3.1.1 Sensor Types

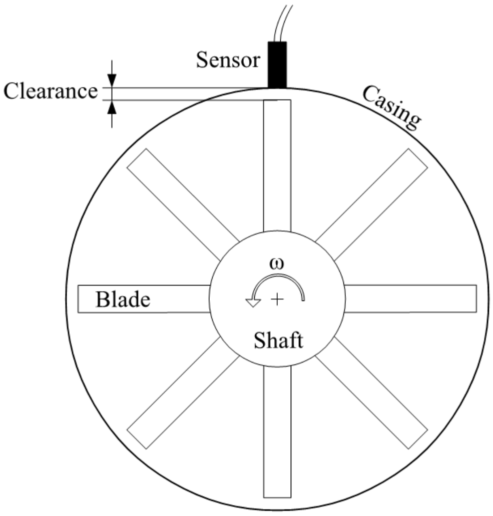


Figure 3.3: Sensor mounted on an axial turbomachine

As has been mentioned, the aspect of choosing a sensing principle determines the characteristics of the signal greatly. Due to the nature of turbomachinery, only a few sensor types have proven to be reliable in the given hostile environment. High temperature, particle loaded flows and vibrations have the effect of not only decreasing signal quality, but as well damaging measurement hardware, if not contained properly. One thing that all presented sensors have in common is the placement on the machine, when trying to measure the clearance. Figure 3.3 shows the mounted sensor, facing in radial direction.
Other than that, sensors can be divided into contact sensing and contact-free sensing. At

contact sensing the sensor has to be in constant pairing with the components to measure, in the present case the blades, as is with strain gauges at strain measurement for example. Since clearance measurement is conducted between the stationary casing and the spinning rotor where differences in velocity are considerable, contact-free sensing has proven to be more suitable. By mounting the sensor plane with the casing the influence of the sensor on the flow can be limited and the stress on all involved components is kept to a minimum. Many sensor types of this type have been developed and tested in turbomachinery application, which has lead to the identification of a few sensing principles to be suited best. A general overview of the most commonly used sensors has been provided by authors like Sheard et al. (1997), including:

- Capacitive Sensors
- Eddy Current Sensors
- Optical Sensors
- Microwave Sensors

Capacitive Sensor: The idea behind capacitive sensing is easily explained by looking at a plate capacitor in general, displayed in figure 3.4. Two opposing conductor plates are separated by a non conductive region. When applying a driving voltage, charge displacement takes place, which correlates to the area A and the distance d of the plates.

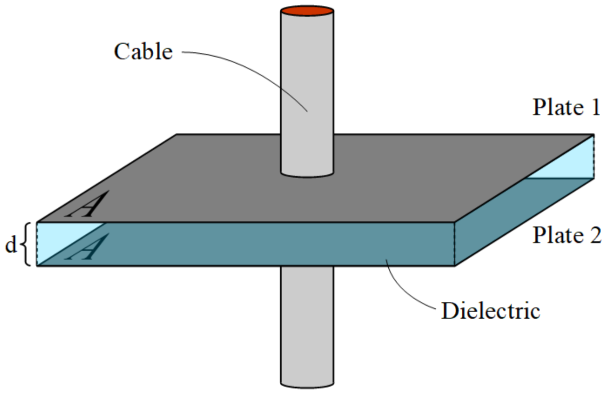


Figure 3.4: Plate capacitor

The governing equation for this sensor type is given by

$$C = \epsilon_0 \epsilon_r \frac{A}{d} \tag{3.1}$$

where C represents the capacitance in Farad, A the plate area, d the distance between the two plates, ϵ_0 the permittivity of vacuum and ϵ_r the relative permittivity of the non-conductive area (aka. dielectric). Additionally, equation

$$C = \frac{Q}{V} \tag{3.2}$$

linking capacitance C, voltage V and charge Q in Coulomb can be introduced. Assuming to know the driving voltage V for the capacitor and given the fact that the capacitance C changes according to the distance d of the plates, the change in charge Q can be measured in order to obtain information about the distance. This leads to

$$\triangle Q = \triangle CV = \frac{1}{\triangle d}\epsilon_0\epsilon_r AV \tag{3.3}$$

In case of clearance measurement in turbomachinery the sensor is set up to be one of the two capacitor plates, while the rotor component (blade or shroud) is set to be the other. The distance between the sensor tip and the passing object corresponds to the distance d in given equations. Note, that for relatively moving objects, such as the rotor, the plate area A has to be modeled as a changing variable as well, leading to

$$\triangle C = \frac{\triangle A}{\triangle d}\epsilon_0\epsilon_r V \tag{3.4}$$

This equation can be used to link a capacitance value to distance. Close analysis of this sensor type in combination with advances in charge amplified clearance measurement has been done by Haase and Haase (2013). Note, that apart from the low price, the size of this sensor poses its main advantage. Products, like the Capacitec® HPT-75, show a maximum in recommended sensing range that is commonly equal, or close to equal to its diameter. Dependent on the expected clearances an adequate model can be acquired that provides the desired resolution while still being of reasonable size.

Eddy Current Sensor: Eddy current sensors utilize the effect that eddy currents are formed, when a moving conductive material moves through a magnetic field. This is due to the fact that voltage is induced into said material, which can be assumed to be a part of the rotor passing the sensor. These currents cause the creation of another magnetic field on their own. Compared to the initial magnetic field, the newly induced field is opposing the first one, which leads to the two fields counteracting each other.
The strength of this interaction is coupled with the distance between the fields and can be

measured. Apart from the advantageous size of eddy current sensors, the nature of electromagnetic fields enables this sensor type to be operational even within particle contaminated flows, as long as the blade is ferromagnetic. This, on the other hand, is not always given in turbomachinery nowadays, since lightweight materials - such as titanium - are on the rise due to their performance under high rotational velocities and temperatures. Another disadvantage of eddy current sensors is the dependency of the signal on the blade's passage velocity and its temperature, which makes the extraction of the desired information difficult. Nevertheless, the principle has been studied and applied in turbomachinery application, as has been presented by Fabian et al. (2005).

Optical Sensor: Measurements of rotor displacement via optical methods has been discussed by many authors, like Jia and Zhang (2011). Basically, the system consists of a light source that focuses light beams onto the surface that should be evaluated, and picks up the reflection through a photo-detector, where the information is obtainable in terms of distance-dependent intensity. The main advantage lies within the huge temperature range this method can be applied in, compared to the previously mentioned sensor types. Nevertheless, this is outweighed by the huge price and size difference between the optical methods and the previously introduced sensors. Additionally, in order for light being properly reflected, high standards concerning clean surfaces are needed, which can hardly be achieved in turbomachinery.

Microwave Sensor: Comparable to optical systems in size and expensiveness, microwave sensor application can be compared to small range radars. The complex connections between physical value and signal and the effort to link those two precisely renders this option too complicated to model for the underlying purposes. Nevertheless, in case of the measurement region showing high temperatures, particle contamination and unlimited space, this method might still be considered feasible. Even though it has been a lesser chosen alternative, research has been conducted by authors like Szczepanik et al. (2012).

Comparison: In order to provide the arguments for choosing a sensing method, table 3.1 sums up the information that has been presented throughout the previous pages. Due to the fact that after the creation of the clearance monitoring system corresponding hardware has to be purchased and the sensing principle has to be committed to for the rest of the project, this choice requires a few aspects to be considered. First, the final measurement system will be applied in a testing facility that uses externally compressed air with relatively low temperature. This results in low requirements concerning temperature tolerances which eliminates the need for complicated systems like optical or microwave sensors. This argument can be supported even further by the absence of a combustion chamber. The low particle load of the working fluid allows for the selection of one of the two cheaper options - capacitive or eddy current sensors. Second, the necessity of a ferromagnetic rotor disqualifies the eddy current system. This requirement can not always be met in the testing facilities where rotors are often composed out of aluminum. Due to the reasons above, the author of this thesis tends towards modeling the remaining capacitive sensing principle, which has another huge advantage over the other systems: the simple dependency between signal and clearance.

Table 3.1: Comparison of different sensing methods by Sheard et al. (1997)

Sensor	Resolution	Temperature	Size	Cost
Capacitive	5% of Range	1100 °C	Very Small	Low
Eddy Current	5% of Range	650 °C	Small	Low
Optical	0.05 mm	1550 °C	Large	High
Microwave	0.10 mm	1200 °C	Medium	Medium

3.1.2 Circuit Methods for Capacitive Sensors

In order to further understand the domain of capacitive sensing, a distinction between passive and active sensor behavior has to be made. Passive sensors themselves do not produce any detectable electric signal, unlike for example active piezoelectric sensors that make use the electrical signal originating in special crystals under applied stress. Therefore an auxiliary power source is needed in order to be able to detect the interaction of these passive sensor elements (membrane, etc.) with the physical value. Two methods that are used to realize an analog waveform through passive capacitive sensors can be introduced: carrier- or direct current-based. This section only introduces the basic idea behind these two types, more detailed research and performance testing has been done by Haase and Haase (2013).

Carrier Based: Carrier based methods include frequency, phase and amplitude modulation. Generally speaking, a carrier wave is applied to the sensor, making the system reactive. When the sensor interacts with the surrounding, the information gets embedded into the driving wave, resulting in a shift of a certain parameter, dependent on the chosen method.

In frequency and phase modulation, the angle of the original signal is modulated, whereas amplitude modulation modulates - as the name suggests - the amplitude. Usually the former methods are preferred, due to the fact that noise influences the amplitude of a signal and thus leads problems when the amplitude is used as information carrier.

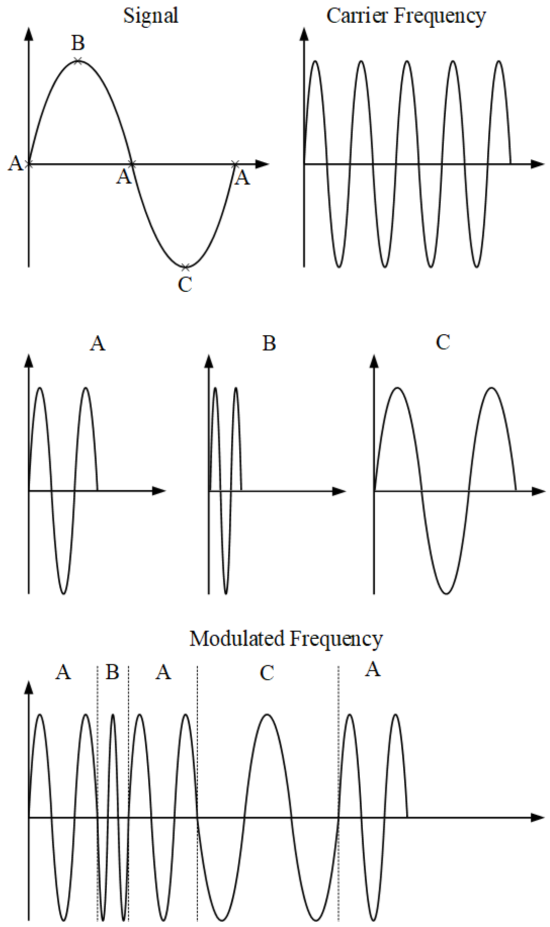


Figure 3.5: Exemplary Frequency Modulation

Frequency modulation can be explained by looking at the sine wave called 'signal', displayed in figure 3.5 left up that should be modulated. Additionally, another sine wave at a certain frequency - the 'carrier frequency' right up - is defined, which is modulated accordingly. In the second row on the left, the carrier frequency's response to a signal amplitude of zero (A) can be seen. Note, that the carrier frequency has not changed. When the signal amplitude increases towards its maximum (B), the frequency compared to the initial carrier frequency is increased respectively, as is displayed in the second row in the middle. As soon as the signal amplitude has passed its peak and starts decreasing, the carrier frequency reduces as well, until a minimum is reached at the point of minimal amplitude of the signal (C). Like

this, amplitude and frequency information of the signal can be modulated in a way, where it can later be demodulated and read. Generally the modulation takes place continuously and the modulated signal contains more frequency components between maximum and minimum value than is shown in figure 3.5 at the bottom.
The biggest drawback of all of these methods is the fact that a lot of available bandwidth is used up by the carrier wave, due to detection and filtering processes required to extract information. This is especially bad, since the information in the present case already requires significant frequencies to begin with. As a result, the demands for hardware can increase drastically, as has been discussed by Haase and Haase (2013) extensively.

Direct Current Based: An alternative method to supply passive capacitive sensors tends towards usage of constant voltage to drive the sensor (dc-based). This method can be explained by looking at equation 3.2, mentioned above for capacitive sensors:

$$C = \frac{Q}{V}$$

When applying a constant voltage V to this equation and having passing rotor components change the capacitance C, the information is delivered through a change in charge Q. Therefore circuits containing charge amplifiers have been developed to make the changes in charge detectable. Due to the nature of capacitors in this setup, only a change of capacitance can be detected, which means it is not able to pick up on an absolute value of capacitance and, consequently, absolute clearance information. Yet, the detection of bypassing blades is possible since there are times without a blade being present in sensor range, which means this value of capacitance corresponds to ∞ distance. Knowing this, the difference between the signal at no blade being present and the peaks can be measured and is representative of the clearance. The advantage of this method is that the problem of bandwidth reduction, as mentioned above at the carrier based methods, does not occur and less complex signal processing is required. This method, in combination with particularly developed analog processing units, has been refined by Haase and Haase (2013) in recent years.

Comparison: The obvious simplicity of the direct current based method makes it an apparent choice for the modeling process. This is only enhanced by the fact that deriving the signal from a frequency modulation can be handled separately and is even recommended to happen in the analog state (bandwidth requirements). Therefore, the signal that is simulated using the capacitive sensing principle is chosen not to be modulated.

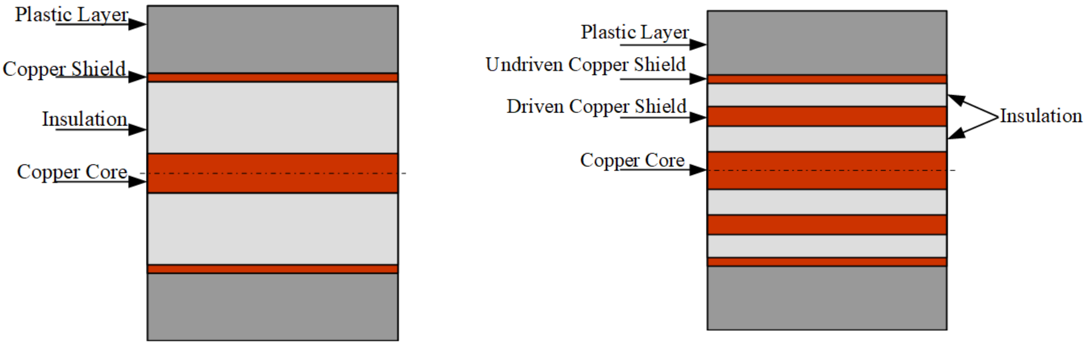
3.2 Additional Hardware

As has been stated before, the modeling of the interaction between the sensor and the rotor in combination with a circuit method already provides reasonable accuracy for the development of the clearance monitoring system. Nevertheless, in order to comprehend many processing steps and in order to be conclusive while describing the measurement chain, the remaining components are outlined briefly.

3.2.1 Cable Types

After the signal has been picked up by the sensing device, the analog signal has to be transported to the next instance via a cable. The goal here is to preserve as much of the initial information as possible, which can be challenging due to the fact that the amplitude of the electrical signal can be diminishing. Especially when working with components like capacitors, where charge is stored until the governing parameters change, an effect called leakage current - the current originating in a discharging capacitor - can occur that is picked up and distorts the information. Therefore studies have been conducted on the effects of different cable-sensor-combinations, again by Haase and Haase (2013), regarding the maximization of the linear sensor range.

Generally speaking, two cable configurations are available that can be paired with the respective sensor. Both, cables and sensors, are available in guarded and unguarded versions and can be combined as desired, resulting in different performances.



(a) Coaxial Cable

(b) Triaxial Cable

Figure 3.6: Coaxial and triaxial cable design

Coaxial: Coaxial, or unguarded cables can be found in combination with unguarded sensors. The probe usually consist of the sensing tip and a casing, which is grounded. The cable, displayed in figure 3.6 (a), is made out of a central conductor and a surrounding insulation, followed by a copper shield and a plastic outer layer. As already mentioned, the occurrence of unwanted additional electrical signal components has to be avoided in order to preserve information quality. Unfortunately, the difference in capacitance of the inner conductor and the insulation can cause signal components in the same amplitude range as the desired signal. This effect is dependent on the length of the cable, which is why compact measurement system setups are always recommended. Another downside to this cable type is the sensor's suffering from considerable fringing of the electric fields, again resulting in a reduction of the signal quality.

Triaxial: To counteract the effects of coaxial setups just mentioned, triaxial, or guarded cables are available. In combination with guarded sensors, they can supply an additional layer embedded into the insulation that is usually driven at the same voltage as the tip, to ensure a reduction in capacitance difference and reduce fringing of the field lines at the tip. To see the differences in design compare figure for coaxial cables 3.6 (a) to figure 3.6 (b).

3.2.2 Analog to Digital Converters

As has been the case with the presentation of the cables types, this section does not introduce aspects of the measurement chain that are being modeled. The presentation of this hardware element rather serves the purpose of providing a fundamental understanding of signal flow and introducing necessary vocabulary that will be used throughout the thesis.

Theoretically, the information can be digitalized and handed to a computer for further processing after it has been transfered using cables. Nevertheless, depending on the sensor type and circuit, some analog preprocessing can be required or useful. Taking for example a frequency modulated signal, it can be a significant difference in hardware requirement, if the signal is demodulated at an analog state and therefor possibly reduced in frequency, instead of passing the high frequency signal through a converter. The possibilities at this stage of the measurement chain are endless and highly dependent on the composition of the system, which is why a generalized explanation is avoided in this thesis. Therefore the focus shall be the discussion of so called Data-Aquisition (DAQ) devices - or AD converters - concerning effects on the analog signal while being digitalized.

Characterization: Currently the signal is still analog, meaning there is theoretically infinite information between two discrete points in time and the resolution of the amplitude is infinitely precise. From here on, after passing the data through an AD converter, the data is quantified. This is due to the nature of the digital domain, where values can only be comprehended in a quantified manner. Taking for example the integer data type, where the resolution is limited to natural numbers, a problem called quantization error arises. This problem is visualized in figure 3.7, where the transformation of an analog signal (up) to the integer domain (down) is visualized. Note the difference between input and output signal. Even thought there are different data types that have a more suitable stepping size, they all are finite when it comes to resolution nonetheless.

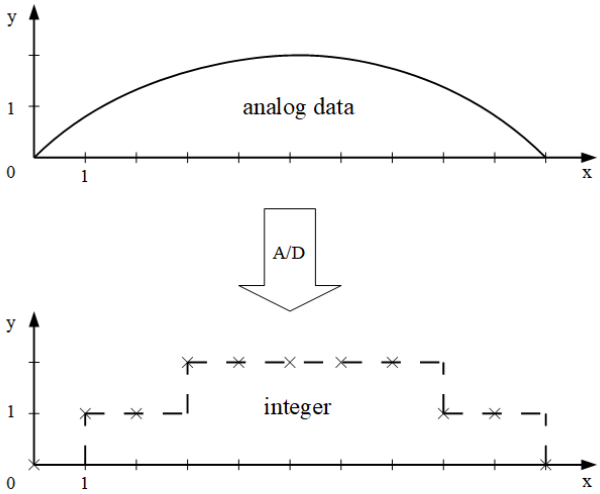


Figure 3.7: Signal discretization

Leaving aside requirements on the digital data type for accurate measurement, the analog signal is always reduced to a finite number of data points. This discretization of a signal poses the biggest difference between analog and digital signals and is described in connection with 'sample rate'.

Sample Rate: An AD converter takes samples of the analog signal every $\frac{1}{sample\ rate}$ seconds. The information between these two discrete points in time is not picked up and delivered to the processing unit, which can lead to problems when having low rates available and trying to detect high frequency data. A mathematical analysis of this problem leads to the so called Nyquist theorem, which states the importance of a sample rate that is higher than twice the expected signal frequency. The theorem is written as

$$f_{sample} > 2 \cdot f_{signal} \tag{3.5}$$

By committing to this expression, an effect called aliasing can be avoided. This means that the set of sampled data points is unambiguously able to represent the sampled signal frequency. If this criteria is not met and aliasing occurs, measurement results as shown in figure 3.8 are possible. Here, the high analog signal frequency is misinterpreted by the sampled data, leading to a completely different output signal frequency.

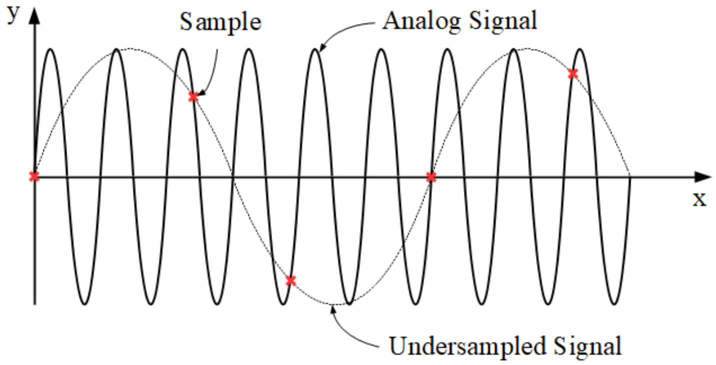


Figure 3.8: Aliasing effect

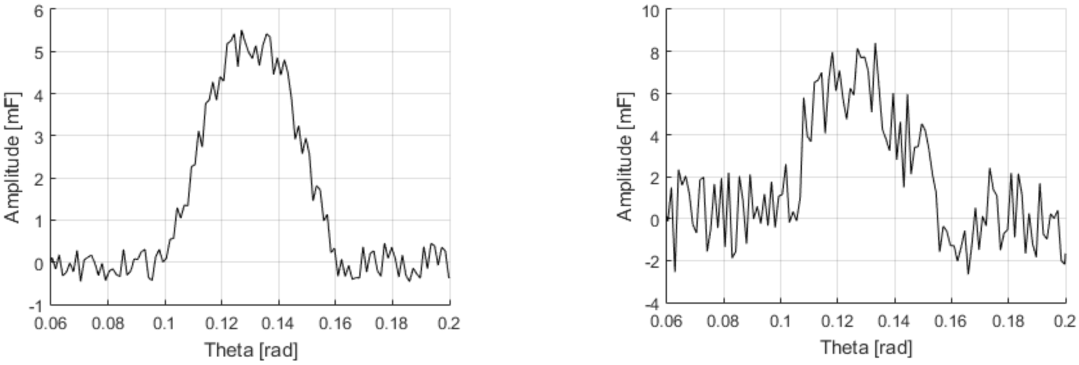
In order to ensure that this criteria is met, the signal is low-pass filtered before being passed into the AD converter, leading to the removal of frequencies that are higher than half the sample frequency. This process can be found in respective literature under the expression of anti-alias-filtering.

3.2.3 Noise

Until now effects decreasing the quality of the signal, for example due to leakage current, have been called secondary signal components. The technical expression for these disturbances is generalized as noise. This includes every aspect of a signal that has not been purposefully recorded and, dependent on the amplitude of these components, can be the reason for corrupted results. Since only the sensor and the chosen circuit method are modeled for the signal generation the expression noise is treated generalized. The model does not differentiate between sources and does not base noise on any physical phenomenon. It is seen as sufficient to account for its presence. Therefore, the introduction of the Signal-to-Noise-Ratio (SNR), which in this thesis is defined by

$$SNR = \frac{max(Signal)}{Noise} \tag{3.6}$$

helps in describing the quality of signals. High SNR means little noise being present, while low SNR indicates high levels of noise overlaying the signal. The differences can be seen in figure 3.9, where the same signal with two different amount of noise is portrayed.



(a) SNR = 10

(b) SNR = 2.5

Figure 3.9: Same signal affected by different signal-to-noise ratios

In order to improve signal quality and amplify desired information while suppressing noise, different filtering techniques are available. The choice has to be made according to where the information is embedded. If the information is located in lower frequency regions than the noise, low-pass filtering the signal can suppress unwanted components, while leaving the information untouched. If the contrary is the case, high-pass filtering can lead to the desired effect. Additionally, for more complex processing, windowed filters can be chosen. In either case, the choice is dependent on the underlying signal components and has to be made individually.

4 Mathematical Background

The following chapter introduces the mathematical principles used for the development of the software environment. The focus lies on providing the basic equations and connections in order to comprehend the statements made throughout the rest of the thesis, not elaborating complicated examples. The methods shown here are sorted in the order of their application within the signal generation and clearance monitoring routines. Starting at the signal generation in Matlab, the capacitive sensing principle that is modeled requires for the computation of the area stated in equation 3.4. Since the contours surrounding this area are available, Green's theorem can be applied, which requires for a brief explanation. Next, after the signal is successfully simulated and transmitted to LabView to be processed, the first step is filtering the signal to improve the quality. Therefore, this chapter evaluates the principle of low-pass filters and shows important effects on the signal. The filtered signal is subsequently passed through a peak detection routine within LabView in order to reduce the complex signals to the essential information. The idea behind this step requires evaluation since, similar to low-pass filtering, the quality of the processing depends on the understanding of the domain. Ultimately, the underlying mathematical principle of a processing step realized within LabView is introduced that allows for information about the rotor's position to be computed using three arbitrary clearance signals.

4.1 Green's Theorem

Since the chosen sensor type uses an area to link a distance to a signal, the modeling process for the Matlab routine has to deal with the mathematical description of that surface. As has been stated, the model relies on areas enclosed by polygons, which is why Green's theorem can be used for determining the value of the area A using the contour C of the surrounding curve. The theorem, explained by Khan Academy, originates from two-dimensional calculus where it is used to link the line integral over a curve C to a double integral over the area A that is enclosed by C. This section introduces the basic ideas behind the theorem and proceeds with explaining how the area A can be computed directly.

Curl: As will be seen later on, the mathematical definition of curl is needed for the definition of the theorem. The curl of a two-dimensional vector field $\vec{F}$ in terms of a line integral $\oint$ can be written as follows:

$$rot\vec{F}_{2D} = \lim_{|A_{x,y}| \to 0} \left(\frac{1}{|A_{x,y}|} \oint_C \vec{F} \cdot \vec{dr} \right) \tag{4.1}$$

Here, $\vec{F}$ stands for a two-dimensional vector field, $A_{x,y}$ represents an enclosed area centered at (x, y) and C is a counterclockwise oriented closed curve that encloses $A_{x,y}$. The equation can be understood as $(\oint_C F \cdot dr)$ representing fluid rotation along the boundaries of the given closed line C. It can be shown that the rotation following the contours of $A_{x,y}$ corresponds to the area (without proof), which is why it makes sense to divide this expression by the area to gain information about average rotation per unit. The result is information about the curl in an entire region, centered at the point (x, y) (compare figure 4.1). Apart from the integral representation of the curl, it can be written in vectorial form as

$$rot\vec{F}_{2D} = \nabla \times \vec{F} = \begin{pmatrix} \frac{\partial}{\partial x} \\ \\ \frac{\partial}{\partial y} \end{pmatrix} \times \begin{pmatrix} F_x \\ \\ F_y \end{pmatrix} = \frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \tag{4.2}$$

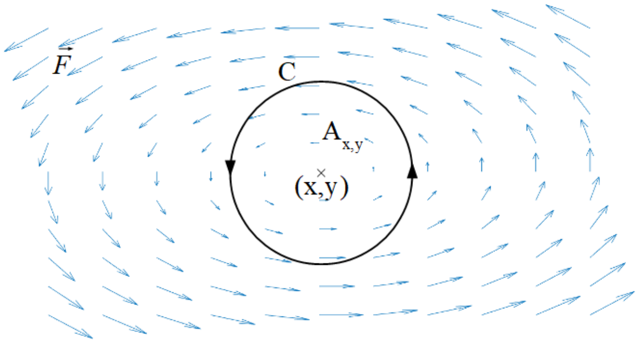
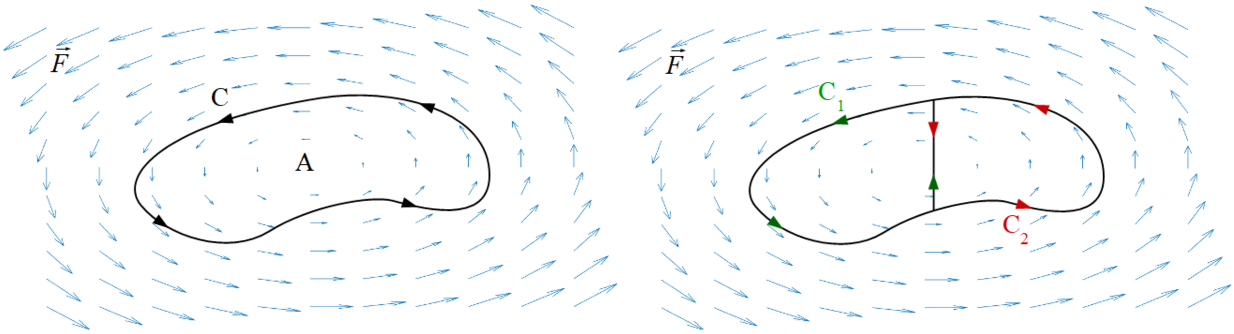


Figure 4.1: Curl

Theorem: The idea behind Green's theorem is to look at a predefined region A, enclosed by the curve C, that lies within a vector field $\vec{F}$, as is shown in figure 4.2 (a). The goal is to describe the rotation given by $\oint_C \vec{F} \cdot \vec{dr}$ by alternative means, replacing the line integral. Note, that this expression equals the previously mentioned curl, without the division by the area A.



(a) Initial Configuration

(b) Division

Figure 4.2: Green's Theorem

The idea is to divide the area A. Starting with one division, the curl for each separate area can now be obtained by $\oint_{C_1} \vec{F} \cdot \vec{dr}$ and $\oint_{C_2} \vec{F} \cdot \vec{dr}$ respectively. Graphically, this step can be seen in figure 4.2 (b). When defining counterclockwise orientation for both line integrals, the results of the line integrals along the cut have equal magnitude, but are oriented opposing and thereby canceling each other out. This leads back to the initial curve C

$$\oint_C \vec{F} \cdot \vec{dr} = \oint_{C_1} \vec{F} \cdot \vec{dr} + \oint_{C_2} \vec{F} \cdot \vec{dr} \tag{4.3}$$

Proceeding to add more cuts to the area does not change the correctness of this statement, as long as the positive orientation of the line integrals is defined equally. The equation for an increasing number of cuts N can therefore be written as

$$\oint_C \vec{F} \cdot \vec{dr} = \sum_{i=1}^{N} \oint_{C_i} \vec{F} \cdot \vec{dr} \tag{4.4}$$

As it is usual in infinitesimal calculus, the amount of cuts is increased, so that the area enclosed by the line integrals $\oint_{C_i}$ strives towards zero ($\lim\limits_{|A_{x,y}| \to 0}$). In this case, the expression $\oint_{C_i} \vec{F} \cdot \vec{dr}$, describing the rotation along the contours of the now infinitesimally small area $|A_{x,y}|$, is equal to equation 4.1 for the two-dimensional curl that has been multiplied by the area itself

$$\begin{aligned} \oint_{C_i} \vec{F} \cdot \vec{dr} &= \left[\lim_{|A_{x,y}|_i \to 0} \left(\frac{1}{|A_{x,y}|_i} \oint_{C_i} \vec{F} \cdot \vec{dr} \right) \right] \cdot |A_{x,y}|_i \\ &= rot\vec{F}_{2D,i} \cdot |A_{x,y}|_i \end{aligned} \tag{4.5}$$

For the curl along the entire contour C this leads to

$$\begin{aligned}\oint_C \vec{F} \cdot \vec{dr} &= \sum_{i=1}^{N} \left[\lim_{|A_{x,y}|_i \to 0} \left(\frac{1}{|A_{x,y}|_i} \oint_{C_i} \vec{F} \cdot \vec{dr} \right) \right] \cdot |A_{x,y}|_i \\ &= \sum_{i=1}^{N} rot\vec{F}_{2D,i} \cdot |A_{x,y}|_i\end{aligned} \tag{4.6}$$

what can be written in infinitesimal form as

$$\oint_C \vec{F} \cdot \vec{dr} = \int\int_A rot\vec{F}_{2D} dA \tag{4.7}$$

Spelling out the vector product on the left side

$$\vec{F} \cdot \vec{dr} = \begin{pmatrix} F_x \\ \\ F_y \end{pmatrix} \cdot \begin{pmatrix} dx \\ \\ dy \end{pmatrix} = F_x dx + F_y dy \tag{4.8}$$

and using the vectorial definition of the curl from equation 4.2 to replace the right side of the expression, the resulting equation

$$\oint_C (F_x dx + F_y dy) = \int\int_A \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right) dA \tag{4.9}$$

is known as Green's Theorem.

Area: This theorem is a mathematical tool to replace a double integral with a line integral, which can be used to simplify certain expressions. Other than that, a simple expression to determine the area A within an region enclosed by C can be derived. This is done by simply demanding that the solution of the theorem is equal to A and writing

$$\oint_C (F_x dx + F_y dy) = A \tag{4.10}$$

Focusing now on finding a way to fulfill this equation requires to look at Green's theorem in its entirety

$$\oint_C (F_x dx + F_y dy) = \int\int_A \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right) dA \tag{4.11}$$

Apparently the goal is determining values for F_x and F_y that results in

$$\left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y}\right) = 1 \tag{4.12}$$

so that

$$\oint_C (F_x dx + F_y dy) = \int\int_A (1) dA = A \tag{4.13}$$

with $dA = dxdy$. There are multiple sets of variables for F_x and F_y that satisfy this condition, nevertheless the values

$$F_x(x,y) = -\frac{y}{2} \tag{4.14}$$

$$F_y(x,y) = \frac{x}{2} \tag{4.15}$$

have proven to be easy to compute, while meeting the requirements stated above.

$$\int\int_A \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y}\right) dA = \int\int_A \left(\frac{\partial x}{\partial x 2} + \frac{\partial y}{\partial y 2}\right) dA \tag{4.16}$$

$$= \int\int_A \left(\frac{1}{2} + \frac{1}{2}\right) dA \tag{4.17}$$

$$= \int\int_A (1) dA = A \tag{4.18}$$

Having found values for F_x and F_y, the left side of the theorem can be transformed accordingly by inserting F_x and F_y.

$$\oint_C (F_x dx + F_y dy) = \oint_C \left(\frac{x}{2} dy - \frac{y}{2} dx\right) \tag{4.19}$$

$$\oint_C \frac{1}{2}(xdy - ydx) = A \tag{4.20}$$

This final equation can be used to determine the area within the boundary curve C.

4.2 Low-pass Filter

Since the presence of noise on signals has to be expected on every signal, the implementation of a noise reducing filter is at hand. The low-pass filter has already been mentioned within the introduction in section 3.2.3 defining noise. In signal processing, low-pass filters are used to remove signal components with frequencies above a predefined cutoff frequency f_c, or rather dampen them, while leaving signal components with frequencies beneath f_c unaffected. Within the signal processing unit LabView this type of filter is used in order to enhance signal quality and amplify the desired information.

By looking at the Bode-plot of a low-pass filter, displayed in figure 4.3, the general behavior of this signal processing step can be explained. Note, that instead of displaying in terms of the frequency f the angular frequency $\omega = 2\pi f$ is used. The frequency is plotted logarithmic on the horizontal axis, where it is common to reference characteristics using the cutoff frequency ω_c. This is the frequency where the real filter's amplitude response has a drop in magnitude of about 3 dB. The filter's influence changes within one decade above and one decade below this frequency.

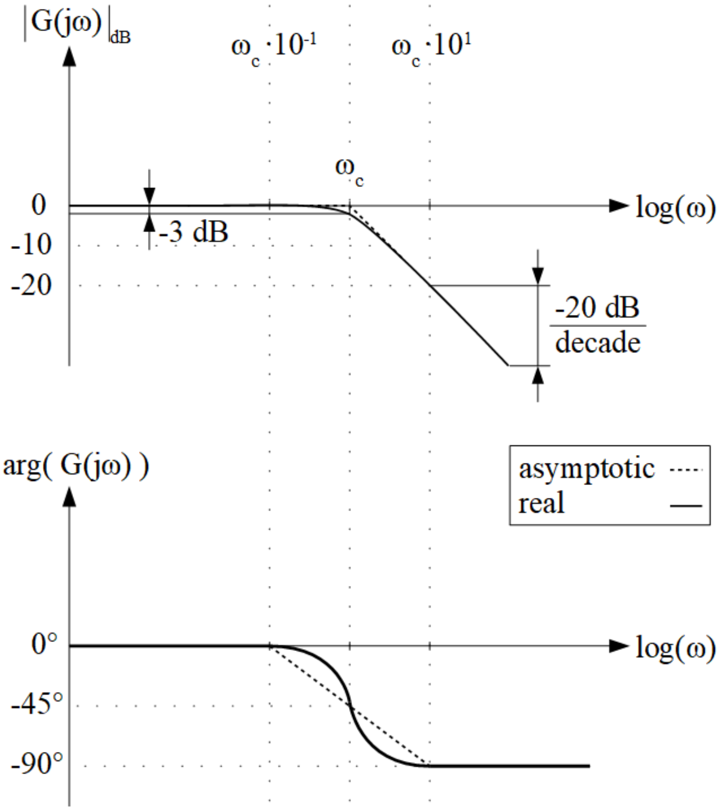


Figure 4.3: Bode plot of a single-pole low-pass filter

While frequencies below the frequency $\omega_c \cdot 10^{-1}$ show the unity gain (0 dB), every frequency above it has a negative gain. At frequencies above $\omega_c \cdot 10^1$ the gain decreases steadily with $\frac{20\ dB}{decade}$. To understand the effect of negative gain on a signal it can be thought of as reduction of signal amplitude. For example, a signal with the frequency ω_c is affected by the 3 dB negative gain that has been mentioned. As will be shown in the following analysis of the filter, this translates to a loss of signal amplitude of around 30%. All changes in magnitude come alongside with a change in phase. The shift in phase at the cutoff frequency is exactly -45 deg. Furthermore, note the different ways of portraying this filter: asymptotic and real. The description of the realistic filter behavior has just been given. The asymptotic form of displaying simplifies the behavior, but approximates the same fundamental ideas concerning cutoff frequency and shift in phase.

Mathematically this type of filter can be described using its transfer function that shows the frequency dependent change of amplitude and phase. This function is obtained by looking at a RC-circuit, displayed in figure 4.4. This circuit is the simplest form of a low-pass filter, often referred to as first-order Butterworth filter in respective literature. It consists of a resistor R and a capacitor C that are driven by the voltage U_{in}. When the amplitude of this driving voltage changes over time with a certain frequency, the effects described previously using the Bode-plot can be seen at the output voltage U_{out}.

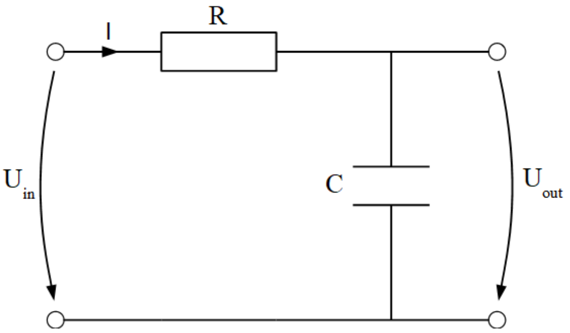


Figure 4.4: RC-low-pass circuit

Starting by applying Kirchhoff's second law to the apparent circuit, the equations

$$U_{in} = IR + IX_C \tag{4.21}$$

$$U_{out} = IX_C \tag{4.22}$$

are found. The transfer function $G(s)$ is defined as

$$G(s) = \frac{U_{out}(s)}{U_{in}(s)} \tag{4.23}$$

and can therefore be written as

$$G(s) = \frac{IX_C(s)}{IR + IX_C(s)} \tag{4.24}$$

with X_C being the frequency dependent resistance of the capacitor, according to

$$X_C(s) = \frac{1}{sC} \tag{4.25}$$

Reducing equation 4.24 for I results in

$$G(s) = \frac{\frac{1}{sC}}{R + \frac{1}{sC}} \tag{4.26}$$

$$= \frac{1}{1 + sRC} \tag{4.27}$$

or written in the frequency domain ($s = f\omega$)

$$G(j\omega) = \frac{1}{1 + j\omega RC} \tag{4.28}$$

Before phase and magnitude are calculated, the equation can be expanded by multiplying dividend and divisor with the complex conjugate of the divisor, which leads to the final equation for the frequency dependent transfer function

$$G(j\omega) = \frac{1}{1 + j\omega RC} \cdot \frac{1 - j\omega RC}{1 - j\omega RC} \tag{4.29}$$

$$= \frac{1 - j\omega RC}{1 + (\omega RC)^2} \tag{4.30}$$

For further calculation, the definition of the cutoff frequency f_c in electronic circuits can be given as

$$f_c = \frac{1}{2\pi RC} \tag{4.31}$$

or written in terms of the angular frequency ω

$$\omega_c = 2\pi f_c = \frac{1}{RC} \tag{4.32}$$

As has been mentioned before, signal components with that exact frequency show a loss in amplitude of 3 dB and a shift in phase of -45 deg.

Phase: In complex numbers the phase is defined as

$$tan\varphi = \frac{Im}{Re} \tag{4.33}$$

where the fraction in case of the provided transfer function in equation 4.30 is

$$\frac{Im}{Re} = \frac{\frac{-\omega RC}{1+(\omega RC)^2}}{\frac{1}{1+(\omega RC)^2}} = -\omega RC \tag{4.34}$$

leading to a phase at ω_c of

$$arg(G(j\omega_c)) = arctan(-\omega_c RC) = arctan(-\frac{RC}{RC}) = arctan(-1) = -45^\circ \tag{4.35}$$

Magnitude: The magnitude of a complex number is defined by

$$|G(j\omega)| = \sqrt{Re^2 + Im^2} \tag{4.36}$$

Applying this equation to the complex form of the transfer function 4.30, the result is

$$|G(j\omega)| = \sqrt{\left(\frac{1}{1+(\omega RC)^2}\right)^2 + \left(\frac{-\omega RC}{1+(\omega RC)^2}\right)^2} \tag{4.37}$$

$$= \sqrt{\frac{1+(\omega RC)^2}{(1+(\omega RC)^2)^2}} \tag{4.38}$$

$$= \frac{1}{\sqrt{1+(\omega RC)^2}} \tag{4.39}$$

Substituting the general ω with the cutoff value ω_c, the absolute value for the transfer function 4.30 at the cutoff frequency is

$$|G(j\omega_c)| = \frac{1}{\sqrt{1+(\omega_c RC)^2}} = \frac{1}{\sqrt{1+(\frac{RC}{RC})^2}} = \frac{1}{\sqrt{2}} \tag{4.40}$$

which, in order to be displayed in the Bode-plot, equals

$$|G(j\omega_c)|_{dB} = 20log(|G(j\omega_c)|) \approx -3[dB] \tag{4.41}$$

4.3 Peak Detection

After reducing noise using the just described low-pass filters, the measurement data still provides more than the required information. In the present case of radial rotor-casing clearance measurement using capacitive sensors, the signal contains additional information about the time of arrival of a certain component or the components shape. Since those details are not required for the underlying purpose, they can be discarded and the signal can be reduced to the essential components. The methods applied for that purpose within LabView is peak detection, which manages to find the location and amplitude of localized maxima and minima within data sets. Note, that there does not exist a standardized, or superior method for that purpose. The performance of an algorithm always depends on the provided inputs and usually has advantages and disadvantages. For given purpose, finding methods for the detection of local maxima within data stored in one-dimensional arrays is the matter at hand. In case of an array containing only one maximum, the most straight forward solution appears to be to apply a sorting algorithm that sorts the data from biggest to smallest value and returns the first value in the array. For this simple case this might actually be the fastest solution, but apart from not retrieving the index of this peak, the result is already corrupted if multiple local maxima are present within the data. This thought experiment can be seen in figure 4.5. In order to make peak detection independent from the data input, another approach has to be chosen.

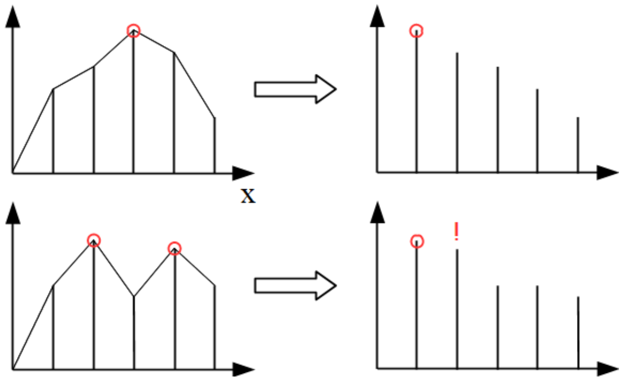


Figure 4.5: Sorting algorithm

The solution presented here takes into account the specific nature of data that can be expected in clearance measurement. Usually, measurement data is affected by noise, which leads to a significant amount of local peaks and valleys. Therefore some kind of data smoothing possibilities have to be included into the detection algorithm. A common approach for problems with present specifications is to fit curves into the data, differentiating the resulting equation and checking for extrema. The processing software LabView contains predefined routines performing these calculations. Nevertheless, the fundamental principle is outlined here. Note, that only curve fitting is described at this point, since checking the resulting equations for extrema by differentiating does not require further explanation. The explanation is based on work done by Miller (1992).

Linear Fit using Least Squares: The method of 'least squares' is an approach used for finding functions, describing a given set of data points. In short, it strives to find the coefficients of a function with predefined order that fits the provided set of data points in terms of minimizing the squared, resulting difference between the actual value and the fitted value. This can easier be understood by looking at figure 4.6, where the data is described by the means of a linear function f(x).

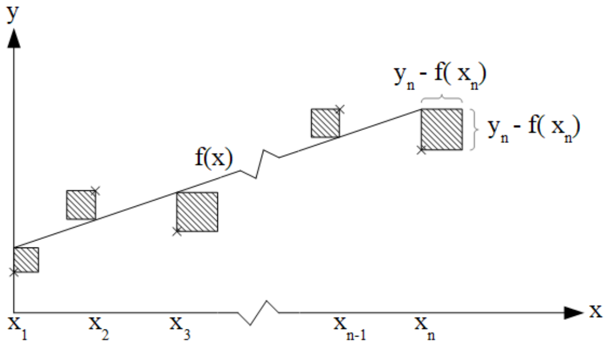


Figure 4.6: Linear function f(x) fit into a set of data (x_i,y_i) using 'Least Squares'

Let the goal be to fit a linear function $f(x) : y = a_1x + a_0$ into a set of data. The resulting error between the actual value y_i and the function y can be written as

$$\triangle y_i = y_i - f(x) = y_i - (a_1x_i + a_0) \tag{4.42}$$

This expression can be squared and renamed to e_i, representing the remaining error as

$$e_i = [y_i - (a_1x_i + a_0)]^2 \tag{4.43}$$

Adding up the errors for all N values in the present data set gives

$$E = \sum_{i=1}^{N} [y_i - (a_1x_i + a_0)]^2 \tag{4.44}$$

This equation is representative of the overall error that is made when fitting the curve dependent on the parameters a_1 and a_0. To find values for these parameters that minimize E the statements

$$\frac{\partial E}{\partial a_1} = 0 \tag{4.45}$$

$$\frac{\partial E}{\partial a_0} = 0 \tag{4.46}$$

have to be fulfilled. Written in full, the equations are

$$\frac{\partial E}{\partial a_1} = \sum_{i=1}^{N} 2[y_i - (a_1 x_i + a_0)] \cdot (-x_i) = 0 \tag{4.47}$$

$$\frac{\partial E}{\partial a_0} = \sum_{i=1}^{N} 2[y_i - (a_1 x_i + a_0)] \cdot (1) = 0 \tag{4.48}$$

Dividing both equations by 2, they can be rewritten as

$$\sum_{i=1}^{N} \left(\sum_{i=1}^{N} x_i^2\right) a_1 + \left(\sum_{i=1}^{N} x_i\right) a_0 = \sum_{i=1}^{N} x_i y_i \tag{4.49}$$

$$\sum_{i=1}^{N} \left(\sum_{i=1}^{N} x_i\right) a_1 + \left(\sum_{i=1}^{N} 1\right) a_0 = \sum_{i=1}^{N} y_i \tag{4.50}$$

Displaying these statements as a system of equations that is transformed to solve for a_1 and a_0 explicitly gives

$$\begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} 1 \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^{N} x_i y_i \\ \sum_{i=1}^{N} y_i \end{pmatrix} \tag{4.51}$$

By solving this set of equations, the values of a_1 and a_0 fulfill the desired criteria of minimizing the sum of combined areas of the squares shown in figure 4.6. Note, that in order to be mathematically accurate, the direction of the extremum has to be proven. The criteria for the found extremum to be in fact a minimum is

$$\frac{\partial^2 E}{\partial a_1^2} > 0 \tag{4.52}$$

$$\frac{\partial^2 E}{\partial a_0^2} > 0 \tag{4.53}$$

Only when both equations apply, the calculation can be considered conclusive.

Polynomial Fit using Least Squares: Apart from a linear function, also known as a first-order polynomial $y = a_1 x + a_0$, many other useful functions can be fit to approximate the data. The same process can be repeated for a k-dimensional polynomial $y = a_k x^k + a_{k-1} x^{k-1} + ... + a_1 x + a_0$ in order to fit a set of data with $k+1$ values. This relation is easily derivable from the fact that a k-dimensional function leads to $k+1$ unknown parameters a_i and thus to a system of $k+1$ equations, such as shown previously for a linear function:

$$\begin{pmatrix} a_k \\ \vdots \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^N x_i^{2k} & \cdots & \sum_{i=1}^N x_i^{k+1} & \sum_{i=1}^N x_i^k \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^N x_i^{k+1} & \cdots & \sum_{i=1}^N x_i^2 & \sum_{i=1}^N x_i \\ \sum_{i=1}^N x_i^k & \cdots & \sum_{i=1}^N x_i & \sum_{i=1}^N 1 \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^N x_i^k y_i \\ \vdots \\ \sum_{i=1}^N x_i y_i \\ \sum_{i=1}^N y_i \end{pmatrix} \tag{4.54}$$

Solving this generalized equation allows for fitting a polynomial of any desired order into the data.

4.4 Unambiguous Definition of a Circle

As has been mentioned before, after simulating the signal in Matlab and processing it using the monitoring routine in LabView the combined clearance information of multiple sensors can be used to approximate the rotors absolute position within the casing. In order to limit the required sensors this thesis assumes the shape connecting the rotor's tip to be a perfectly circle. This means that for shrouded and bladed rotor configurations three sensors are sufficient to inscribe that circle into the tips. Figure 4.7 shows the proposed idea.

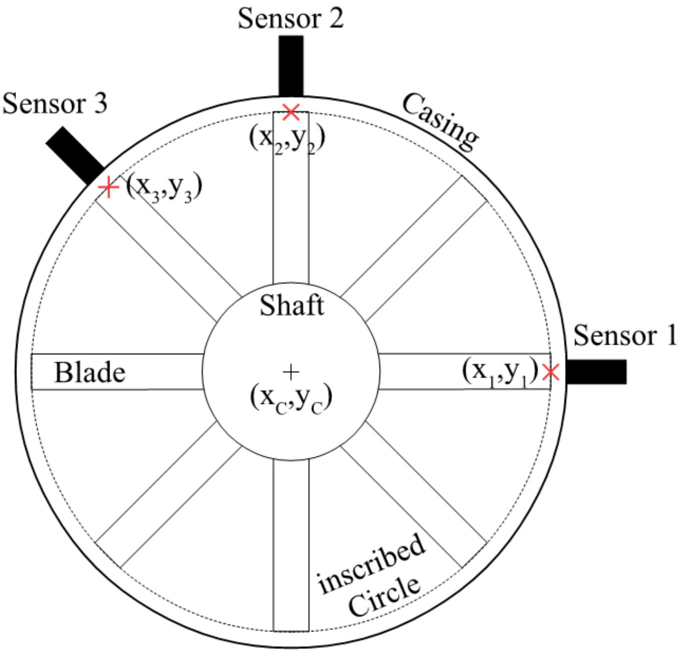


Figure 4.7: Circle inscribed into the tips of a bladed rotor using three sensors

The following pages provide a fast and effective solution for the definition of a circle using three arbitrary points on its circumference, derived from an idea presented at the AmBrSoft Website. In two-dimensional space, the radius of a circle centered at (x_C, y_C) can be calculated by

$$r = \sqrt{(x - x_C)^2 + (y - y_C)^2} \tag{4.55}$$

By squaring the equation, simplifying the expression and introducing a functional parameter

A

$$A \cdot r^2 = A \cdot (x^2 - 2xx_C + x_C^2 + y^2 - 2yy_C + y_C^2) \tag{4.56}$$

can be written and transformed to

$$Ax^2 + Ay^2 + Bx + Cy + D = 0 \tag{4.57}$$

where $A = const.$, $B = -2Ax_C$, $C = -2Ay_C$ and $D = A(x_C^2 + y_C^2 - r^2)$. To solve this problem unambiguously, three arbitrary points on the circle are necessary. Inserting these into the equation above results in a system of four equations

$$\begin{bmatrix} x^2+y^2 & x & y & 1 \\ x_1^2+y_1^2 & x_1 & y_1 & 1 \\ x_2^2+y_2^2 & x_2 & y_2 & 1 \\ x_3^2+y_3^2 & x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.58}$$

Note, that the first row of this system represents the general equation $Ax^2 + Ay^2 + Bx + Cy + D = 0$. This is done in order to acquire a 4×4 matrix that can be solved in terms of a determinant

$$det(M) = \begin{vmatrix} x^2+y^2 & x & y & 1 \\ x_1^2+y_1^2 & x_1 & y_1 & 1 \\ x_2^2+y_2^2 & x_2 & y_2 & 1 \\ x_3^2+y_3^2 & x_3 & y_3 & 1 \end{vmatrix} = 0 \tag{4.59}$$

by making use of the Laplace expansion, which is given for the expansion for a row by

$$det(M) = \sum_{i=1}^{n} (-1)^{i+j} \cdot a_{ij} \cdot det(M_{ij}) \tag{4.60}$$

Here, i stands for the row to expand from, j for the column and M_{ij} for the $(n-1) \times (n-1)$

sub-matrix of M. Applying this operation the result for $i = 1$ is

$$(x^2+y^2)\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} - (x)\begin{vmatrix} x_1^2+y_1^2 & y_1 & 1 \\ x_2^2+y_2^2 & y_2 & 1 \\ x_3^2+y_3^2 & y_3 & 1 \end{vmatrix}$$
$$+(y)\begin{vmatrix} x_1^2+y_1^2 & x_1 & 1 \\ x_2^2+y_2^2 & x_2 & 1 \\ x_3^2+y_3^2 & x_3 & 1 \end{vmatrix} - (1)\begin{vmatrix} x_1^2+y_1^2 & x_1 & y_1 \\ x_2^2+y_2^2 & x_2 & y_2 \\ x_3^2+y_3^2 & x_3 & y_3 \end{vmatrix} = 0 \tag{4.61}$$

The sub-determinants now represent the coefficients A, B, C and D from equation 4.57.

$$A = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}, \quad B = -\begin{vmatrix} x_1^2+y_1^2 & y_1 & 1 \\ x_2^2+y_2^2 & y_2 & 1 \\ x_3^2+y_3^2 & y_3 & 1 \end{vmatrix},$$

$$C = \begin{vmatrix} x_1^2+y_1^2 & x_1 & 1 \\ x_2^2+y_2^2 & x_2 & 1 \\ x_3^2+y_3^2 & x_3 & 1 \end{vmatrix}, \quad D = -\begin{vmatrix} x_1^2+y_1^2 & x_1 & y_1 \\ x_2^2+y_2^2 & x_2 & y_2 \\ x_3^2+y_3^2 & x_3 & y_3 \end{vmatrix} \tag{4.62}$$

After obtaining the values for the functional parameters, equation 4.57 can be transformed and used to solve

$$x_C = -\frac{B}{2A} \tag{4.63}$$
$$y_C = -\frac{C}{2A} \tag{4.64}$$
$$r = \sqrt{\frac{B^2 + C^2 - 4AD}{4A^2}} \tag{4.65}$$

This is the center position in two-dimensional space and the radius of the circle that has been fit into the three points on its circumference.

5 Matlab Signal Generation

In order to be able to develop a monitoring unit, the signal containing the clearance information has to be simulated. As has been discussed in the previous chapters, the signal used in this thesis is based on a mathematical model. This approach allows for cheaper development and more flexible testing, while permanently being able to verify the processing. This chapter starts with introducing the basic idea behind the independent machine and measurement system model that are implemented using Matlab. Due to the anticipation of variations in blade geometry, three different machine models are created:

1. rotor with shrouded blades
2. rotor with simple blade tips
3. rotor with blades that have squealer tips

This is followed by the description of the measurement system model. The measurement chain is reduced to the modeling of a capacitor sensor powered by a direct current circuit, as has been described in chapter 3. Note, that for each of the three mentioned rotor geometries slight variations in the approach for the signal generation are taken, which is the reason for dealing with every geometry in a separate routine. The description of the calculation steps for each of the signal generation routines takes place after the introduction of the models. This is followed by the conclusion of this chapter with the validation of the generated signals using real world measurement signals and a study of important parameters.

5.1 Modeling Process

By introducing independent machine and sensing models the resemblance to actual measurement can be contained, while being able to individually optimize system parameters on machine and measurement side. The goal is to create models that allow to nullify certain influences, in order to be able to detect signal-changing aspects. While the previous chapters have given a glimpse on the range of all influential parameters, this section narrows down the domains more detailed and specifies the available settings.

5.1.1 Turbomachine Model

The machine model is created in accordance with section 2.4.1. There, a plane intersecting the rotor perpendicular to the axis connecting the bearings has been introduced. This plane is displayed in figure 5.1 on the left side. The derived model is a geometric replication of a rotor moving within the casing and can be seen in figure 5.1 on the right. The center of the casing is seen as center of a coordinate system (0,0). The two essential referencing parameters for the shaft's location within this two-dimensional plane are:

- Offset: The stationary component of the shaft's centerline displacement within the casing.
- Eccentricity: The current distance between the stationary offset and the absolute position.

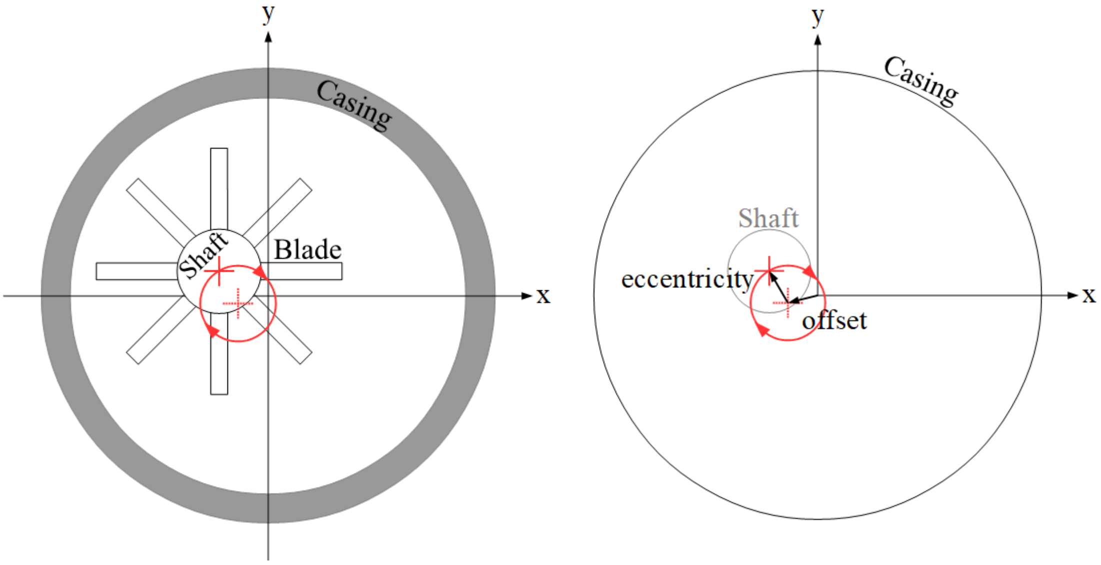


Figure 5.1: Machine model creation

After defining the center of the casing to be the center of a coordinate system, the first modeling step is is to describe the location of the shaft. In accordance with section 2.4.1, where the dynamic movement due to rotational velocity has been highlighted, a variable called eccentricity is introduced. It represents the current value of displacement amplitude of the shaft's center line from a stationary offset within the casing. Apart from the dynamic displacement, this centerline can be diverting from the center of the casing due to stationary misalignment of the shaft within the casing, which is respected with the introduction of the variable offset. The detailed portray of the model and all its parameters can be seen in figure 5.2. The combination of those two parameters is used to track the center position of the shaft at any moment according to equation

$$Shaft\ Center = \begin{pmatrix} Offset_x \\ \\ Offset_y \end{pmatrix} + Eccentricity \begin{pmatrix} cos(\theta) \\ \\ sin(\theta) \end{pmatrix} \tag{5.1}$$

Note the presence of the angle θ. The model computes the rotor's rotation using an incremental step $\triangle\theta$, allowing it to be used as degree of freedom.

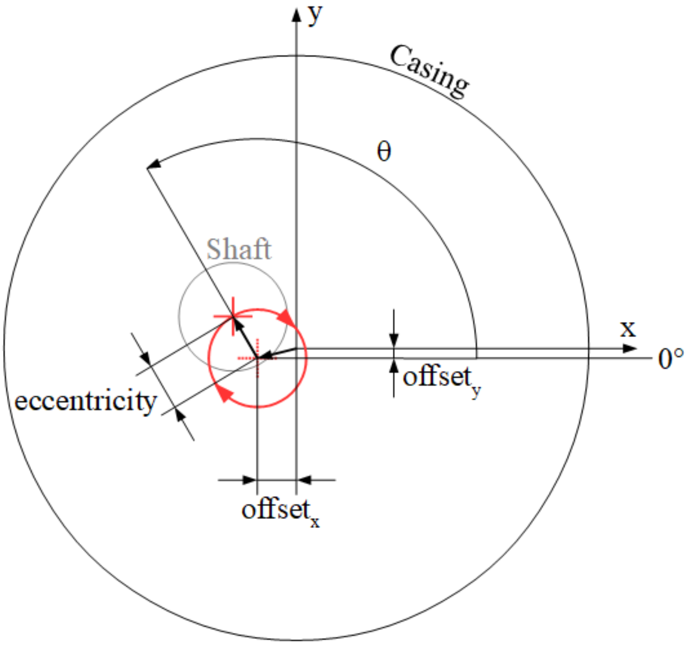


Figure 5.2: Two-dimensional machine model

At this point, differences in cascade design have to be respected. Shrouded blade rows require the definition of a parameter describing the radius of the outer point of the shroud, while bladed configurations introduce blade lengths and - in case of squealer tips - parameters defining the additional tips. In any case, the blade row is defined as rigid in any direction. This allows for directly connecting the individual blade geometries to the shaft and thereby coupling the blade's movement directly to equation 5.1 shown above. The individual modeling steps for each of the introduced geometries is presented more detailed later on. Note, that in order to refine the model to simulate machine behavior more precisely, following steps can be taken:

- Imperfections concerning circularity of casing, shaft or shroud are implemented by simply defining a dependency on θ for each parameter. A function can be defined instead of the constant value shown above. This allows for the modeling of any deformation without having to alter the fundamental equation 5.1.
- Deviations of blade length due to manufacturing tolerances can be modeled by specifying individual lengths for each blade that is attached to the shaft. In the same way, time dependent functions can be used to model the heat deformation due to changing operational temperatures of the turbomachine.
- Dynamic shaft behavior, as discussed in section 2.4.1, results in varying eccentricity. Due to the incremental nature of the calculations, this parameters can be defined as functions of θ instead. This allows for complex behavior, like the dependency on angular velocity, to be portrayed, while still being able to use equation 5.1.

The nature of the simulation with separated models for sensing and machine allows for the implementation of successive details, like the examples shown above, including oscillation and three dimensional rotor movement. However, the goal of this thesis is not to model turbomachine behavior, which is why the accuracy provided by equation 5.1 suffices. Any additional details do not improve the development of the signal processing unit, since dependencies would increase and isolated parameter studies could be compromised.

5.1.2 Measurement System Model

After providing a machine model, the second aspect of signal generation has to be addressed in this section. For the measurement system model, the main requirement shall be the independence of the machine model, in order to make both expandable without interfering with the performance of the entire simulation.

System Selection: In chapter 3 the selection of capacitive sensing in combination with a direct current circuit for the underlying thesis has been mentioned. The reasons for this choice are the simplicity of the sensing principle and the availability of hardware for further development steps. The model does not strive to accurately represent the entire measurement chain, nor to model all influential parameters that have been discussed prior. The goal is much more, to provide a system that reacts to the machine model in a way that is comparable to the real-world equivalent of the mentioned principle. Generally speaking, the output shall be a signal that contains the informations inserted into the simulation via machine parameters and replicate real world signals originating from that sensor type.

Model: Looking back to chapter 3, the concept of capacitive sensing was presented as the pairing of two plates, separated by a non-conductive layer. In clearance measurement processes one capacitor plate is the sensor, which is supplied through any of the mentioned circuit methods, whilst the other plate is the blade. Equation 3.4 provides the governing equation for capacitive sensing:

$$\triangle C = \frac{\triangle A}{\triangle d}\epsilon_0\epsilon_r V$$

In accordance with the dc-based circuit method, the driving voltage V is assumed to be constant. Additionally, the same assumption is made for the dielectric constants ϵ_0 and ϵ_r. This leads to a simplified equation, since the constants V, ϵ_0 and ϵ_r can be defined to be 1, leading to the expression

$$C \approx \frac{A}{d} \tag{5.2}$$

The assumption of $V = \epsilon_0 = \epsilon_r = 1$ can be justified, by interpreting them as scaling factors. When it comes to deriving the information from the signal, these factors only shift the entire dependency signal-to-distance and thus can be neglected. A and d on the other hand are a result of the interaction of the rotor with the sensing area, which is therefore modeled in reference to the real world sensor. The applied concept can be seen in a two-dimensional plane in figure 5.3. The displayed electrical field has a strong non-linear character outside of its so called linear sensing range, which in case of capacitive sensors lies usually within the range of the sensors diameter (stated in chapter 3). Therefore, the complicated structure is broken down to modeling the linear sensing range, approximated by the red rectangle shown in the figure. The entire sensing principle - the interaction between sensing area and rotor - is reduced from a volumetric problem in three-dimensional space to a problem in two dimensional space, where the overlapping area of blade and sensor are chosen to represents the area A of the plate capacitor. The distance between the plates d is assumed to be the distance between sensor tip and geometric center of the overlapping polygon A. For every angular computational step $\triangle\theta$, the overlapping rotor-sensor area is determined and a resulting signal is computed, according to

$$\frac{Q}{V} = C = Signal \approx \frac{A}{d} \tag{5.3}$$

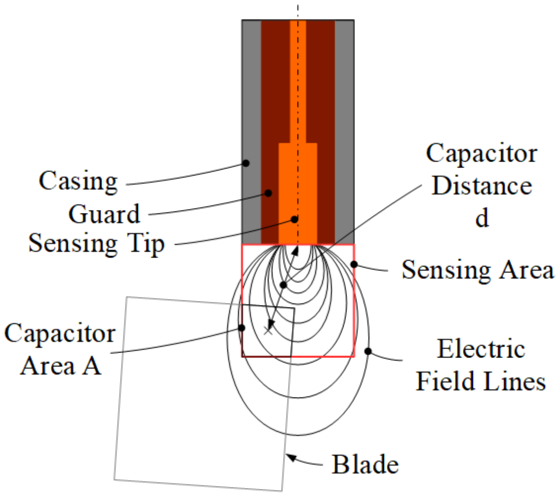


Figure 5.3: Capacitive sensing model

This resembles the changing capacitance caused by a passing blade in real world measurement. While the direct current circuit supplies the sensor with constant voltage V, the passing blade causes the movement of charge Q. This is amplified by a charge amplifier and translated by the electronics to a signal that depends on the changing area A and distance d of the capacitor pairing. The question about accuracy might arise when neglecting the non-linear area and assuming a square sensor field. Fortunately, most accuracy is needed when the rotor and the sensor are interacting the strongest, resulting in a peak capacitance. This is the case when the rotor component and the sensor are the closest and thereby the clearance is the smallest. For that reason sensors are bought according to the expected clearance in real-world measurement, leading to the peak being located within their linear range. This model is not an accurate representation of capacitive sensing, just like the machine model does not claim to display real-world behavior in every aspect. Nevertheless, it fulfills the requirements of encoding clearance information into a signal, in a way that resembles real-world capacitive sensor signals. Due to that the modeling of interaction between capacitor plates is not refined any further.

5.2 Matlab Routines

As has been stated at the beginning of this chapter, the availability of different rotor geometries has to be anticipated. This results in three different signal generation routines, all alternating in certain details. While the machine models only differ in portrayed geometry for shrouded, simple bladed and squealer bladed rotors, the true differences are caused by the difficulties in determining the radial clearance between sensor tip and rotor. The most straight forward method of calculating the radial rotor-casing clearance is to establish a model of the rotor geometry as well as its position within the casing and then compute the geometric distance between the closest rotor element to the sensor tip. The problem with this method is the detection of this element. From the point of view of an inwards facing sensor, mounted on the casing, with limited detection range, detected components can vary.

First, taking a rotor with a shroud. The shroud is constantly within vicinity of the sensor and is therefore picked up on by the sensor, resulting in a continuous change of signal amplitude. This signal always relates to the distance between sensor tip and shroud underneath the sensor. That is why the calculation based on geometric connections is possible. Generally, this means that the routine for this rotor geometry uses the machine model but not the measurement model. The clearance is calculated using the geometric dependencies as will be shown in the dedicated section for the shroud signal generation routine.

On the other side, bladed rotor configurations are available. In this case, where there are times where blades are interacting with the sensor and times where no coupling between the capacitor plates exists, the resulting change in capacitance caused by a passing blade is a peak. The problem with calculating clearance based on geometry lies within determining which rotor element is relevant, which is why it has proven to be inconvenient to try to describe a geometric distance for bladed rotors. Therefore, the established model for the measurement system is applied. The resulting interaction between machine and measurement model is treated as output signal. This leads to the encryption of the desired clearance information according to equation 5.3

$$\frac{Q}{V} = C = Signal \approx \frac{A}{d}$$

The output signal of the three routines are subsequently passed to LabView, where the signal processing takes place. In case of the shroud signal, the clearance data is already directly available in the signal, since it is in fact the geometric distance between sensor tip and the shroud underneath the sensor. The bladed signals show the change in capacitance caused by the blade-sensor interaction, which leads to the need of deriving the clearance from the signal using linear interpolation within LabView.

5.2.1 Shrouded Blades

As has been stated, the Matlab signal generation routine for shrouded rotors does not require using the measurement system model introduced in section 5.1.2. The simplified approach is due to the comfortable calculation of the true clearance between a sensor, mounted plane with the casings inner wall and the shroud. Figure 5.4 shows the sensor mounted on the casing, interacting with the shroud that shall be simulated. The sensor is assumed to be unable to detect the blades underneath the shroud so that the output signal is continuous, comparable to measuring a shaft. This assumption is made due to the fact that, if blades would be detected, there would be no structural difference in output signals between this routine and the one dealing with unshrouded blade geometry. Therefore, the routine uses the machine model's geometry dependencies to determine the clearance. This true value of the radial shroud-casing clearance is treated as output and is subsequently passed to LabView for processing.

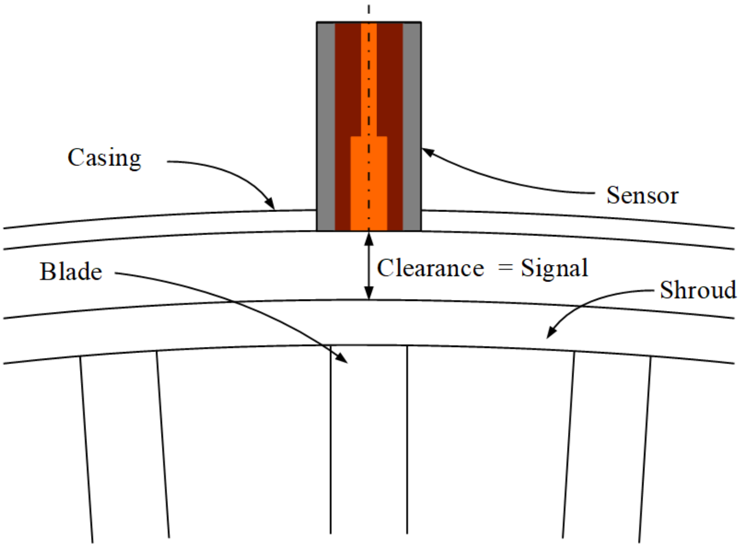


Figure 5.4: Shrouded routine clearance signal

5.2.1.1 Initialization

The first step is the setup of the geometric parameters within Matlab for the machine model that is displayed in figure 5.5. The defined parameters are:

- Radius casing R_{Casing}
- Radius Shaft R_{Shaft}
- Radius Shroud R_{Shroud}
- Offset$_x$,Offset$_y$
- Eccentricity
- Sensor Count
- Sensor Position δ

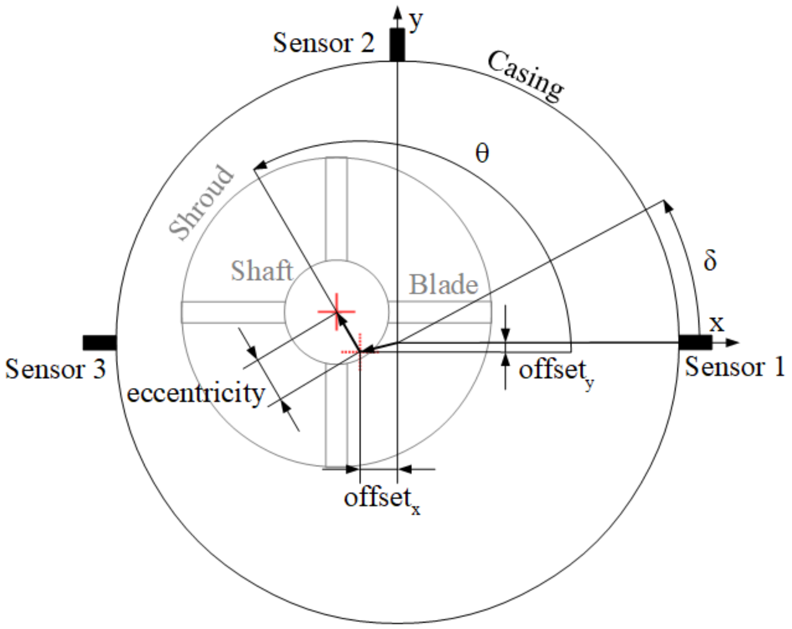


Figure 5.5: Model of shrouded blade configuration

Additionally, the spatial stepping size $\triangle\theta$ in form of steps in angular direction is defined. Note, that the entire routine assumes the stationary casing center to be at [0,0] within a two-dimensional plane. Every deviation of the shaft results in a displacement in two coordinates [x,y]. After the definition of the underlying geometry parameters, the initialization of the computational domain is performed, starting with preparing the user interface by setting up plotting areas within a figure and data matrices that are predefined in order to save computational power that would otherwise be needed for resizing said matrices. Subsequently, the connection to a Transmission Control Protocol (TCP) server is established in order to broadcast the signal. Finally, the setup is concluded by the creation of an iterative instance that loops for a predefined amount of times and every time increases the stepping parameter by $\triangle\theta$: the main loop.

5.2.1.2 Main Loop

The first step within each main loop iteration is to establish the location of the shaft's center within the casing's coordinate system. For that purpose the equation 5.1 (introduced within the machine model) is used:

$$Shaft\ Center = \begin{pmatrix} Offset_x \\ \\ Offset_y \end{pmatrix} + Eccentricity \begin{pmatrix} cos(\theta) \\ \\ sin(\theta) \end{pmatrix}$$

In accordance with the model presented (rigid rotor), this is the center point of the rotor. In the present case of a shroud and based on the assumption that no blades can be detected by the sensor, the modeling of the blades can be spared and instead only the shroud can be placed. This is done in two steps: first a shaft circle is placed. Subsequently, the shroud circle is placed concentrically with the shaft. The results are plotted into the user interface at this point, allowing the user to track the rotor's location and verify output signals.

In the other routines, this is the place where the machine's interaction with the sensing area would be computed. Here, due to the simplistic model, a shortcut is used. Instead of setting up a sensor field and checking for interaction, a geometric distance is calculated and used as output signal. Once per iteration, the signal is evaluated, based on the geometry shown in figure 5.6. In terms of trigonometry, the clearance at the position of sensor 1 can be calculated by

$$Signal = Clearance = Radius\ Casing - Shaft\ Center_x - L \tag{5.4}$$

The location of the shaft's centerline has already been computed, resulting in the coordinates of the center $(Shaft\ Center_x, Shaft\ Center_y)^T$. The length L is found by

$$L = Radius\ Shroud \cdot sin(\gamma) \tag{5.5}$$

where the angle γ is

$$cos(\gamma) = \frac{Shaft\ Center_y}{Radius\ Shroud} \tag{5.6}$$

After determining the clearance, it is broadcasted via TCP in order to be picked up by the processing unit and the main loop re-iterates.

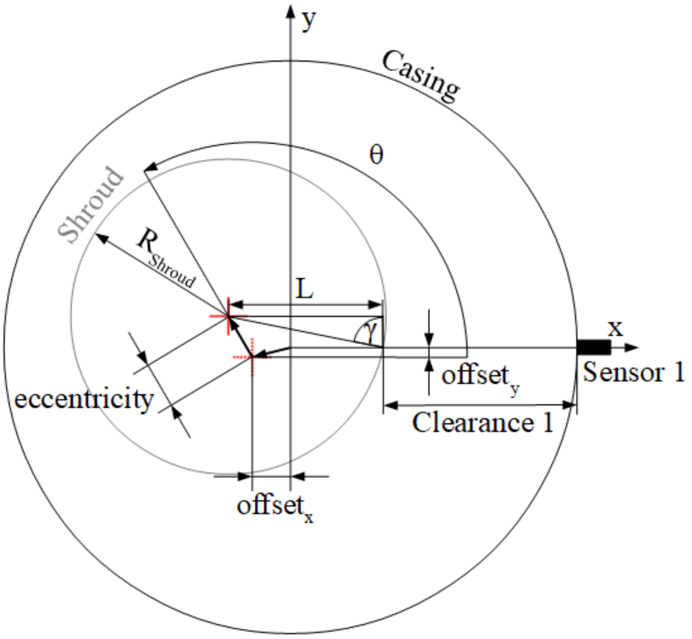


Figure 5.6: Geometry for determining signal at shrouded configuration with $\delta = 0°$

5.2.1.3 Output Signal

The main loop iterates until a final value for the angle θ is reached while incrementing it each step for $\triangle\theta$. This results in the shaft orbiting its stationary ’offset’ with the specified value of ’eccentricity’. Within every iteration the clearance underneath each defined sensor is determined using the described geometric dependencies and sent to LabView via TCP. For that purpose all clearances are attached to a string and broadcasted to the TCP server that is hosted by LabView. The other routines build upon the just discussed calculation sequence by modeling alternative machine geometry and simply refining the details for signal generation in the main loop.

In case of simulating the displayed setup in figure 5.5 with an offset of [1,1] mm and a constant eccentricity of 1.5 mm, the output signal for an entire revolution from $\theta = 0$ to $\theta = 2\pi$ for sensors placed at 0, 90 and 180 deg can be seen in figure 5.7. The figure shows the change in radial shroud-casing clearance amplitude in mm underneath the three sensors. Note, that only the output signal of the shroud routine directly displays clearance amplitude in mm, due to the fact that only here a geometric distance was computed. As is apparent, the circular orbit (eccentricity = const.) of the shaft causes a sine wave in output signal. The clearances at $\theta = 0$ to $\theta = 2\pi$ are equal since the shaft is back in its initial position after a full revolution. Furthermore, the clearance underneath sensor 3 appears to always be bigger than underneath the other sensors. This is caused by the stationary offset of the rotor to [1,1] within the casings referencing system. The orbit with the radius 1.5 mm (eccentricity) is centered at [1,1], which means that the rotor is constantly closer to sensor 1 and 2.

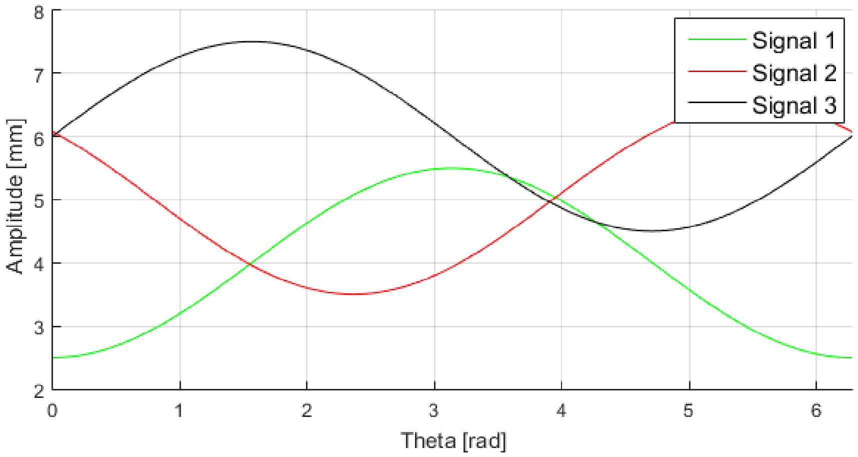


Figure 5.7: Clearance signals of a shrouded machine of three sensors (offset = [1,1] mm & eccentricity = 1.5 mm)

5.2.2 Simple Blades

The direct calculation of the clearance done within the shroud signal routine was only possible due to clear geometric dependencies for the clearance. As soon as the blades are not covered by a shroud, the connection between clearance and signal are not as apparent anymore, which is why the measurement system model has to be used in order to ensure realistic signal behavior. The simple blade signal routine starts by setting up the geometry, interface, data matrices and TCP server connection and subsequently enters a main loop. This loop is iterated while incrementing θ and for every iteration a output signal is created.

5.2.2.1 Initialization

The initialization of the routine starts by defining the underlying geometry parameters. Note, that the only difference between the shrouded and bladed design is the absence of the shroud. This circumstance is visualized in figure 5.8. The derivable geometry parameters are

- Radius Casing R_{Casing}
- Radius Shaft R_{Shaft}
- Offset$_x$,Offset$_y$
- Eccentricity
- Blade Count

- Sensor Count
- Sensor Position δ
- Blade Length L_B
- Blade Width W_B
- Sensor Width W_S

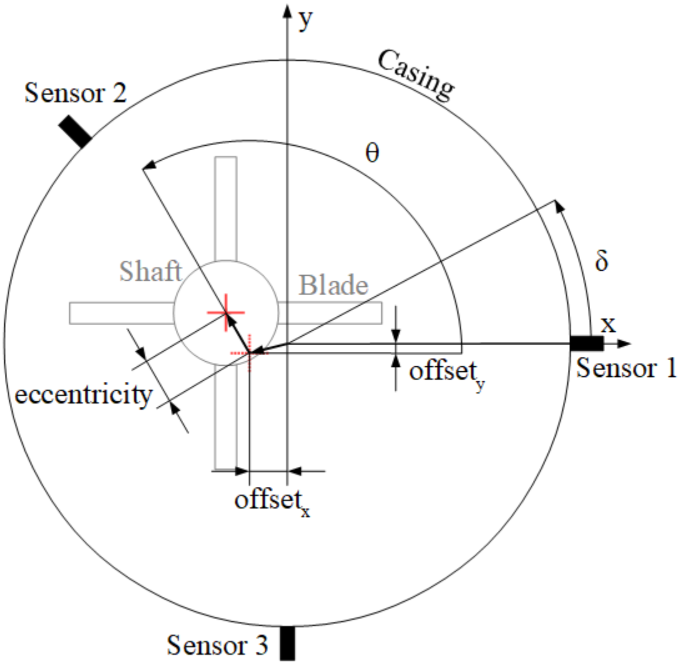


Figure 5.8: Geometry for simple blade configuration

Since this routine relies on the interaction between blades and a sensing area, the first step is to place the square sensing areas within the coordinate system. The location depends on the specified amount of sensors and their angular location δ. Additionally, the angular distribution of the blades is calculated by equally spacing them over the range of 360 deg using the expression:

$$\alpha = \frac{360}{Blade\ Count} \tag{5.7}$$

This concludes the setup of the geometry parameters. Subsequently the same initialization steps introduced in the previous routine (interface, data matrices, etc.) are conducted and the main loop is created.

5.2.2.2 Main Loop

The first steps within each main loop iteration are similar to the one's taken for shrouded rotors. The only difference is a result of the necessity to model the blades in the present case.

Shaft Center: Initially the shaft's center position within the two-dimensional plane is evaluated by equation 5.1:

$$Shaft\ Center = \begin{pmatrix} Offset_x \\ Offset_y \end{pmatrix} + Eccentricity \begin{pmatrix} cos(\theta) \\ sin(\theta) \end{pmatrix}$$

As has been done at the shrouded rotor, this point is used to place a circle with R_{Shaft} within the user interface. Every subsequent machine modeling step requires this center position since the entire rotor is shaped around it's center location. The state of the model can be seen in figure 5.9.

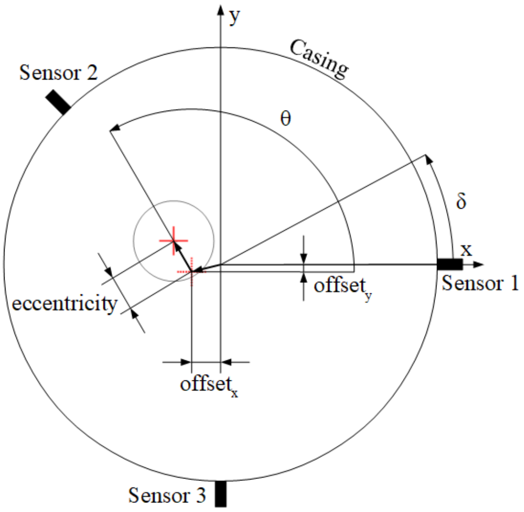


Figure 5.9: Model after shaft is localized

Blade Roots: Since the simple blade signal generation requires the interaction of individual blades with the sensing area, this part of the routine makes use of the shaft's center location and the geometry parameters for the purpose of finding the locations of the roots of the blades. Within a loop, the location of each blade root is determined and the blade number of the blade closest to a sensor - the active blades number - is evaluated. This serves the purpose of only having to compute the interaction of relevant blades with the sensors and thereby reducing computational load in the upcoming steps. The mentioned loop iterates for amount-of-blades i times, iterating the steps shown in figure 5.10.

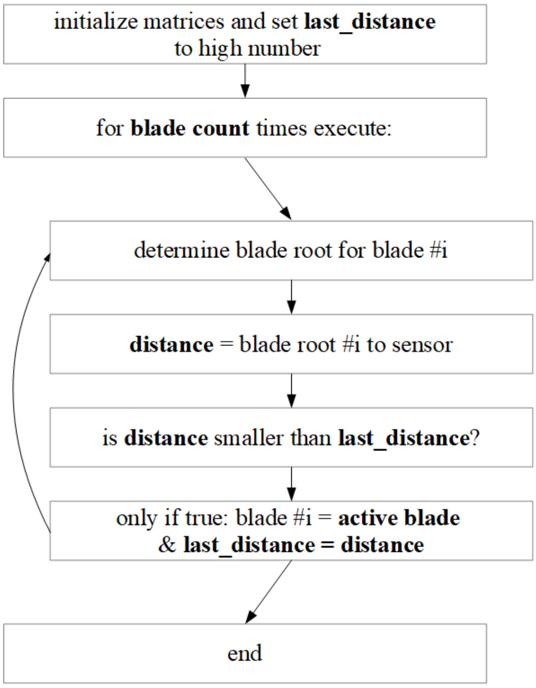

initialize matrices and set **last_distance** to high number
for **blade count** times execute:
determine blade root for blade #i
distance = blade root #i to sensor
is **distance** smaller than **last_distance**?
only if true: blade #i = **active blade** & **last_distance = distance**
end

Figure 5.10: Executed steps within the blade loop

According to the geometry shown in figure 5.11, a root point is determined by solving:

$$Root\ Point_i = \begin{pmatrix} Offset_x \\ Offset_y \end{pmatrix} + Eccentricity \begin{pmatrix} cos(\theta) \\ sin(\theta) \end{pmatrix} + R_{Shaft} \begin{pmatrix} cos(\theta + \alpha \cdot i) \\ sin(\theta + \alpha \cdot i) \end{pmatrix} \tag{5.8}$$

Subsequently, the distance from this root point $[x_i, y_i]$ to each sensor's tip is determined and compared to the distances already calculated. If the distance is smaller than any stored minimum value, it is stored as the new value together with the corresponding blade number i for the sensor. Like this, the 'active' blades that are the closest to each sensor are returned at the end of the iteration over all blade angles α_i. The model state after this step can be taken from figure 5.11.

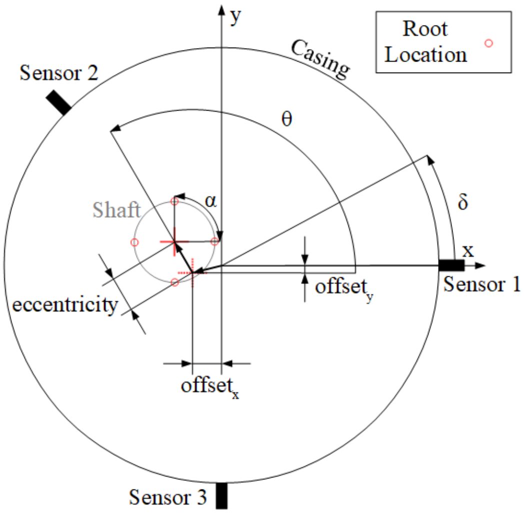


Figure 5.11: Model after blade roots are found

Blade Shaping: Now the position of the shaft has been defined and all blade root points have been computed. The next step is to attach blades to the shaft. In case of the simple geometry, the blades are modeled as rectangle with certain width W_B and length L_B. The routine calculates the four corner points of every blade dependent on the incremental parameter θ and the individual blade angles α_i, routing them to the root points and ultimately storing the data. The result is visible in figure 5.12.

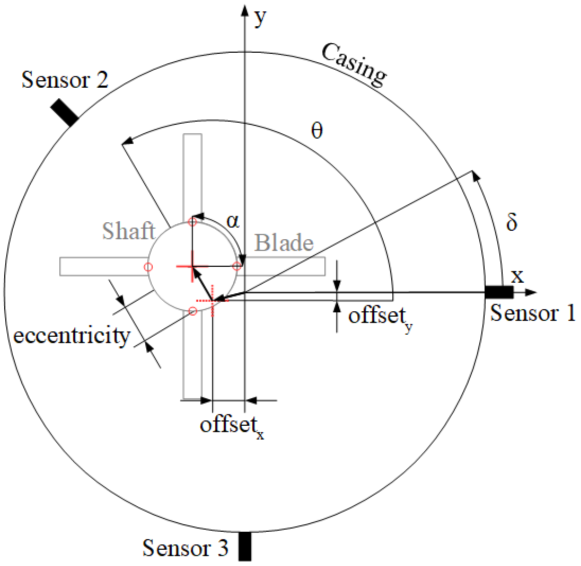


Figure 5.12: Model after blades are attached to the roots

Plotting: Ultimately, the final state of the machine model is displayed in Matlab's user interface since the machine model is evaluated. There, basic information is provided for the user as can be seen in figure 5.13. At the left, the entire rig is displayed two-dimensionally and at the right, a zoom on the sensor area is visible.

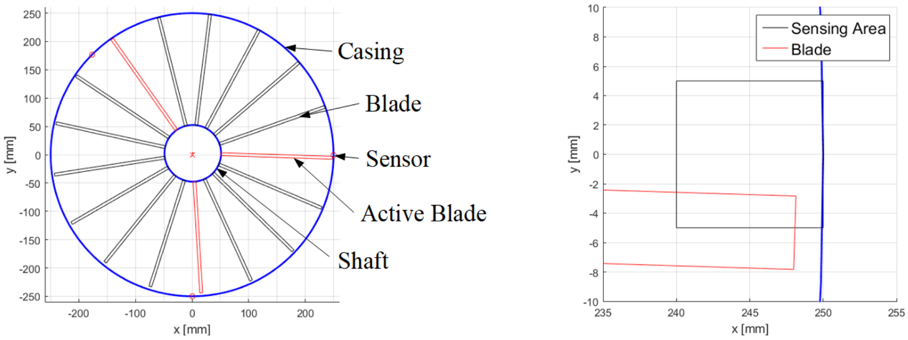


Figure 5.13: Matlab interface for a simple blades

Signal Loop: Here the true difference in simulating the signal starts. Instead of directly calculating the distance between the sensor tip and the rotor like at the shroud routine, a sensing area is defined and intersections with the active blades are detected. This is done within the signal loop that iterates over the amount of sensors, evaluating the signal for each one of them through the steps shown in figure 5.14.

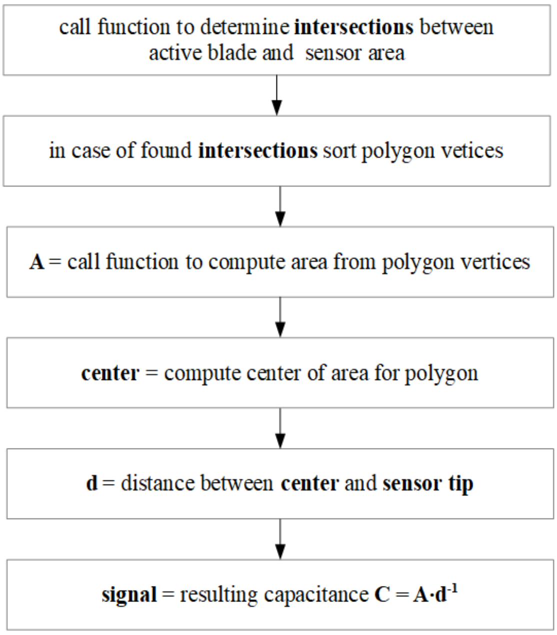

call function to determine **intersections** between active blade and sensor area
in case of found **intersections** sort polygon vetices
A = call function to compute area from polygon vertices
center = compute center of area for polygon
d = distance between **center** and **sensor tip**
signal = resulting capacitance **C** = **A**·**d**-1

Figure 5.14: Executed steps for each sensor to determine signal

First, the active blade and sensor area are checked for intersections using an algorithm by Schwarz. The entire function that has been implemented within Matlab can be found in the appendix of this thesis. In case of the blade and sensing area overlapping, this function returns a set of intersection points, that is subsequently passed to another function sorting the intersection location to be form a counterclockwise closed curve. This polygon is then passed back in form of an array, containing all vertices. The idea is made apparent in figure 5.15.

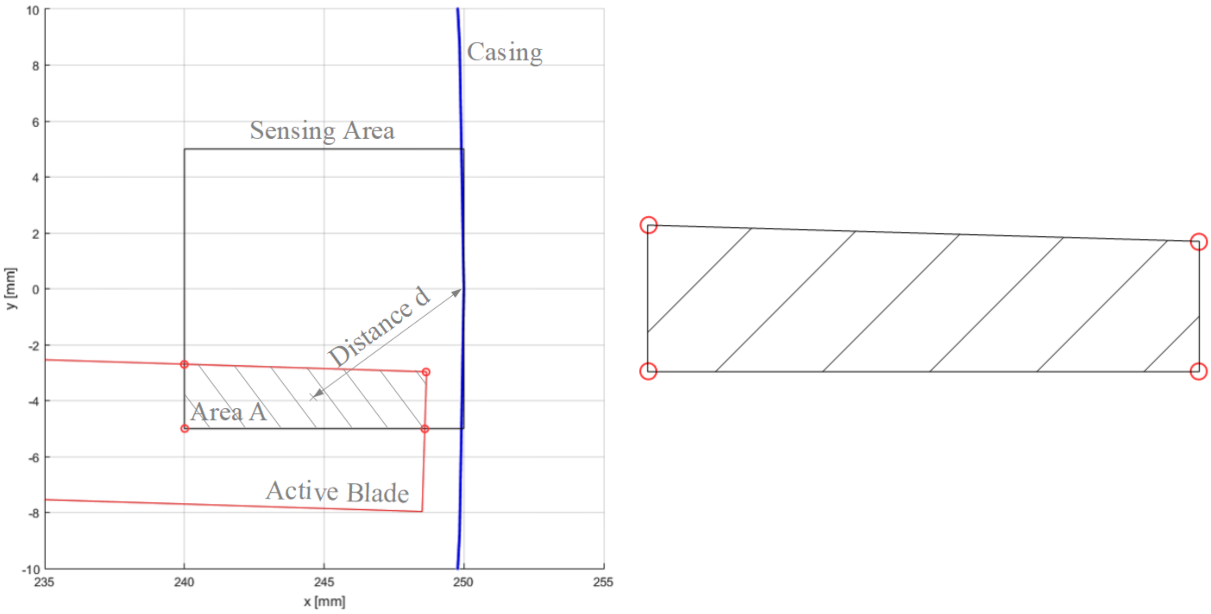


Figure 5.15: Interaction of sensing area and simple blade

The benefit of having a curve that is oriented counterclockwise has been mentioned in section 4.1 in connection with Green's theorem. The derived equation 4.20 for calculating the area in terms of a line integral

$$\oint_C \frac{1}{2}(xdy - ydx) = A$$

can be applied. Instead of a continuous curve, a polygon poses an even simpler form to compute this equation, since there is no need to solve the line integral. In this special case for discrete points connected by straight lines, the line integral can be substituted by a sum and the expression simplifies to

$$2A = \sum_{i=1}^{N}(x\triangle y - y\triangle x) \tag{5.9}$$

For a finite set of counterclockwise oriented points, the result is a positive value for the area A. In case of clockwise orientation, the result would be the negative value of A. The computation is performed by implementing the following Matlab function:

```
function [area] = area_algorithm(x,y)
%% (x,y) are arrays containing the vertices
%% area is a scalar containing the value of
%% the enclosed area of the polygon given by
%% (x,y)

  % amount of vertices
  N = length(x);
  % move the polygon to x > 0
  x = x + abs(min(x)) + 1;
  % initialize area parameter
  double_area = 0;
  % main loop
  for i=1:N
    j = i + 1;
        if (j > N)
          j = 1;
        end
        double_area = double_area + x(i) * y(j);
        double_area = double_area - y(i) * x(j);
  end
        % return the area
        area = abs(double_area)/2;
end
```

Furthermore, the vertices are used to compute the center of the enclosed area by making use of the geometrical center of said polygon with N vertices according to

$$\begin{pmatrix} x_C \\ y_C \end{pmatrix} = \begin{pmatrix} \frac{1}{6A}\sum_{i=1}^{N}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \\ \frac{1}{6A}\sum_{i=1}^{N}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \end{pmatrix} \tag{5.10}$$

The distance between $[x_C, y_C]$ and the sensor tip equals d, which concludes all the necessary information to calculate a resulting signal according to equation 5.3

$$C = Signal = \frac{A}{d}$$

The combined signals of all sensors are transformed into a string and sent to LabView for processing and the main loop restarts.

5.2.2.3 Output Signal

As has been the case at the shrouded rotor signal generation routine, the main loop iterates until a final angle θ is reached. The sum of the individual signals for every incremental step $\triangle\theta$ can be seen in figure 5.16. The signal has been created simulating a rotor with 17 blades at an offset of [1,1] mm and a constant orbit eccentricity of 1.5 mm. The three sensors were placed as figure 5.8 suggested: at 0, 135 and 270 deg. Note, that the main differences to the constant shroud clearance signal are the amplitude that is now a capacitance value in mF and the pulsing character of the signals. The peak amplitudes translate to the maximum capacitance that is caused by the blade being the closest to the sensor. At moments of rising and falling capacitance the blades are either entering or leaving the sensing area. As will be proven later, only the peak amplitude can be translated to a true clearance between sensor and blade.

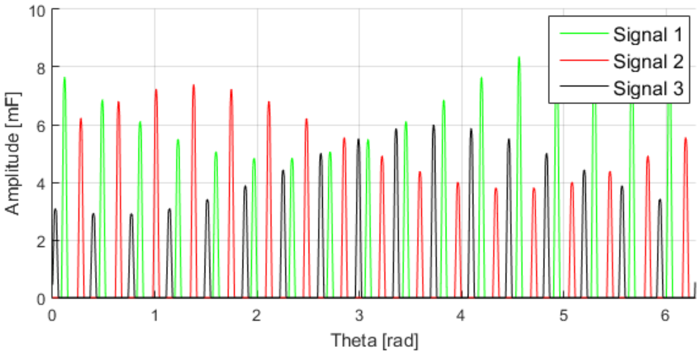


Figure 5.16: Simulation results of the simple blade routine (offset = [1,1] mm & eccentricity 1.5 mm)

5.2.3 Squealer Blades

The last step in complexity added is the simulation of blades that divert from the just discussed rectangular shape. As portrayed in 2.3, blades with squealer tips are used to limit tip leakage flow. In this thesis a squealer blade is modeled by attaching two additional tips to the tip of the simple blades from before. The location and amount of these additional tips can vary in reality, therefore the procedure for two tips can be seen as general approach that can be extended. The result is visible in the output, by having an extra, local peak on top of the original blade peak for each additional tip.

The initialization steps are exactly the same as for the simple blade routine, with the only difference being the addition of geometric parameters describing the squealer tips. These parameters are:

- Squealer Width
- Squealer Height

5.2.3.1 Main Loop

Blade Shaping: After determining the shaft's position, finding the blade's roots and modeling the basic blade rectangle like before, this routine is expanded to add the additional tips. This is done by attaching two rectangles to the existing main blade. The result can be seen in figure 5.17, where the user interface is presented.

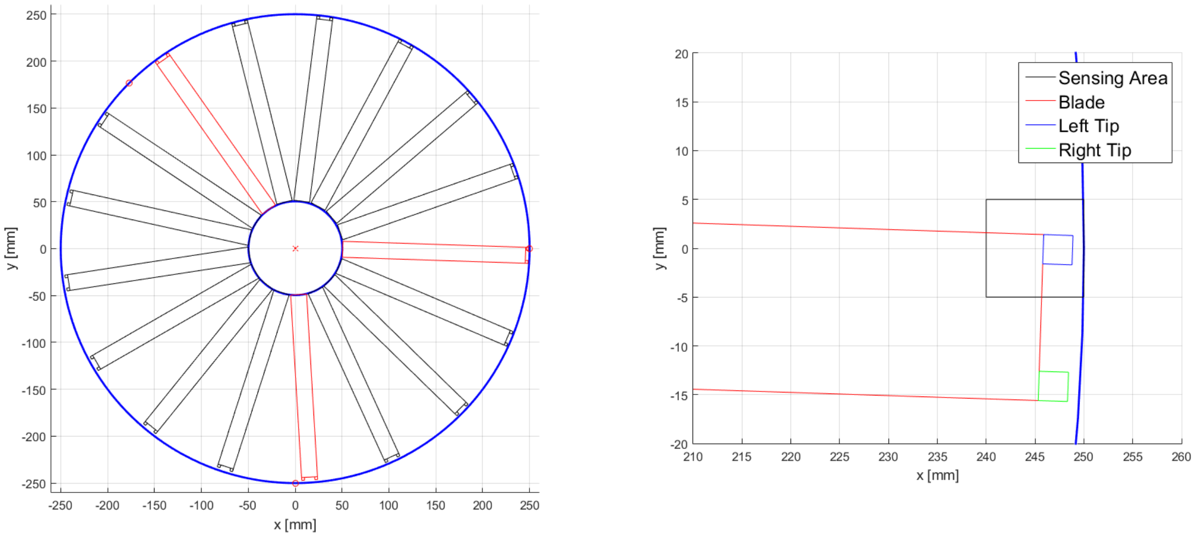


Figure 5.17: Matlab interface at the squealer blade routine

Signal Loop: The next alteration of the simple blade routine is the way the signal is computed. The three separate surfaces - blade, right and left squealer - are treated independently. For each surface the intersection with the sensing area, the resulting polygon, the area A_i and the distance from the geometric center to the sensor tip d_i is calculated, using the steps introduced at the simple blade routine. Subsequently, the three values A_i are added up to A_{total} and the center of this area is estimated by performing a weighted addition of d_i according to

$$d_{weighted} = \frac{\sum_{i=1}^{3} d_i A_i}{A_{total}} \tag{5.11}$$

The polygon describing the entire surface is presented for an exemplary setup in figure 5.18.

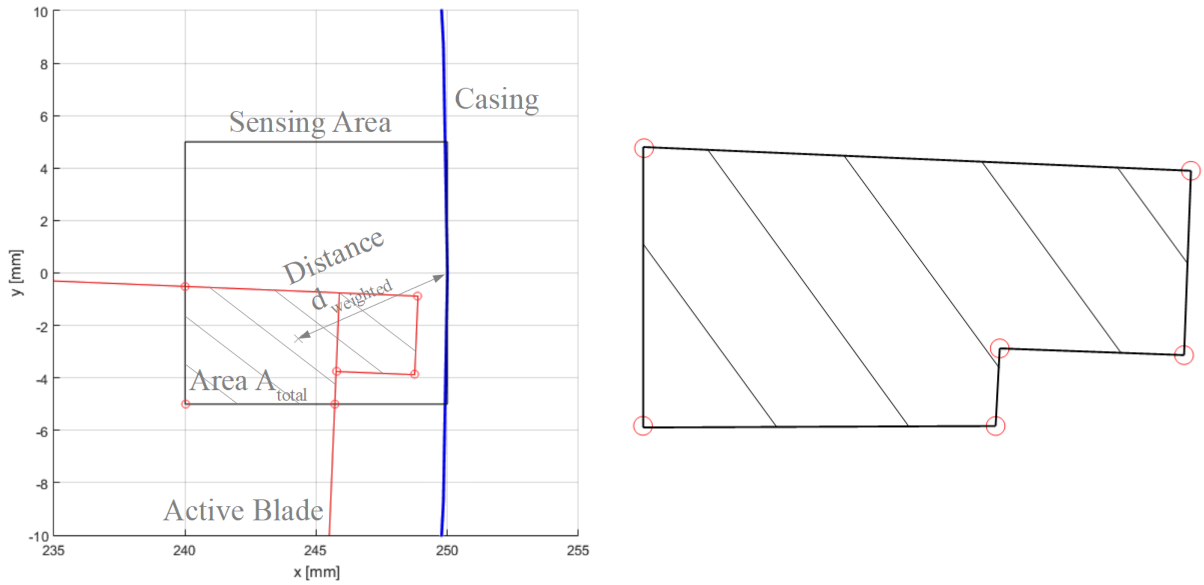


Figure 5.18: Interaction of sensing area and squealer blade (left) and resulting polygon (right)

The final output signal for this iterative step is obtained according to expression 5.3

$$C = Signal = \frac{A_{total}}{d_{weighted}}$$

and sent to the TCP server hosted by LabView.

5.2.3.2 Output Signal

Comparable to the simple blades, the distinguishing differences between the continuous shrouded rotor clearance signal are the amplitude in terms of capacitance and the pulsing structure. In addition to that, the two squealer tips cause two individual local maxima on each of the blade passing peaks. The signal for the simulation of half a revolution of a rotor with 17 blades with an offset of [0.5,0.5] mm and an eccentricity of 0.5 mm can be seen in figure 5.19. The sensors were positioned at 0, 135 and 270 deg. Equal to the simple blade signal, the clearance is not directly visible and has to be derived within LabView.

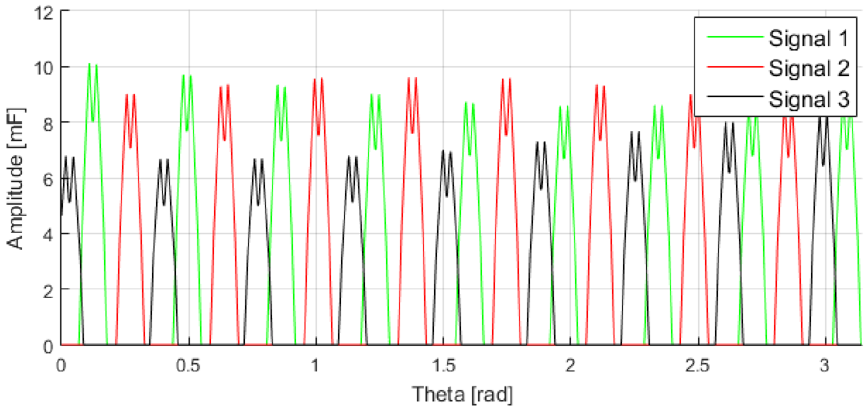
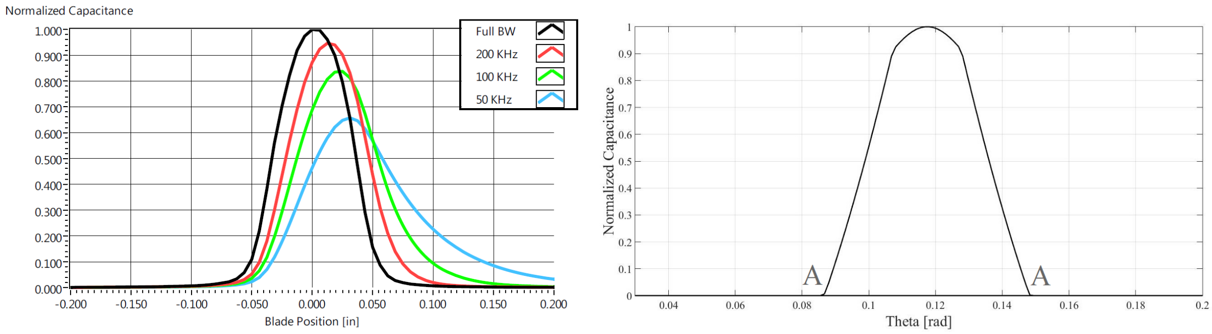


Figure 5.19: Simulation result for squealer blades (offset = [0.5,0.5] mm & eccentricity 0.5 mm)

5.3 Signal Verification

Before starting to apply the radial rotor-casing clearance signals for the development of the LabView routines, the resemblance of real world measurement data has to be proven. The goal is not to achieve a perfect replication of actual signals but rather to ensure that the clearance information is encoded within the signal in the same way. If this applies, the derived clearance monitoring system is guaranteed to be applicable to simulated and real world signals equally. Generally, three separate routines have been created using different machine model geometries and alternating approaches in creation of a output signal. Since the availability of real world measurement data is limited, the verification of the signal for shrouded blades is done by proving accuracy for simple and squealer blades.
In figure 5.20 the comparison of clearance data for a simple blade passing by a capacitive sensor is displayed. The left image (a) shows results of real-world measurement conducted by Haase and Haase (2013). The authors highlight the effect of limited bandwidth on capacitance peaks by low-pass filtering the ideal result shown in black. Since this effect is not modeled in the present thesis, only the mentioned black peak is used as a reference. Image (b) on the right displays a signal that has been simulated with the presented routine for simple blades. It is apparent that the main difference lies within the less progressive and more angular shape of the simulated signal at moments when the blade enters or leaves the sensing area (A).



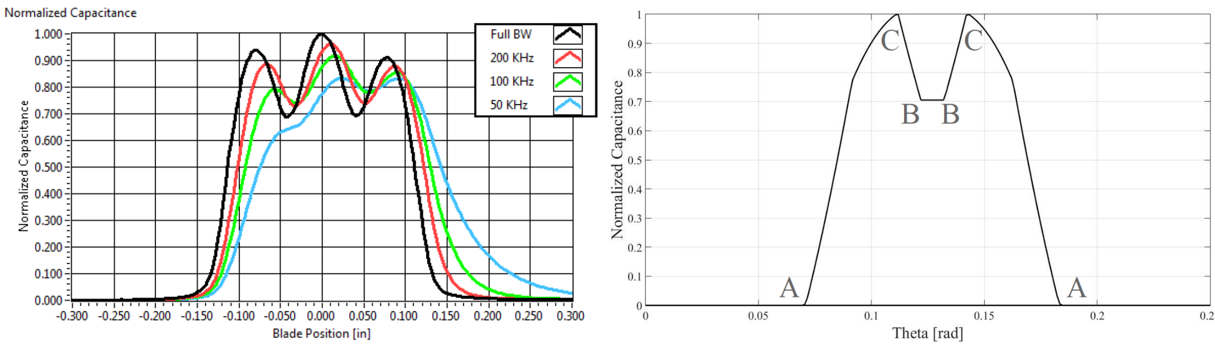
(a) Real signal by Haase and Haase (2013)

(b) Simulated signal

Figure 5.20: Validation of signals created by the simple blade routine

The reason for that lies within neglecting the non-linear range of the sensor by modeling its sensing area as a square, as has been presented in section 5.1.2. In real-world application the sensor detects the arriving blade earlier and more progressive while it is moving through the outer layers of the electrical field. Additionally, in accordance with section 5.1.1, the blades are modeled with perfect 90 deg corners. While this amplifies the angular character of the signal even more, the integrity of the information carried by the signal is not affected in both cases. This is due to the fact that the entire clearance information is contained in the peak's amplitude and not in its shape.

Similar to the simple blade geometry, the comparison of squealer blade routine results to real-world measurement can be seen in figure 5.21. On the left (a) Haase and Haase (2013) have analyzed the effect of limited bandwidth on measurement results for squealer blades by low-pass filtering them. Again, the only important signal for the underlying statement is shown in black for idealized system bandwidths. Note, that the author has used squealer blades with three tips on top of the original blade instead of two, as has been the case throughout this thesis. The comparison is still viable since the goal is to clarify if the simulated signals are containing the desired information. Image (b) shows the result of the simulation of a squealer blade. Equal to the simple blade results, the simulation proves to create more angular results. This time the effect is not limited to the base of the peak (A) like before, but reoccurs in times when a squealer passes the center of the sensor (C) and when it leaves the sensing area (B). Nevertheless, the explanation remains the same. The angularity is a result of neglecting the non-linear range of the sensor and the rectangular shape of the blades, leading to the displayed abrupt changes in capacitance. Apart from the differences in signal shape, the presence of two separate peaks suffices for the presented processing routines to be functional.



(a) Real signal by Haase and Haase (2013)

(b) Simulated signal with two tips

Figure 5.21: Validation of signals created by the squealer blade routine

Overall, the deflections of simulated signals from real data are deemed acceptable to justify the design of a processing routine based on them. In combination with individual calibration for every type of simulation, the signal can be expected to contain the clearance information desired.

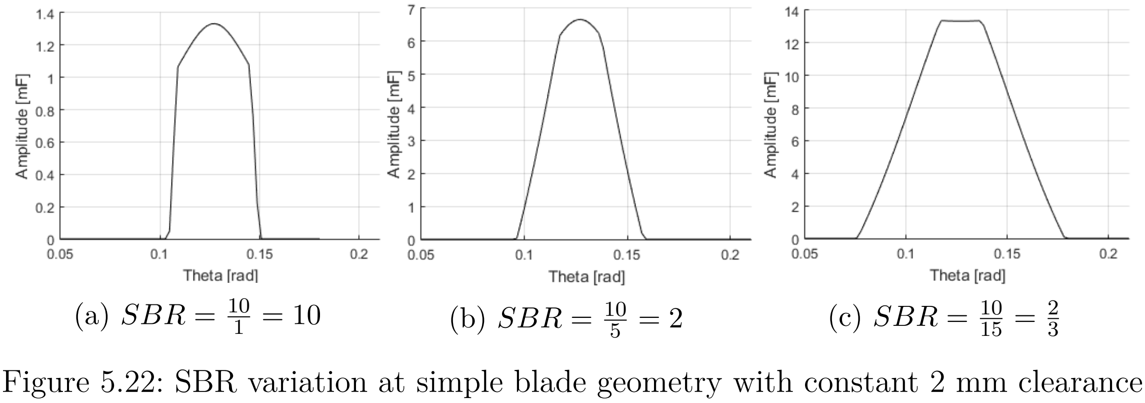
5.4 Parameter Limits

Throughout the development process the combination of two parameters has proven to be able to cause problems, due to the impact of altering signal structure in a way that the peaks do not resemble their real-world equivalent. Therefore, a dedicated discussion of the interaction between 'Sensor Width', describing the square representing the sensing area, and 'Blade Width' is at hand. In the beginning of a simulation process a certain Sensor-to-Blade-Width-Ratio (SBR) is specified by the user:

$$SBR = \frac{Sensor\ Width}{Blade\ Width} \tag{5.12}$$

If certain limitations of SBR are exceeded, the resulting signal shape might lead to peak detection problems.

Simple Blade: Demanding the capacitance peaks to look like shown in section 5.3, the shape can be rather different for certain SBR compositions. Figure 5.22 shows the results for the variation from narrow blade widths (a) to wider blade widths (c) - always at constant sensor width. While high SBRs (a) yield signal forms diverting from the ideal clearance peaks, the information is still contained. Whereas at low SBRs (c), the danger of having a constant amplitude or two detectable peaks might compromise the signal processing results. At a SBR of two, shown in image (b), the best results are obtained. Generally, the lower limit of SBR = 1 for successful signal simulation can be stated. Starting at this ratio, where the blade and the sensing area are of equal width, the resulting peaks show periods of near constant amplitude. Note the differences in maximum amplitude in all three cases, leading to the necessity of calibration for every different SBR simulated.



(a) $SBR = \frac{10}{1} = 10$ (b) $SBR = \frac{10}{5} = 2$ (c) $SBR = \frac{10}{15} = \frac{2}{3}$

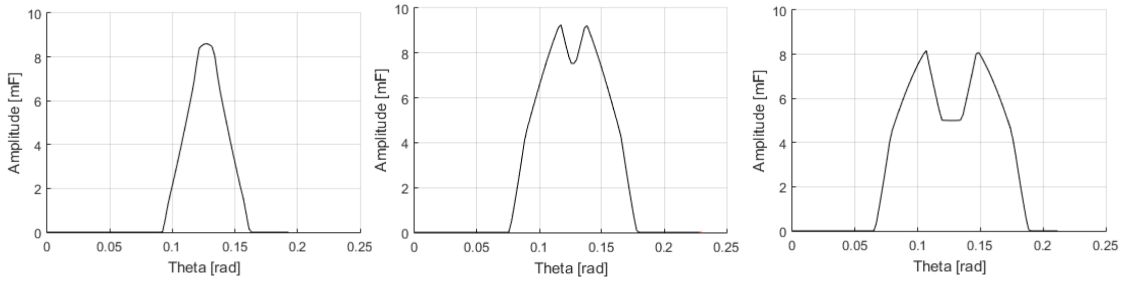
Figure 5.22: SBR variation at simple blade geometry with constant 2 mm clearance

Squealer Blade: While analyzing simple blade geometry, only two parameters were deemed important for the simulation of correct signals. Now, at blades with additional detectable tips on top of the actual blade, two more factors describing the geometry have to be accounted for: the width and height of the additional edges. In figure 5.23 results for the parameter variation of

- Sensor Width
- Blade Width
- Tip Width
- Tip Height

is presented. The parameters were always set in order for the main body of the blade and the two tips to interact with the sensing area at some point. First, in figure 5.23 (a), a high SBR with a tip width of 1 mm and a tip height of 2 mm is visible. The underlying method does not manage to pick up on the two additional tips, since at the point of maximal amplitude, the sensing area covers the blade and the tips, resulting in a single capacitance peak being visible.

Lowering the SBR and increasing tip width to 3 mm and tip height to 4 mm, to fit the increased blade width, results in the peak shown in figure 5.23 (b). As soon as the sensing area square is too small to pick up on both tips entirely, two separate peaks become visible. Alternating the parameters further to an even lower SBR and a tip height of 5 mm at unchanged tip width in figure 5.23 (c), the separate peaks in amplitude become more distinguishable. This behavior leads to the conclusion that the sensing area has to be small enough, so that the two tips are never fully covered at the same time by the area, or only one peak is detected. As mentioned at the simple blade before, the difference in maximum amplitude for the three presented parameter variations requires for calibration for every parameter setup.



(a) $SBR = \frac{10}{7}$ (b) $SBR = \frac{10}{15}$ (c) $SBR = \frac{10}{20}$

Figure 5.23: SBR variation at squealer blade geometry with constant 1 mm clearance

6 LabView Signal Processing

Chapter 5 concluded with the introduction of three different routines that simulate different machine and measurement system models. The output signal of each routine is continuously sent to a TCP Server, hosted by the monitoring routine, where they are processed and vital information is derived. Due to its strengths in signal based computing and wide spread usage in technical applications the software package LabView is used for that purpose. Caused by the fundamental difference in signal structure as a result of different rotor geometries - continuous or pulsing with possible variations - the development of three individual routines is conducted:

- Shrouded rotor signal processing unit
- Simple blade signal processing unit
- Squealer blade signal processing unit

The most apparent information contained in each of the provided signals is the radial rotor-casing clearance underneath a sensor. This data shall be derived from the signal and subsequently be processed to gain additional information about the machine. Apart from the reconstruction of the orbit and stationary offset that were specified, the goal is to determine the smallest existing rotor-casing clearance at any time. To comprehend the functionality and dependencies this chapter starts with outlining the most important processing tools used within the individual routines. They are treated in order of appearance and shall prepare the reader for the step by step discussion of the individual routine steps afterwards.

6.1 Processing Tools

The purpose of this section serves the introduction and explanation of key processing steps used within all routines. Starting at the freshly simulated signal, the first step in any routine can be the filtering of the signal to enhance its quality. This is followed by reducing the signal to the key information, the clearance underneath the sensor. In case of the simulation of the shrouded rotor the signal already directly correlates to the distance between sensor tip and shroud. This statement does not apply to the signals of the bladed configurations. The only part of the signal that correlates to a clearance minimum between sensor and blade are the peaks. Therefore, the applied peak detection algorithms are showcased in combination with a verification method involving a so called 'key phasor'. After obtaining the desired data from the signals, the following steps include the approximation of the shaft's location and extrapolating this data to gain an idea about the smallest available radial rotor-casing clearance margin.

- Low Pass Filtering
- Peak Detection
- Usage of a Key Phasor
- Translating Signal to Distance
- Calculation of the Shaft's Location
- Detection of the smallest Rotor Clearance

6.1.1 Low Pass Filtering

While discussing the entirety of the measurement chain in chapter 3, many secondary signal components have been mentioned and even specifically treated in section 3.2.3, defining the expressions noise and Signal-to-Noise-Ratio (SNR). Together with the introduction of low-pass filtering in section 4.2, this leads up to the usage of said filter type for signal enhancement by using the filter's most important attribute - the dampening effect it has on selective frequency components. It has been shown that at the cut-off frequency f_c the loss in magnitude of $\approx$ 3 dB comes along with a phase shift of -45 deg. Frequency components higher than f_c are affected more than those lower than f_c. This can be used to effectively reduce noise, since in the present case the blade peaks occur with a significantly lower frequency than some noise components. Note, that not every secondary component in a signal can be removed with this process.
Unfortunately, the application of a low-pass filter to increase signal quality comes at a price. If the cut-off frequency of the filter is set to the blade-passing frequency, the loss in magnitude of 3 dB occurs. Translating this value from the frequency domain with

$$-3[dB] = |G(j\omega_c)|_{dB} = 20log(|G(j\omega_c)|) \tag{6.1}$$

the resulting gain of

$$|G(j\omega_c)| = 10^{(-\frac{3}{20})} = 0.7071 \tag{6.2}$$

causes a reduction in amplitude of around 30% on the signal components with that exact frequency. Since the amplitude of the signal refers directly to the clearance, an error of this size renders the data useless. The resulting strategy leads to increasing the cut-off frequency, so that mainly noise is affected by the filter. Due to the fact that the desired blade-passing signal itself is composed out of multiple frequency components - some being much higher than the blade-passing frequency itself - a loss in magnitude has to be expected even then. The implemented filtering sub-routine that is used throughout the processing is displayed in figure 6.1. After inputting a data array, the first step is to address filtering artifacts. These are errors occurring at digital filters in the beginning and the end of the data set. They can be removed by taking the first and the last value of the input array, creating two new arrays containing one time only first array elements and one time only last array elements. The length of these new arrays is set to be the same length as the input array, which usually suffices, unless the input array contains very few elements. Now, the array containing first elements is attached to the beginning, and the other one to the end of the input array. This leads to the suppression of said unwanted artifacts. After that, a process called forward-backward filtering is applied. The transformed array is inverted, filtered and then inverted and filtered again. This leads to one time shifting the phase in one direction and directly afterwards shifting it back the same amount, while the amplitude is affected only in the first filtering step. The resulting array is clipped to contain only the central part with the initial length and handed to the output.

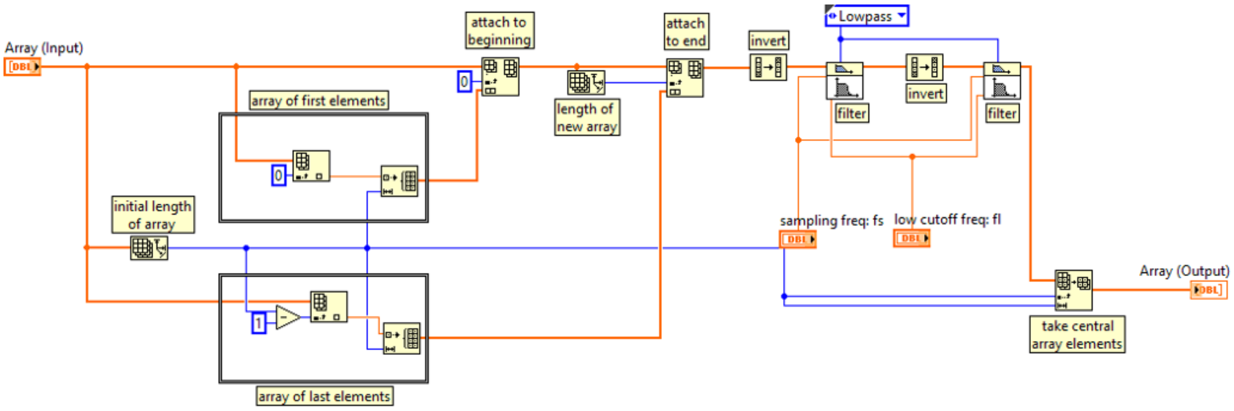


Figure 6.1: Forward-Backward low-pass filtering routine in LabView

6.1.2 Peak Detection

As a result of the measurement system model simulated in Matlab, the blade's interaction with the sensing area that has been computed using equation 5.3

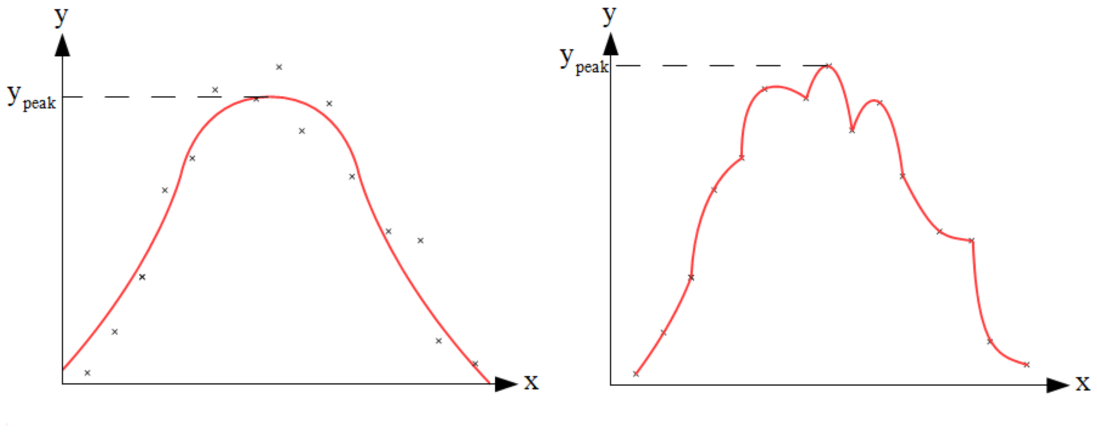
$$C = Signal = \frac{A}{d}$$

show following tendencies:

- As soon as the blade enters the sensing area, the result for the overlapping area A increases until it reaches a maximum, when the blade is directly in front of the sensor. Subsequently, the value for A falls back to zero when the blade moves away from the senors tip and leaves the sensing area.
- The distance d behaves in the exact opposite way. Initially, when the blade enters the sensing area, the value for d is a maximum and falls off towards the sensor tip. After reaching a minimum when the blade is right underneath the sensor tip, the value increases again until no interaction between blade and sensing area is recorded.

These tendencies allow for the statement that the resulting signal reaches the maximum, when the blade is the closest. This peak in the signal amplitude is therefore representative of the clearance between the blade and the sensor's tip. Note, that only the peak links to a clearance, since A and d increase and decrease due to the blades lateral movement and not just based on radial displacement.

In section 4.3 a method for detecting these peaks has been introduced. After fitting a curve into the available data and differentiating the resulting function, information about peaks and valleys can be obtained. Something that has not been specifically mentioned though, was the smoothing effect of curve fitting. Dependent on the amount of data points N that have been set to be used for the calculation of the curve, and the order of the polynomial, a significant amount of noise is evened out, as is displayed in figure 6.2 (a). The problem is a loss of accuracy. If the amount of data for a curve fit is set too high and the order of the polynomial is set too low, peaks might be underestimated and valleys overestimated. This means the resulting polynomial predicts the peaks less and less accurate, the higher N is. The software package LabView provides a sub-routine dealing with peak detection. The source code for this routine is encrypted and can therefore not be analyzed directly. Nevertheless, LabView's documentation at the National Instruments Website states the usage of second order polynomial curve fitting using the 'least squares' method that has been presented in the mathematical background in section 4. The routine provides an input called 'width', resembling the amount of data points N the routine uses at once. The lower limit to be able to fit a second order polynomial $y = a_2x^2 + a_1x + a_0$ is three. When setting up this value, the resulting polynomial contains all of the provided data points, achieving the highest accuracy. By successively fitting second order polynomials into sets of 'width' amount of points and then continuing to the next set, the entire data is reduced to $\frac{Amount\ of\ Samples}{width}$ individual polynomials that can be checked for peaks and valleys by differentiation. Figure 6.2 (b) shows the improvement in resemblance with the actual data of stepwise approximating the curve, compared to using a single fitting step, as has been the case in figure 6.2 (a).



(a) Curve fitting using a single fitting step (b) Curve fitting using sets of three points

Figure 6.2: Differences in accuracy for fitting second order polynomials

When thinking about the data from figure 6.2 as a peak in capacitance due to a blade passing by and the local fluctuations as noise affecting the clean signal, the difference in detected peak amplitude y_{peak} between (a) and (b) becomes apparent. Note, that in noise-heavy signals, low 'width' can lead to detection of local amplitude spikes caused by the noise itself, what poses a threat to the integrity of the routine. At a high 'width' on the other hand, the maximum peak value might be underestimated. The answer to the question of which one is more accurate is very situational and shall not be evaluated for the given example, since only the effect of smoothing should be pointed out at this moment.

Problem: While the peak detection algorithm integrated into LabView performs well at detecting peaks inside the array, the underlying mathematical models fail at the edges of the array. To explain the arising problems a thought experiment can be conducted: Imagine having two consecutive clearance data arrays. The first one shows rising capacitance due to an approaching blade and the second one contains decreasing capacitance values as a result of the same blade moving away from the sensing area. The idea is made apparent in figure 6.3. The maximum value lies either at the right end of the first array, or at the left end of the second array - the statements made apply in either case. Note, that left indicates a lower index within an array and thus an earlier time of sample recording.

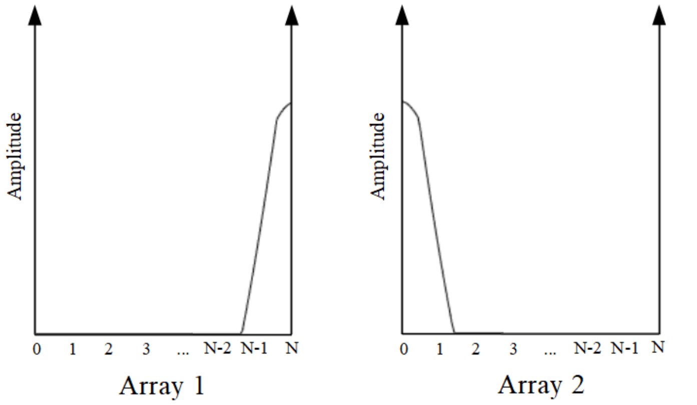


Figure 6.3: Two consecutive signal data arrays with a peak at the cut

Handing those two arrays to the peak detection sub-routine yields no peak information. This is due to the fact that the sub-routine starts at the left side of the first array taking 'width' amount of samples, fitting the polynomial, differentiating the resulting equation and proceeding to the next 'width' amount of samples until it reaches the right end. Since the amplitude A was said to be rising steadily with increasing index i, the statement $A_i < A_{i+1}$ can be made. This leads to the fact that the last fitted polynomial p(x) is steadily rising and thus does not have a maximum according to $\frac{dp(x)}{dx} = 0$. As soon as the next sample set - the second array mentioned previously - is buffered and sent to the processing unit, the peak detection sub-routine starts at the left and finds amplitudes according to $A_i > A_{i+1}$, resulting in an polynomial that steadily decreases and neither yields any extremum.

The problem just described concerning peaks at the edges of arrays only gets worse when increasing the 'width' setting of the detection sub-routine. For a peak with linear slops, the 'width' setting, for which a peak could not be detected in relation to the offset of the edge of the array with N elements, is shown in table 6.1

Table 6.1: Detection limits regarding the 'width' setting

'width'	index of the peak, so that no peak was detected
5	N-1
6	N-2
8	N-3
10	N-4
12	N-5

Generally the statement

> A peak at index N-i of an array with N elements can only be detected, if $\frac{width}{2} \leq i$.

describes the boundaries for the functionality of the sub-routine for a signal with linear slops at a gradient of ±1. Note, that this statement is based on empirical data, since the source code for the implemented routines could not be accessed. Changing the slops leads to different results, since the fitted second-order polynomial might be shaped differently. The tendency of missing peaks at the cut remains nonetheless.

Solution: The problems arising are rather rooted in array-based data processing than in the detection routine. Nevertheless, since the integrity of the data is based on stable detection of every blade, this problem has to be addressed. To avoid referencing mistakes by missing peaks, the routine is expanded to repeat the following steps within the data processing loop (B) (compare figure 6.4):

1. Receive sensor signal in form of an array.
2. Label this array 'Array 1'.
3. Run the peak detection routine with 'Array 1' as input.
4. Pass peak information on to referencing.
5. Store 'Array 1' for next loop iteration.
6. Receive sensor signal in form of an array.
7. Label this array 'Array 2'.
8. Take X amount of samples from the end of 'Array 1' and from the beginning of 'Array 2' respectively.
9. Merge the samples together, forming a new array.
10. Label this array '2^{nd} instance'.
11. Run the peak detection routine with '2^{nd} instance' as input.
12. Pass peak information on to referencing.
13. Run the peak detection routine with 'Array 2' as input.
14. Pass peak information on to referencing.

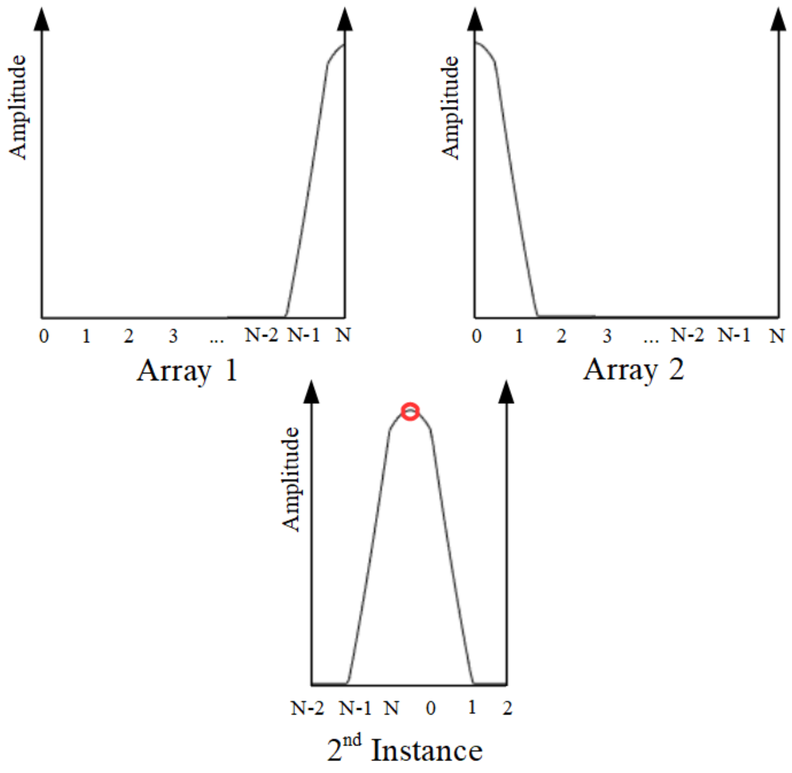


Figure 6.4: Solution for problem at array-based peak detection

This procedure allows for the detection of peaks that were initially at the edge of an array, by moving them to the middle of a new array. This second instance is accompanied by intermediate steps that serve the avoidance of double detection of peaks by cross-referencing indices of the peaks found throughout the entire process. The method ensures loss free peak detection with LabView's integrated peak detection sub-routine, which has already been mentioned to be vital for the integrity of the entire processing routine.

6.1.3 Key Phasor

A problem that arises during measurement conducted at rotors with either none (shroud) or many (bladed) distinctly shaped rotor components is that referencing can be difficult. Since the measurement system samples data only at discrete points in time and angular velocities of the machine are usually transient to some extent, an external signal has to be used for tracking the angular position of the rotor - a 'key phasor'. This sensor's sole purpose is creating a pulse once per full revolution of the rotor.

In combination with predefined 'First Blade after the Sensor' information, the signal of the phasor can be used to confirm that each clearance signal is assigned to the corresponding blade. This is done by comparing the time stamp of the last peak before the key phasor signal (peak 1) and the peak after the key phasor signal (peak 2) to the key phasor signal itself. The key phasor signal has to arrive between the two consecutive peaks and the blade number of peak 2 has to be equal to the 'First Blade Number for Sensor i'. Note, that instead of using a time stamp a global x-scale like in figure 6.5 can be established, fulfilling the same purpose.

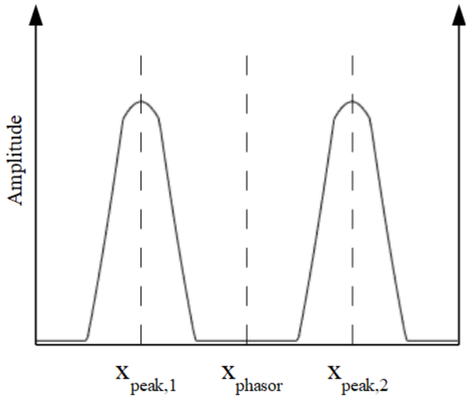


Figure 6.5: Concept of data validation using a phasor signal

6.1.4 Signal to Clearance

In case of the bladed rotor signals, the information is provided in terms of capacitance over time. The simple and squealer blade routines calculate the signal according to equation 5.3:

$$C = Signal = \frac{A}{d}$$

At this point it has to be emphasized that the radial rotor-casing clearance is not equal to the value d. This distance was a byproduct of the capacitive sensing model. Figure 6.6 shows the circumstance.

Therefore, the signal is translated to the clearance using a calibration curve. The data for this curve is recorded by setting up a centered shaft (no eccentricity nor offset) with constant blade length. Due to the centering, the clearance is equal at the entire circumference. This setup is simulated and the maximum signal amplitude is obtained with LabView.

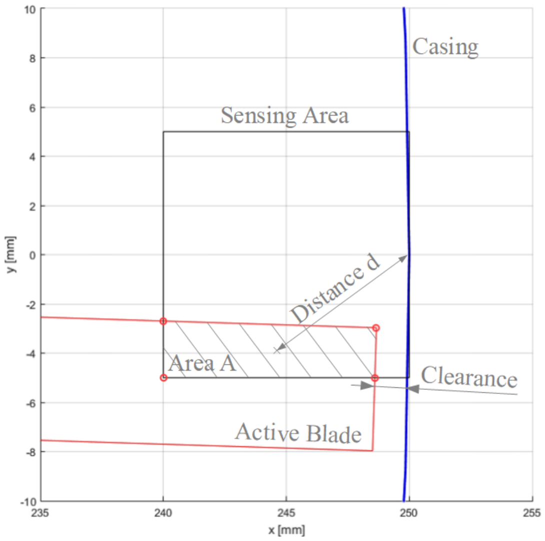


Figure 6.6: Distinction between clearance and distance d

Since the machine geometry is known, this amplitude refers to the clearance obtained by:

$$Clearance = Radius\ Casing - (Radius\ Shaft + Blade\ Length) \tag{6.3}$$

By repeating this process for clearances ranging from zero to five millimeters in smalls steps, a curve connecting signal and clearance is acquired. In figure 6.7 the discrete (signal,clearance) inputs can be seen in terms of a cross. Through linear interpolation every signal can thereby be linked to a clearance.

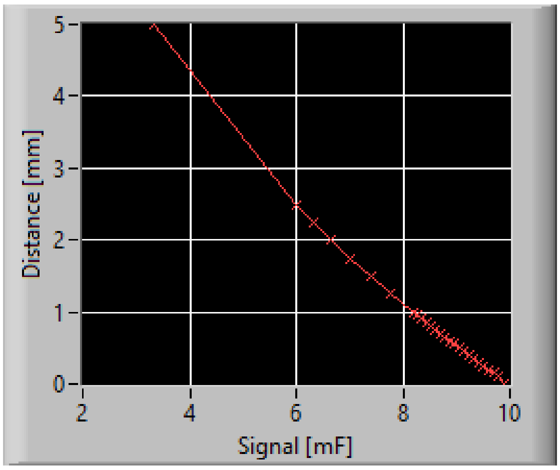


Figure 6.7: Calibration curve within LabView

6.1.5 Shaft Location

Note, that up to this point no specifications concerning the amount of sensors used for measurement have been made. Generally, a single sensor is sufficient to acquire clearance data between the sensors tip and the rotor component. If two additional sensors are mounted on the casing - making a total of three - the mathematical principle concerning the unambiguous definition of a circle by the means of three points on its circumference that has been presented in section 4.4, can be applied. In order to do so, the definition of the position of each of the sensors on the casing's circumference has to be specified:

- Sensor 1 Position δ_1
- Sensor 2 Position δ_2
- Sensor 3 Position δ_3

By knowing the sensor positions and the radius of the casing, the detected clearances can be assumed to be points of a circle that is inscribed into the shroud or tips of the blades respectively. A sub-routine allows for the computation of the center of said circle and its radius according to equations 4.63 to 4.65

$$x_C = -\frac{B}{2A}; \quad y_C = -\frac{C}{2A}; \quad r = \sqrt{\frac{B^2 + C^2 - 4AD}{4A^2}}$$

The three points in two-dimensional space are acquired by solving the expression

$$\begin{pmatrix} x_i \\ \\ y_i \end{pmatrix} = (Radius\ Casing - Clearance_i) \begin{pmatrix} cos(\delta_i) \\ \\ sin(\delta_i) \end{pmatrix} \tag{6.4}$$

for each individual sensor. The idea is made apparent in figure figure 6.8. Assuming that the three clearances used to define the circle are all recorded at the same time and that the machine components all are perfectly circular, the resulting center of the circle (x_C,y_C) is the momentary position of the shaft. The shaft's movement can be tracked by computing this position repeatedly and using the resulting path to assess the machine's operational condition. Note, that this procedure implies that a perfect circle can be inscribed into the shroud tip or the blade tips respectively. The error this leads to in case of length variations of the blades or imperfect shroud circularity will be analyzed later on.

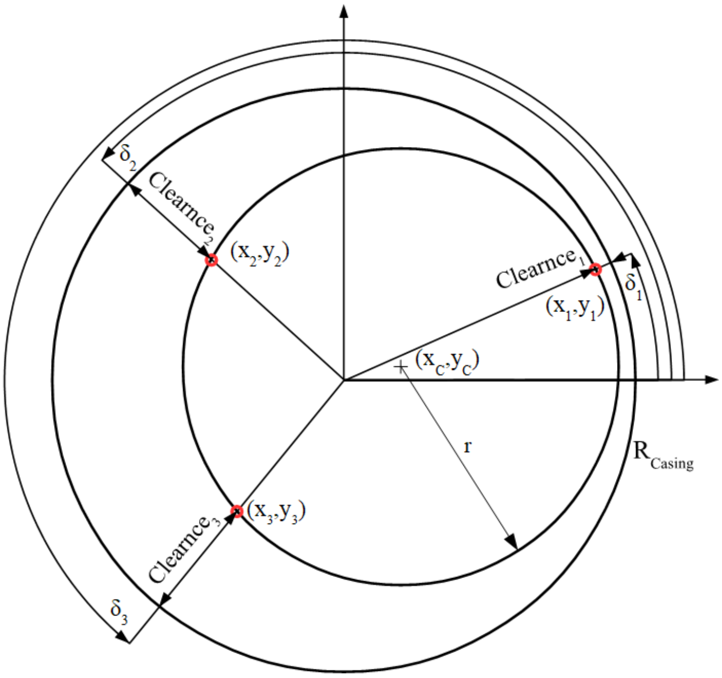


Figure 6.8: Circle inscribed into the blade tips/shroud

6.1.6 Rotor Clearance

As a result of the successful determination of the shaft's position using the clearance data underneath the sensors, the smallest clearance between the rotor and the casing can be approximated. Note, that in figure 6.8 this value coincidently equaled the clearance signal of the sensor at position δ_1. In a more general location, the clearances between sensor tip and rotor might not show this minimum. This case is displayed in figure 6.9. The location of the smallest clearance is found by drawing a line from the center of the casing (0,0) to the casing, while passing the center of the shaft (x_C,y_C). Therefore, this value can be calculated using

$$Min.\, Clearance = R_{Casing} - (\sqrt{x_C^2 + y_C^2} + r) \tag{6.5}$$

Both shaft center (x_C,y_C) and the radius of the circle inscribed into the tip r are a result of the unambiguous definition of circle using the three sensors.

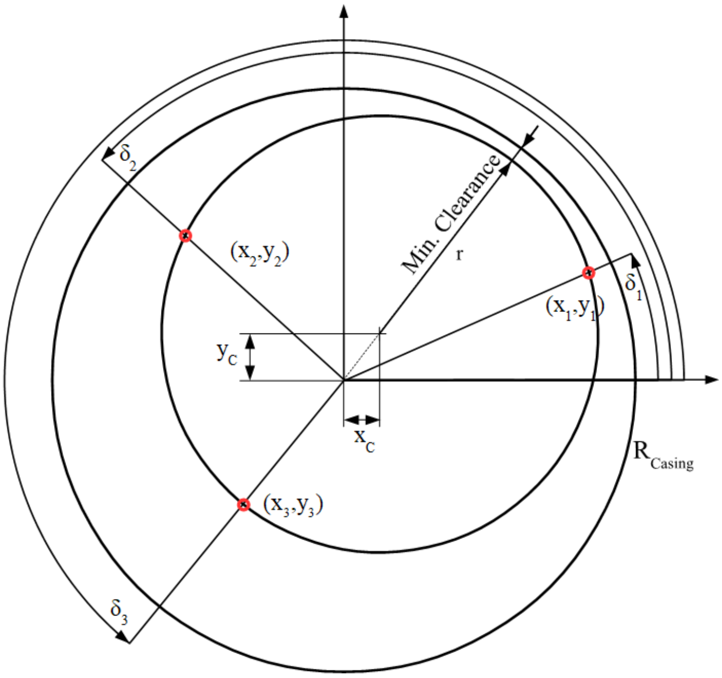


Figure 6.9: Smallest radial rotor-casing clearance

6.2 Routines

After explaining the essential tools used for data processing, the structures implemented into LabView are now introduced. First, the routine for shrouded blades shall be discussed, focusing mainly on the framework since the continuous nature of the signal does not require peak detection. Second, the additional complexity of detecting peaks and storing the clearances accordingly in the case of simple blade geometry is outlined and the implementation discussed. Ultimately, the solution for the challenge provided by the detection of additional peaks of squealer blades is presented.
The development of the individual signal processing routines shall rely on as little geometry input from the user as possible. This is due to the fact that every input error affects the overall quality. Therefore, the amount of input is limited to the most essential parameters. To be more specific, only the following machine details are requested by the processing units in order to be functional:

- Radius of the Casing
- Amount of Blades
- Angular Sensor Positions

Additionally only parameters concerning routine functionality are set. They include information about the first blade passing each sensor after a key phasor signal, selection of the incremental or decremental referencing system, number of rotations to use for averaging data and number of squealer tips. After the setup, a similar routine flow is processed for every geometry. The basic layout is visualized in figure 6.10.

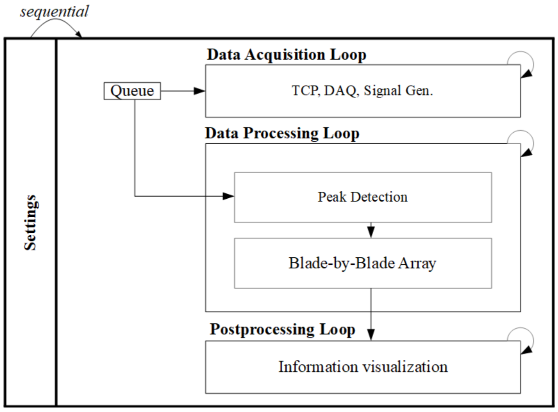


Figure 6.10: Basic LabView routine layout

6.2.1 Shrouded Blades

Settings: The first step is to demand the parameter values. In figure 6.10 the outermost frame resembles a sequence. The routine waits at 'Settings', until the user has specified the necessary input parameters like the radius of the casing or amount of blades in the cascade. As soon as the setup is done, the routine proceeds to spawn three parallel loops. (A) the 'Data Acquisition Loop', (B) the 'Data Processing Loop' and (C) the 'Postprocessing Loop'. All these loops are infinite while-loops, being iterated as often as needed until the entire routine execution is stopped.

Data Acquisition: LabView's convention concerning shifting data from one loop to another requires for the displayed 'queue', a data package designed to transport data within LabView. In the present case, a TCP server is initialized within loop (A), in order to acquire Matlab's simulated signal. In real-world measurement, this step can be replaced by DAQ routines that read measurement data from according devices. In either case the data gets picked up and is attached to the end of the queue.

Data Processing: As soon as this has happened, the queue is updated and therefore data available within loop (B) that is responsible for the data processing. This entity waits until the queue is not empty, then takes the data from the beginning of the queue, optionally filters the data and runs various processing steps. In the case of shroud signals the data does not have to be processed, since every sample within the array relates to a clearance value over time. Therefore the data is directly stored and made available for the loop (C), where the clearance can be postprocessed and additional information acquired.

Data Post-Processing: Since the clearance between the sensor and the shroud is already available, the shaft's location can be approximated with the method mentioned in section 6.1 in the next step. At a certain angular position of the rotor θ_0, three sensors record arbitrary values that are used to solve the circle's equation and thereby obtain the momentary shaft location for θ_0. Doing this for multiple values of θ_i within the course of one revolution of the rotor shows the shaft's orbit.

Apart from immediately calculating the shaft center for every set of sensor clearance values available, the data can instead be buffered over the course of multiple revolutions. By subsequently averaging the clearance values for each sensor corresponding to each θ_i of the rotor, an average clearance is obtained. The result of approximating the shaft's location using these values is a more stationary picture of the orbit. Note, that at an excessive amount of revolutions used for averaging, the resulting orbit does not show short-lasting effects.

Additionally, another option for orbit prediction can be introduced. Instead of using clearance information at each spatial step θ_i to estimate the orbit, all clearances for one sensor over the course of one revolution can be averaged, resulting in a mean clearance per revolution. If this average value of each sensor is used to inscribe the circle, the resulting point is the centroid of the shaft's orbit line, equal to the stationary offset defined in the machine model. These combined data processing steps lead back to the information specified within the creation of the machine model 5.1.1:

- Offset: The stationary component of the shaft's centerline displacement within the casing.
- Eccentricity: The current distance between the stationary offset and the absolute position.

Figure 6.11 shows LabView's output of a shaft simulated at an offset of [1,1] mm and an eccentricity of 0.5 mm. The red crosses indicate shaft center locations at discrete points. The black cross at the center of the red orbit is the processing result of the shaft's stationary offset. The entire visible graph shows one full revolution of the rotor.

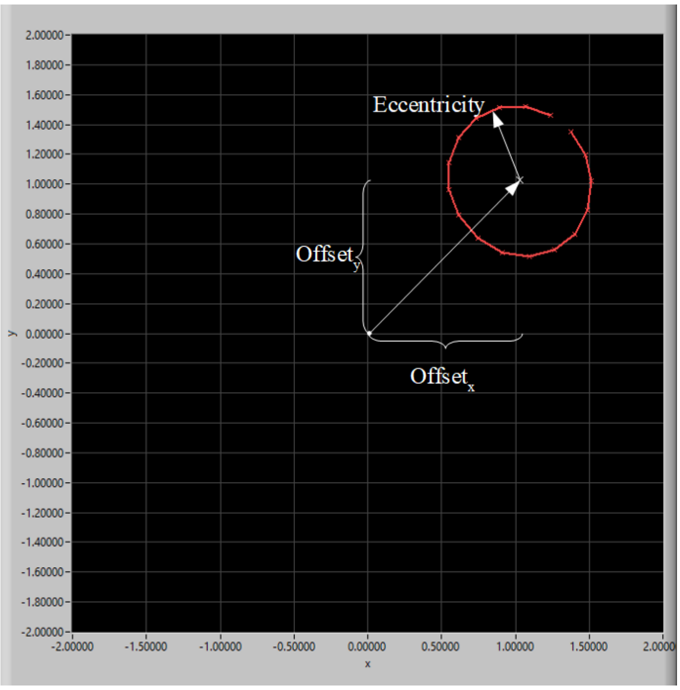


Figure 6.11: Orbit information displayed in LabView

Additionally, since the shaft's orbit was obtained, the minimum clearance between the rotor and the casing can be approximated. In accordance with the procedure introduced in section 6.1.6, the LabView routine calculates the minimum clearance for every position of the shaft (red crosses in figure 6.11) and returns the smallest radial rotor-casing clearance found in combination with the angular position. The result is plotted in a graph in LabView, displayed in figure 6.12. The dashed black line shows the position of the centered rotor, the red cross the rotor's center and the thin black line the casing. The rotor in it's closest state is represented by the thick black line. The red circle indicates a clearance limit that the rotor should never exceed. Note, that this plot does not display true distances since the machine dimensions exceed clearance variation by far and the information would not be conclusive otherwise. Given the fact that the figure has been created using an offset of [1,1] mm and an eccentricity of 0.5 mm this is the resulting minimum rotor-casing clearance for the orbiting shaft shown in figure 6.11 before. Note the angle of minimum clearance of 45 deg that is caused by the definition of a stationary offset in the same direction and a circular orbit.

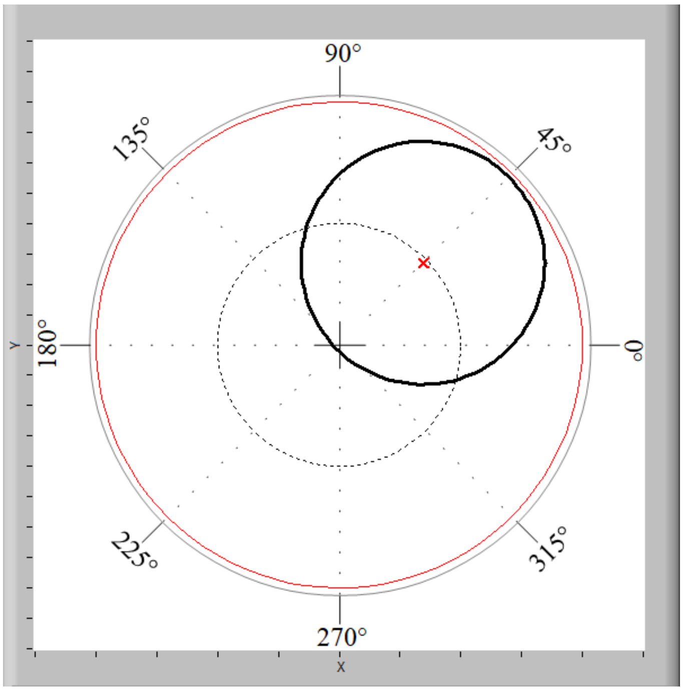


Figure 6.12: Minimum rotor clearance location displayed in LabView

6.2.2 Simple Blades

As soon as the signal shows pulsing instead of continuous character, the fundamental routine from the previous section has to be extended. The main difference is within the data processing loop (B), where the acquired signal has to be processed unlike before.

Data Processing: Instead of having an entire array full of clearance values, the bladed signal consists of peaks that occur at the blade-passing frequency. As has been proven in section 6.1.2, only the peak amplitude of each signal spike contains information of the clearance between the sensor and the blade. In conclusion to that, after extracting the data array from the queue's end, the data is optionally filtered and then handed on to LabView's peak detection sub-routine. The second order polynomials are fit into successive packs of 'width' data points using the least squares method and the maxima are determined. The resulting array containing peak amplitudes is passed on to an entity handling the storage of the peak values, where the individual capacitance values can be referenced with the according blade. The signal of the simple blade Matlab routine displayed within the LabView processing interface can be seen in figure 6.13 in white. The result of the peak detection is indicated using green crosses.

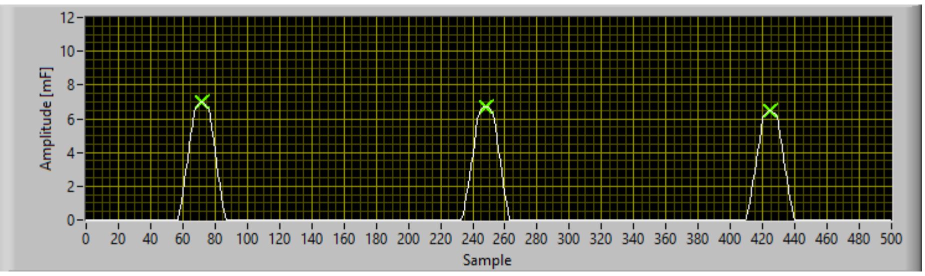
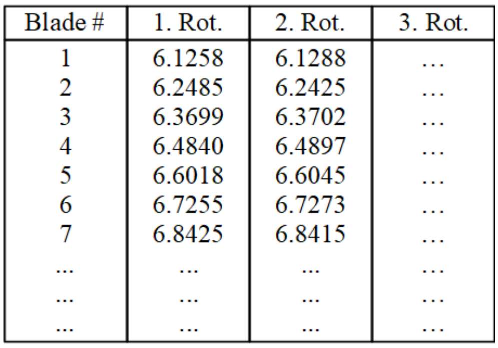


Figure 6.13: Peak detection: signal and peak information of simple blades

In order for the referencing to be functional, four additional settings are introduced:

- First Blade Number for Sensor 1
- First Blade Number for Sensor 2
- First Blade Number for Sensor 3
- Incremental or Decremental Numbering System

The upper three ensure that the first clearance value detected is referenced with the right blade. After that, every successively detected peak value is either stored an index above (incremental) or an index below (decremental) the one before. As soon as amount-of-blades amplitudes are detected and stored, the entire storing process restarts at the initial blade and the next dimension. This allows for the user to define a numbering system on the machine and have the routine use the same system. The result of this process is the multidimensional array containing blade-by-blade clearance information in form of capacitance, displayed in figure 6.14.



Blade #	1. Rot.	2. Rot.	3. Rot.
1	6.1258	6.1288	…
2	6.2485	6.2425	…
3	6.3699	6.3702	…
4	6.4840	6.4897	…
5	6.6018	6.6045	…
6	6.7255	6.7273	…
7	6.8425	6.8415	…
...	...	...	…
...	...	...	…
...	...	...	…

Figure 6.14: Clearance data in terms of capacitance for multiple rotations

Data Post-Processing: First, since the clearance data does not correspond to actual distance yet, the data has to be transformed. Note, that this was not necessary at the shrouded routine, because there the simulated signal has always been a geometric distance. In order to obtain the distance between the sensor and a blade tip, a calibration curve according to section 6.1.4, containing the information of known distance-capacitance combinations, is used. By using linear interpolation, every value for capacitance is linked with a corresponding distance and stored as clearance data in terms of distance. The resulting array for the simulation of 10 revolutions can be seen in figure 6.15, displaying the resulting blade-by-blade clearance underneath one sensor in LabView. The red indicators are the maximum values detected, the green ones indicate the minimum and the white line represents the average clearance.

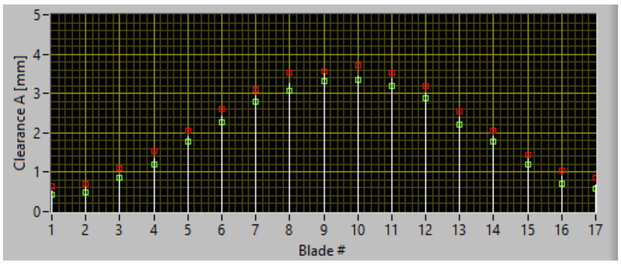
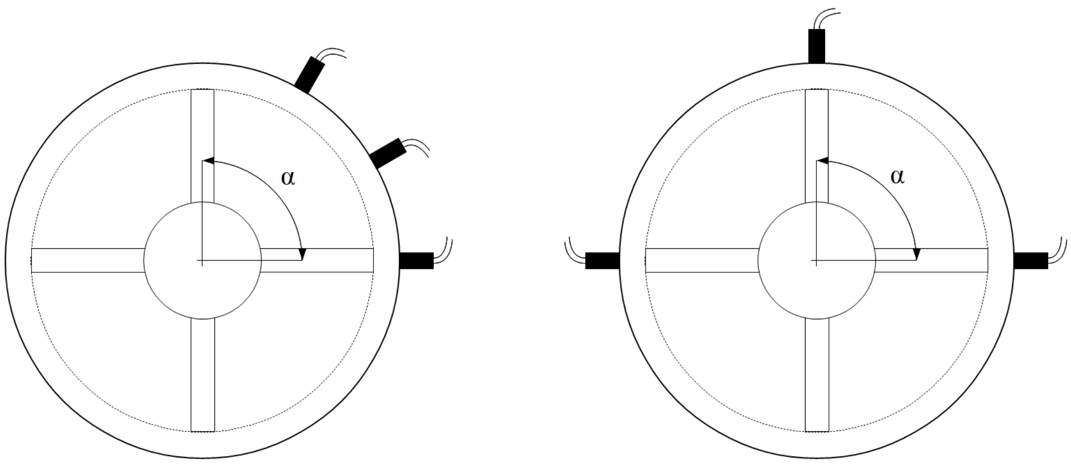


Figure 6.15: Clearance data between blade and sensor of a rotor with 17 blades

Second, the same possibility of computing the shaft's location by the means of data of three arbitrary sensors is available. In the prior case of shrouded blade configuration, the data was assumed to be continuous, meaning that the three sensors were detecting clearance data at all times. Since the sampling frequencies of the sensors were assumed to be synchronized, the three clearance values were captured simultaneously and thereby always corresponded. At the present case of pulsing signals the circumstances change. Even though the sensors sample rates are again synchronized, the blades passing underneath them are not. Figure 6.16 highlights the problem at hand. If the sensors are not adjusted to the blades (a), the sensors pick up on a blade passing by one after another, leading to a time difference in between referencing data. This means that the three required points located at the circle's circumference are not captured at the same time. As a result of the shaft being at a different position for each of the three recorded clearances, the results for equations 4.63 to 4.65

$$x_C = -\frac{B}{2A}; \quad y_C = -\frac{C}{2A}; \quad r = \sqrt{\frac{B^2 + C^2 - 4AD}{4A^2}}$$

are corrupted. In order to maintain the possibility of extrapolating the shaft's exact location from clearance data, the sensors have to be adjusted to the angle distributing the blades α like in figure 6.16 (b).



(a) Unadjusted Setup

(b) Adjusted Setup

Figure 6.16: Problem of sensor placement at a finite number of blades

Note, that the problem caused by sensor misalignment equally affects orbit prediction using clearance data that has been averaged over multiple revolutions. The shaft's movement in times between clearance data being available can not be tracked and therefore still leads to the false approximation of the orbit. Consequently the prediction of the minimum rotor-casing clearance loses accuracy since it is based on the correct approximation of the shaft's location and it's radius. The effect the sensor adjustment has on accuracy while calculating orbit, stationary offset and therefore on minimum clearance prediction shall be discussed and analyzed with more details later on in this thesis.

6.2.3 Squealer Blades

The differences between the routines for simple blade geometry and squealer configuration are limited to the processing of number-of-squealer-tips-detected times as many peaks as before. This routine offers the potential to deal with signals that have more than one peak per blade, which leads to the introduction of another input parameter:

- Number of Tips

According to this setting, the routine reacts with appropriate measures.

Data Processing: In order to be able to successfully deal with a wide range of signals, a convention is established:

> In this thesis, the first detectable peak of the blade is treated as information carrier.

This is in accordance with tip timing application that usually detects the time of blade arrival based on the first time a predefined threshold is exceeded. Furthermore, the first peaks seems the obvious choice due to the fact that contact calibration might be difficult, when having to reference a part of the blade that is hard to access. Note, that it might be required to use the highest, local peak since only there a minimum clearance for the blade is encoded. This thesis neglects this fact by assuming the individual squealer tips to be of equal height. If this assumption proves to be flawed in real world application, the routine has to be altered and the highest peak has to be processed.
Following the stated convention, the incoming data array - after being optionally filtered - is handed to the same peak detection sub-routine as before. Then, in case of two peaks per blade being detected, the peaks are passed on to a new sub-routine that effectively eliminates every second peak of the blade, leaving behind only clearance information that is encoded into the first tip. The result of this process is made apparent in figure 6.17. It displays the capacitance signal of the squealer blade Matlab routine within LabView's interface in white and the found peak using the green cross. Note, that the second peak

caused by the right squealer blade (between samples 280 and 320) has been found, but has already been sorted out. Furthermore, the figure displays a red cross at sample 0. This is a result of the solution implemented to avoid detection problems at the edge of arrays introduced in section 6.1.2. The cross indicates the amplitude of a peak found at the right side of the last data set, which is why it is placed at sample 0. Without the additional measures taken to detect peaks at the edge of arrays this clearance information would have been lost and the referencing of the data to corresponding blades would have been corrupted.

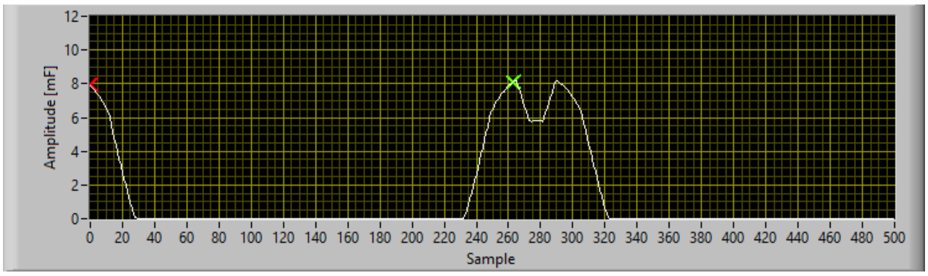


Figure 6.17: Peak detection: signal and peak information of squealer blades

Subsequently, the resulting blade-by-blade array is made available for the post-processing loop (C), where the same steps can be taken as before. There is no difference in processing compared to simple blades, as soon as the signal has been reduced to clearance values. This statement includes the limitations of orbit prediction available for unadjusted sensor setups.

7 LabView Performance

After showing the range of possibilities that the presented LabView routines give the user to process various radial rotor-casing clearance data of an axial thermal turbomachine, this chapter deals with the stability and accuracy of the results. The individual tests are all conducted with selected parameters set to be constant:

- Radius Casing: 250 mm
- Radius Shaft: 50 mm
- Sensor Count: 3
- Peak Detection 'Width': 3 Elements

For the tests, the already discussed machine model, once more displayed in figure 7.1, is simulated with various input parameters. The produced signals shall first elaborate the LabView routines concerning requirements for the measurement system and the influence of the sensor setup. Subsequently, the processing routine is given signals containing geometric and measurement system imperfections. The results are representative of a worst case scenario. This will be followed by conducting a parameter variation within LabView in order to improve the results that have been returned with imperfect input signals.

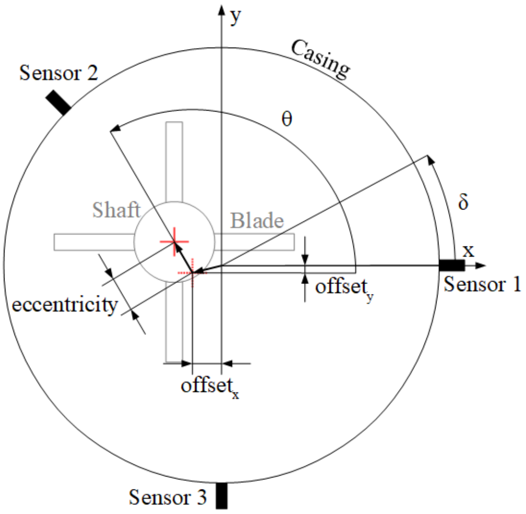
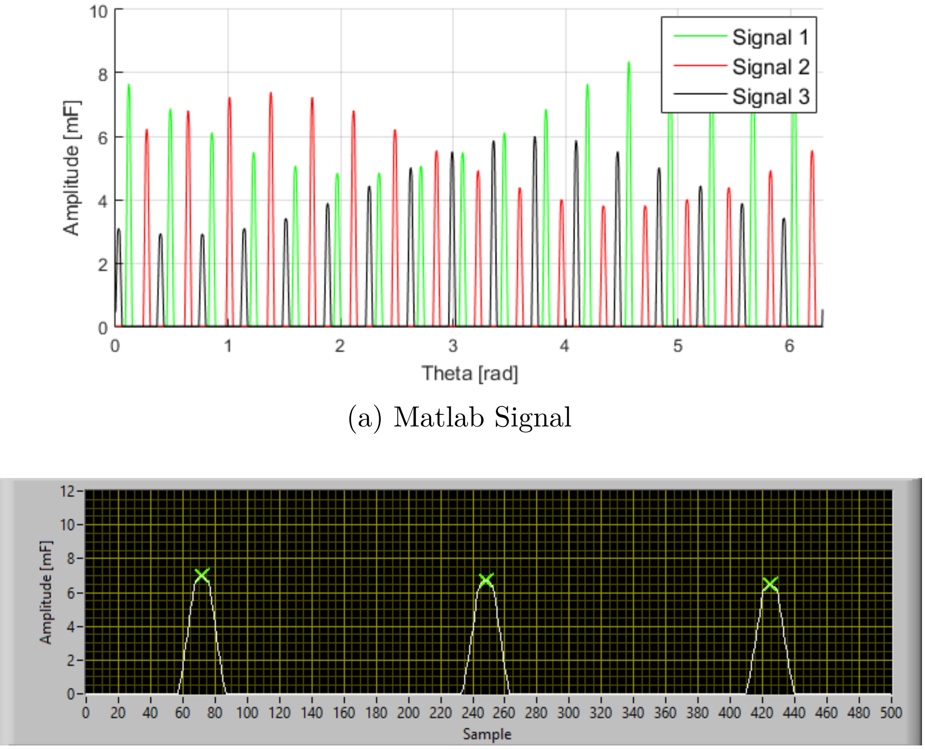


Figure 7.1: Machine Model

For the performance tests a general work flow is introduced. First, the machine model (figure 7.1) is simulated. After every incremental calculation step, the resulting signal (figure 7.2 (a)) is transfered via TCP to LabView. The continuous stream of clearance data in terms of capacitance is buffered by LabView and the array based peak-detection is applied to each data set (figure 7.2 (b)).

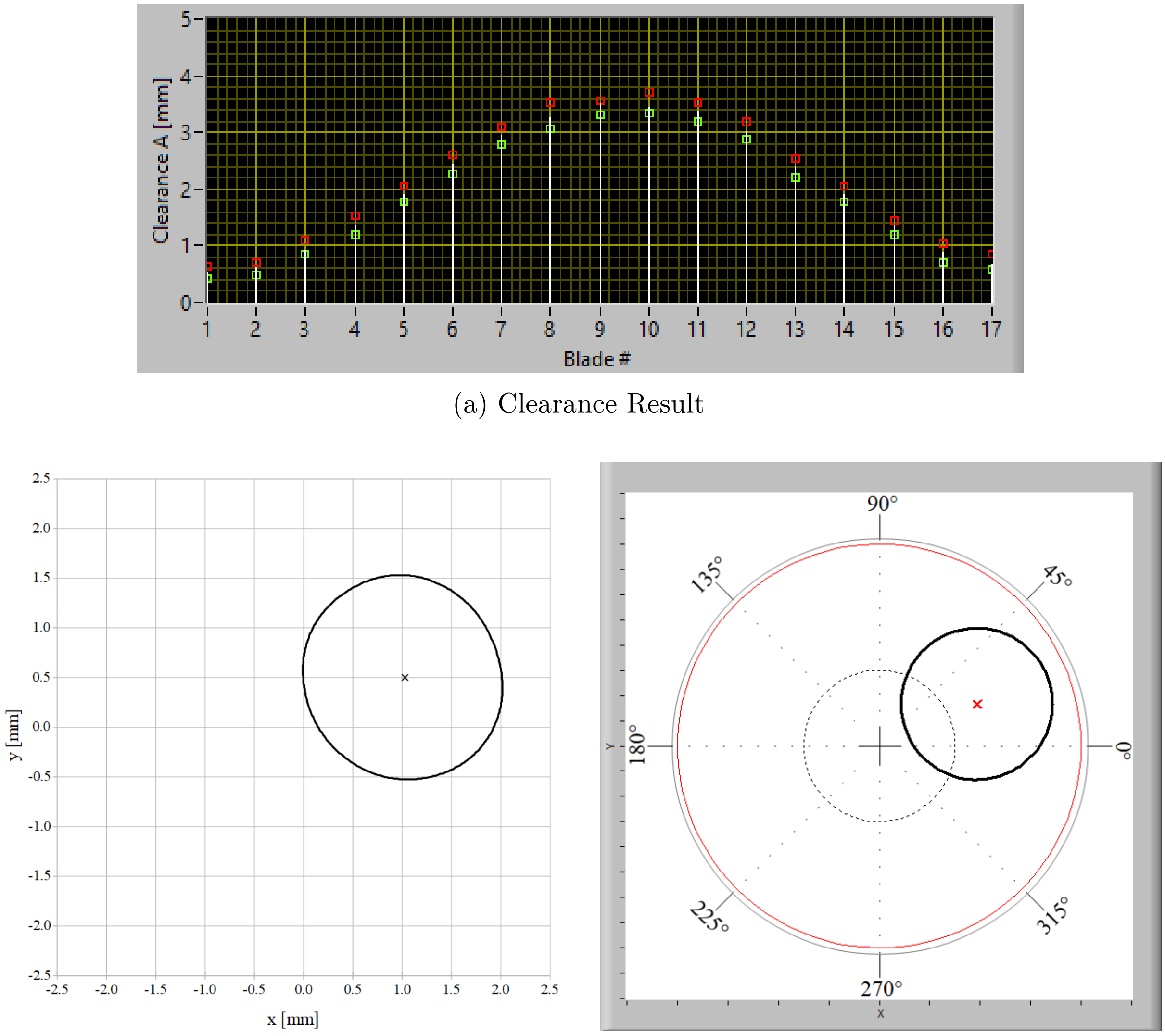


(a) Matlab Signal

(b) LabView Signal

Figure 7.2: Transfer of the simulated signal to LabView

The results of the peak-detection are referenced with the according blades and stored for further processing. Before the next steps lead to the output of the desired information, the capacitance values are transformed into a distance using linear interpolation on a calibration curve. Note, that this is done using the same calibration data throughout the entire chapter in order to acquire comparable results. The resulting clearances between the blades and the individual sensors are then returned to the user (figure 7.3 (a)). After that, the gathered clearance data is used to approximate the shaft's location. The results for the processing of an entire revolution of the shaft is it's orbit and an estimated stationary offset (figure 7.3 (b) - LabView results exported to Excel). The final processing step calculates the magnitude and angular position of the rotor's minimum clearance within the casing (figure 7.3 (c)) using the most recent orbit data.



(a) Clearance Result

(b) Orbit and Offset Result

(c) Min. Clearance Result

Figure 7.3: LabView Information Output

7.1 Sample Rate Requirements

The first essential information that can be obtained with the introduced systems is the required sample rate of the measurement system. For that purpose, the clearance signals are simulated using various different spatial resolution rates and the effect on the processing results will be shown.

Before that, the frequency of the signal that should be measured has to be defined closer. When looking at a turbomachine, seeing less than 3000 RPM is a rare case since high rotational speed is essential for the realized work. When assuming 6000 RPM as operational parameter for an axial turbine with 71 blades, the Blade-Passing-Frequency (BPF) is calculated according to

$$BPF = \frac{6000}{60} \cdot 71 = 7100 \frac{1}{s} = 7100\ Hz \tag{7.1}$$

While introducing AD conversion in section 3.2.2, the expression 'sample rate' was used and the Nyquist theorem as a boundary for successful signal detection without aliasing was introduced with equation 3.5. When applying the Nyquist theorem to the blade-passing frequency, the need for a sample frequency of 14200 Hz becomes apparent. Unfortunately, this frequency is not even close to the sample rate actually needed for measurements with acceptable accuracy. The ideal case at this setup would be to sample the signal at points of a blade being the closest to the sensor (maximum amplitudes). In this case the clearance results are accurate. On the other side, if the sensor only samples an arriving or leaving blade, the peak amplitude would be reduced and false information obtained. Recall, that the peak amplitude is directly linked to the clearance magnitude between sensor and rotor and therefore the detection of the maximum amplitude is indispensable. In order to prove these statements and show their impact, the Matlab routines are used and the results are presented in the following pages.

Over- and Undersampling: In the present simulation the effect of undersampling by limited sample rates can be shown by increasing the spatial stepping width $\triangle\theta$. This leads to the calculation of less points for a full revolution of the rotor and thus less information being sent to the processing unit. Figure 7.4 shows the results of three different stepping widths. In black an oversampled peak with $\triangle\theta = 1500$ is displayed. The resolution by data points is high enough to pick up on the blade's movement while passing the sensing area. The results in green for $\triangle\theta = 200$ and $\triangle\theta = 175$ are both undersampled. Note, that the deviation of the green peak from the correct amplitude in black might be considered acceptable, while at almost the same stepping rate the result might be a peak with half of the true maximum value, shown in red. This uncertainty leads to the necessity of oversampling.

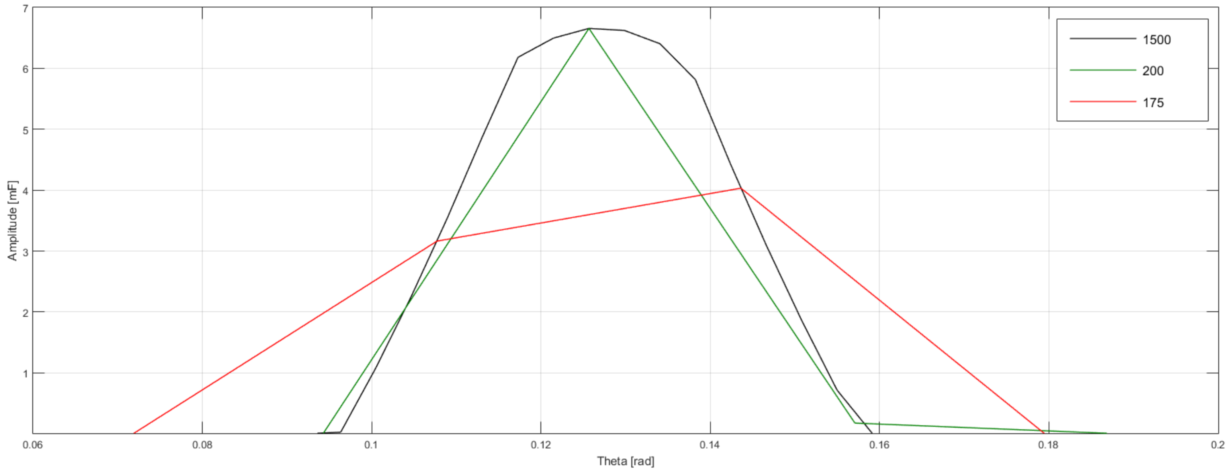


Figure 7.4: Clearance signals for different spatial stepping rates $\triangle\theta$ at simple blade geometry

Generally speaking, higher sample rates allow for resolving more details than lower rates. This statement becomes even more important for difficult geometries, like the squealer blades in figure 7.5. The loss of critical information compared to the oversampled peak in black starts at higher values for $\triangle\theta$. Already at $\triangle\theta = 750$ the blue sensor response data shows a reduced peak amplitude for the second tip. Decreasing the steps per revolution further to $\triangle\theta = 200$ the geometric features of the squealer blade are not resolved anymore and the drop in amplitude renders the data useless.

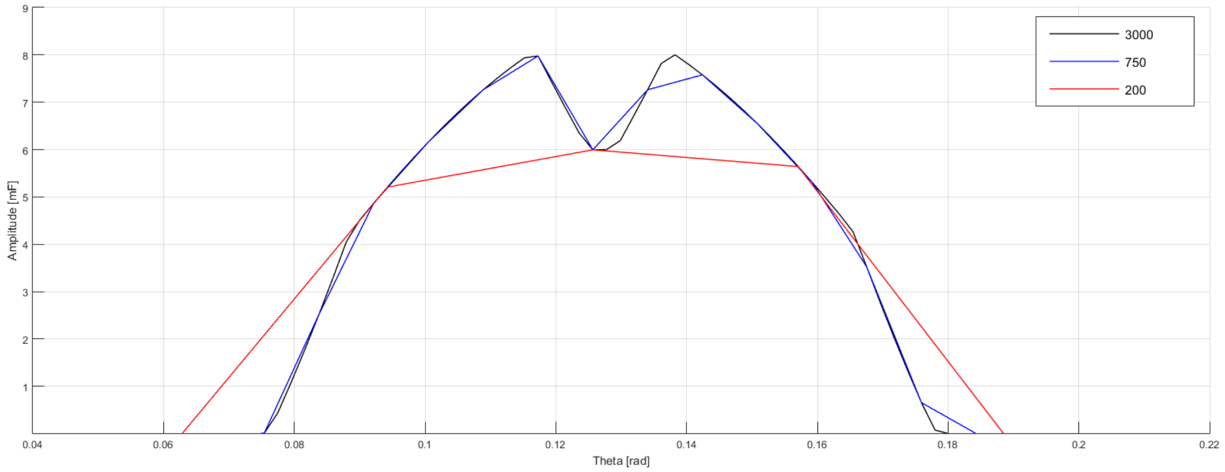
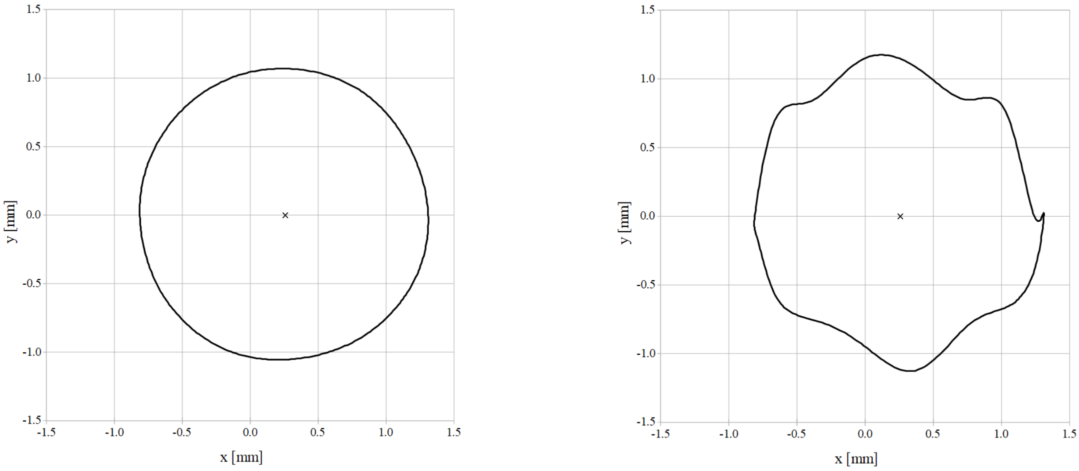


Figure 7.5: Clearance signals for different spatial stepping rates $\triangle\theta$ at squealer blade geometry

Effect on Processing: The problem from a loss in signal amplitude is linked the correlation the peak has with the clearance between blade and sensor tip. Since a high signal amplitude is equal to a small clearance, the effects of undersampling lead to false information being processed in the monitoring routine. The reduced peak amplitude is detected, the value translated into a distance and a false clearance margin is displayed. The effect can be shown using the established systems by processing an oversampled and undersampled signal. The simulations are conducted on a machine with a rotor with 17 blades, a stationary offset of [0.25 0] mm, an eccentricity of 1 mm and following geometry:

- Radius Casing: 250 mm
- Radius Shaft: 50 mm
- Blade Length: 197 mm

The LabView results for the processed shaft locations can be seen in figure 7.6. The stepping rates are $\triangle\theta = 3400$ for the oversampled results (a) and $\triangle\theta = 500$ for the undersampled results (b). The orbit in (a) gives a good representation of the set up machine parameters, while the undersampled signal (b) has led to the rather chaotic orbit on the right.



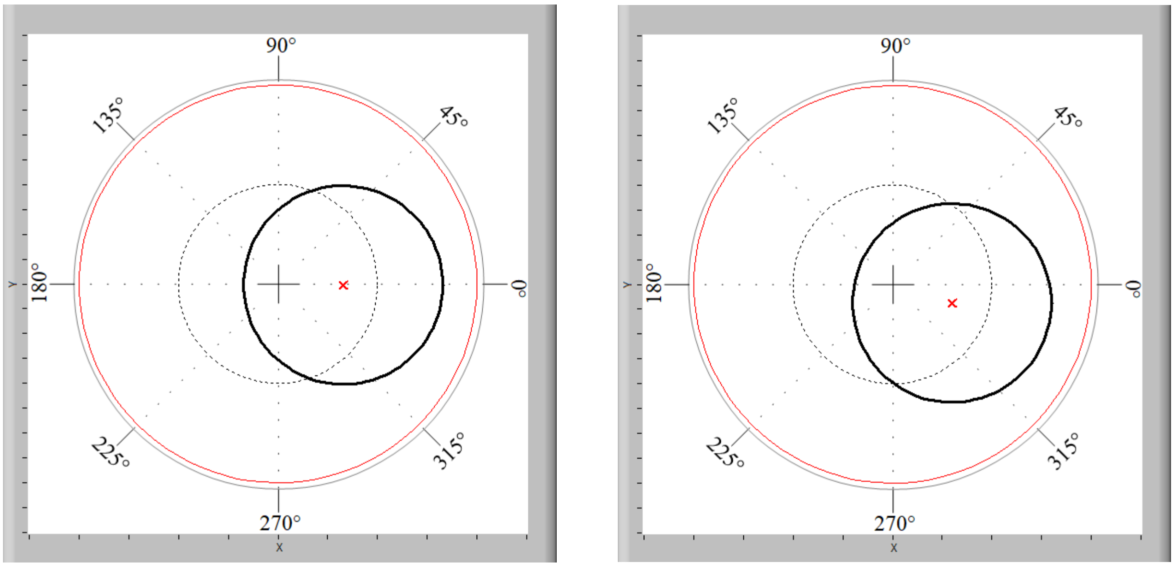
(a) Oversampled Signal

(b) Undersampled Signal

Figure 7.6: LabView orbit results of over- and undersampled signals

As a result of the approximation of the shaft's path LabView estimated the smallest available clearance between rotor and casing. Note, that the expected direction of the minimum clearance should be at exactly 0 deg and show a magnitude of 1.75 mm due to the setup that has been made. The resulting predictions created by LabView can be seen in figure 7.7. While the location at the oversampled signal has been estimated with 359.9 deg and

1.74942 mm (a), the processing of the undersampled signal returned 342.7 deg and 2.2041 mm (b). Note, that this is only one of the possible outputs for the undersampled data. The blades were detected with alternating precision every subsequent revolution. This led to a variance in accuracy of the predicted orbit and thereby to the angular position and margin of the smallest clearance to be estimated differently for every revolution.



(a) Oversampled Signal

(b) Undersampled Signal

Figure 7.7: LabView min. rotor clearance results of over- and undersampled signals

Requirement: Since the effect of undersampling has been shown, a limit for accurate results has to be derived. In table 7.1 rotor-sensor clearance results of simple blades for different $\triangle\theta$ steps are displayed for one sensor. The signal was simulated using a centered rotor setup with a known, constant clearance between the rotor and the casing at the entire circumference of 1 mm. The blade row contained 17 blades, leading to the same amount of peaks displayed for one full rotation. Furthermore, assuming an angular velocity of one revolution per second leads to a BPF of $\frac{1}{17}$ Hz. Starting at a sample rate of about 50 times the blade-passing frequency ($\frac{1}{17} \cdot 50 = \frac{1}{850}$), which correlates to a stepping rate of $\triangle\theta = 850$, the peaks are resolved accurate enough for stationary results. After that, some peaks are cut short while others are sampled correctly, making a calibration impossible. The necessary bandwidth for accurate signals lines up with the results of Haase and Haase (2013). The authors state the necessity of sample frequencies that are about 30-times the signal frequency for accurate measurement.

Table 7.1: Clearance results for a variation of $\triangle\theta$ for simple blades at a 1 mm gap

$\triangle\theta$	Min. Clearance [mm]	Avg. Clearance [mm]	Max. Clearance [mm]
3400	0.999809	0.999809	0.999809
1700	1.00056	1.00056	1.00056
850	1.00318	1.00318	1.00318
500	0.89178	0.94511	0.99845
200	1.00090	1.47334	2.59625

Table 7.2 shows the clearance results for squealer blades at various sample rates for the same setup as before (1 mm constant clearance between rotor and casing). The same pattern as has been witnessed with simple blade geometry can be observed, even though the loss of amplitude starts at higher values of $\triangle\theta$. This can be seen in the increase of clearance starting at $\triangle\theta = 3400$, which correlates to a smaller capacitance peak being detected. ($Clearance \approx \frac{1}{Signal}$) While the resulting values still appear to be stationary down to $\triangle\theta <$ 1000, the increase in clearance shows that the tips of the two peaks have been cut (compare blue peak in figure 7.5). The uncertainty this implies has to be avoided.

Table 7.2: Clearance results for a variation of $\triangle\theta$ for squealer blades at a 1 mm gap

$\triangle\theta$	Min. Clearance [mm]	Avg. Clearance [mm]	Max. Clearance [mm]
5000	0.99924	1.00019	1.00373
3400	1.05375	1.05375	1.05375
1700	1.07669	1.07669	1.07669
850	1.28545	1.28545	1.28545
500	1.15024	1.21845	1.29791

Even though this thesis talks about sample rate in sense of $\triangle\theta$, both of these values represent spatial resolution. Due to this fact, the requirements for accurate measurement conducted on turbomachinery apply to the simulation domain as well as to real-world measurement. To sum up the statements made throughout this section: The necessary sample rate has been proven to be dependent on geometry, the angular velocity and the blade count and lies at around 30 to 50 times higher than the blade-passing frequency.

7.2 Error Propagation

It can be stated that the prediction of the smallest clearance between rotor and casing is directly linked to the accuracy of the approximation of the shaft's orbit according to equation 6.5:

$$Min.\,Clearance = R_{Casing} - (\sqrt{x_C^2 + y_C^2} + r)$$

The values x_C, y_C and r are the solution of the system of equations inscribing the circle into the rotor's blade tips/shroud. Every error that has been introduced while localizing the shaft affects the computation of the min. rotor-casing clearance equally. This has become apparent in the previous section, where undersampled data was processed. The result was a transient approximation of angle and magnitude at the simulation of stationary machine parameters. Instead of repeating the same statements for the rest of the performance study, the accuracies shall be linked within this section. By doing that, every improvement of the shaft's localization accuracy can be transfered to the estimation of the smallest radial clearance between rotor and casing.
In order to show the impact the precision of shaft localization has on the equation above, the Gaussian error propagation can be applied:

$$u_f = \sqrt{\sum_{i=1}^{N} (\frac{\partial f}{\partial x_i} u(x_i))^2} \tag{7.2}$$

In this equation, u_f represents the deviation of function f caused by all parameters i of this function. The idea behind this equation is to show the impact every independent parameter has on the output of the function itself. In the present case, f is equal to equation 6.5:

$$f = R_{Casing} - (\sqrt{x_C^2 + y_C^2} + r)$$

and the parameters are the radius of the casing, the shaft's center position and the radius of the inscribed circle. The partial derivations of f in any parameter i are

$$\frac{\partial f}{R_{Casing}} = 1 \tag{7.3}$$

$$\frac{\partial f}{r} = -1 \tag{7.4}$$

$$\frac{\partial f}{x_C} = -\frac{x_C}{\sqrt{x_C^2 + y_C^2}} \tag{7.5}$$

$$\frac{\partial f}{x_C} = -\frac{y_C}{\sqrt{x_C^2 + y_C^2}} \tag{7.6}$$

Summing up the expressions, the final equation for the deviation of the calculation of the smallest clearance can be written as

$$u_f = \sqrt{(\frac{\partial f}{R_{Casing}} \cdot 0)^2 + (\frac{\partial f}{r} u_r)^2 + (\frac{\partial f}{x_C} u_x)^2 + (\frac{\partial f}{y_C} u_y)^2} \tag{7.7}$$

$$u_f = \sqrt{\frac{x_C^2 u_x^2 + y_C^2 u_y^2}{x_C^2 + y_C^2} + u_r^2} \tag{7.8}$$

Note, that the deviation of the casings radius has been set to zero since the value is a constant that is entered by the user. The other values and their deviations all influence the accuracy of the final result.

$$Min.\ Clearance_{real} = Min.\ Clearance_{ideal} \pm u_f \tag{7.9}$$

Due to this connection, the main objective for the rest of this thesis is to determine influential factors on the orbit and offset prediction and to derive ideal settings to optimize the accuracy of the approximation. Consequently, the results of the estimation of the smallest clearance will improve in precision.

7.3 Sensor Setup

The measurement systems sample rates are only one of the factors that can influence the processing results. As has been mentioned in section 6.2.2, the lack of corresponding positioning data in terms of rotor-sensor clearance data can cause problems when using the unambiguous definition of the circle to locate the shaft's center position. This phenomenon can occur when angular blade and sensor distribution are not adjusted to each other. At unadjusted setup, the rotor blades are passing the sensor successively, in the worst case all three at a different time. This leads to the rotor moving to a different location in between clearance data being recorded and therefore the reconstruction of the shaft's center being corrupted. This section will analyze this circumstance and additionally present informations about the impact of different angular sensor positions - aligned with the blades or not.
Since both, the simple blade and the squealer blade routine, break down to clearance values after peak detection and can be assumed to be equally precise in case of correct calibration and setup, the simple blade routine is used for signal generation from this point on, due to less computational power being needed for the simulation with high sample rates.

7.3.1 Sensor Alignment

Eccentricity: The distance from the stationary offset to the current position of the shaft has been treated as a constant up to this point. In reality, the value might even change within one revolution of the rotor. This fact has to be anticipated when processing clearance data gathered using three arbitrary sensor. As has become apparent already, the quality of the information for the user greatly depends on the accuracy of the calculation of the shaft's center position within the casing. Therefore the first step is to show the influence of adjusted and unadjusted sensor setup on the reconstruction of a variety of different eccentricity values. Following parameters are set up to be constant:

- Blade Count: 17
- Offset: [0 0] mm
- Radius Casing: 250 mm
- Radius Shaft: 50 mm
- Blade Length: 198 mm
- $\triangle\theta$: 5000

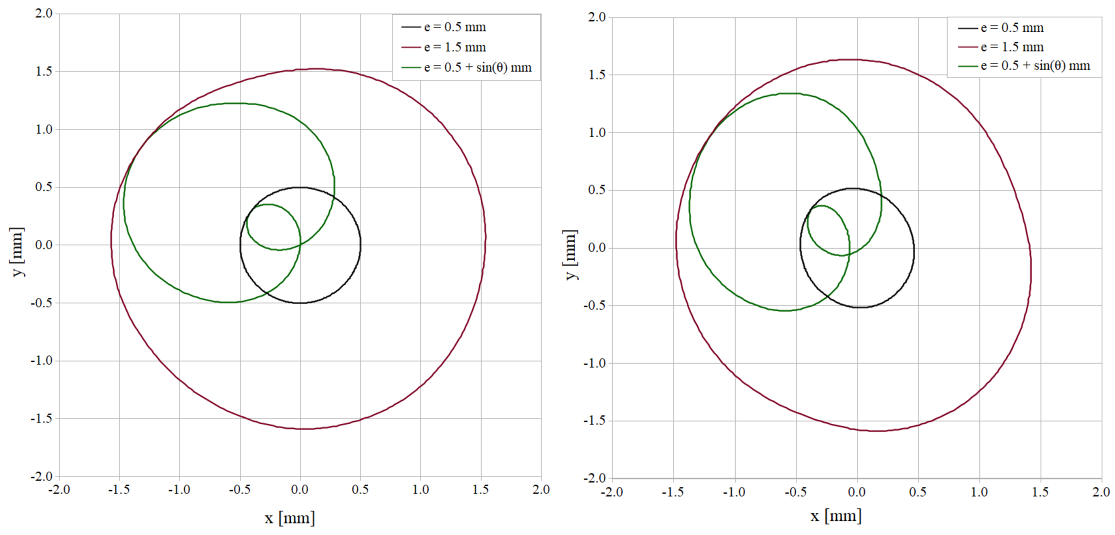
Figure 7.8 (a) shows the results for various settings for eccentricity at adjusted sensors. Two different absolute values have been set - 0.5 mm (black) and 1.5 mm (red) - resulting in circular shaft center paths. Alternatively a variable eccentricity, following the equation

$$Eccentricity = 0.5 + sin(\theta) \tag{7.10}$$

has been computed (green). Even though the path might seem physically irrelevant, this is done to demonstrate the modeling and detection possibilities with only 17 blades at adjusted sensors. Overall, the circular paths are conserved for the absolute values and the 45 deg tilted sine equation fits the expected behavior. Note, that the function has been tilted intentionally by adding $\frac{\pi}{4}$ to θ.

When looking at the unadjusted sensor results, displayed in figure 7.8 (b), the calculation of the orbit seems to suffer a lack of corresponding points to solve the equation system of the circle, as has been predicted. The main difference compared to the adjusted sensor results are the misshaped orbits. This poses the danger of false assessment of the shaft's movement state and subsequently effects the accuracy of magnitude and angular position of the smallest clearance between rotor and casing.

Even thought all of the displayed results show a deflection of their ideal paths, in case of the adjusted sensors the shape of the orbit is contained more accurately than at the unadjusted sensor setup. Since the error of about 0.2 mm in the adjusted case is almost constant for the entire circumference, it can be countered by adjusting the calibration curve, which will be shown later on. This would not improve the shape of the orbits for unadjusted sensors.



(a) Adjusted setup

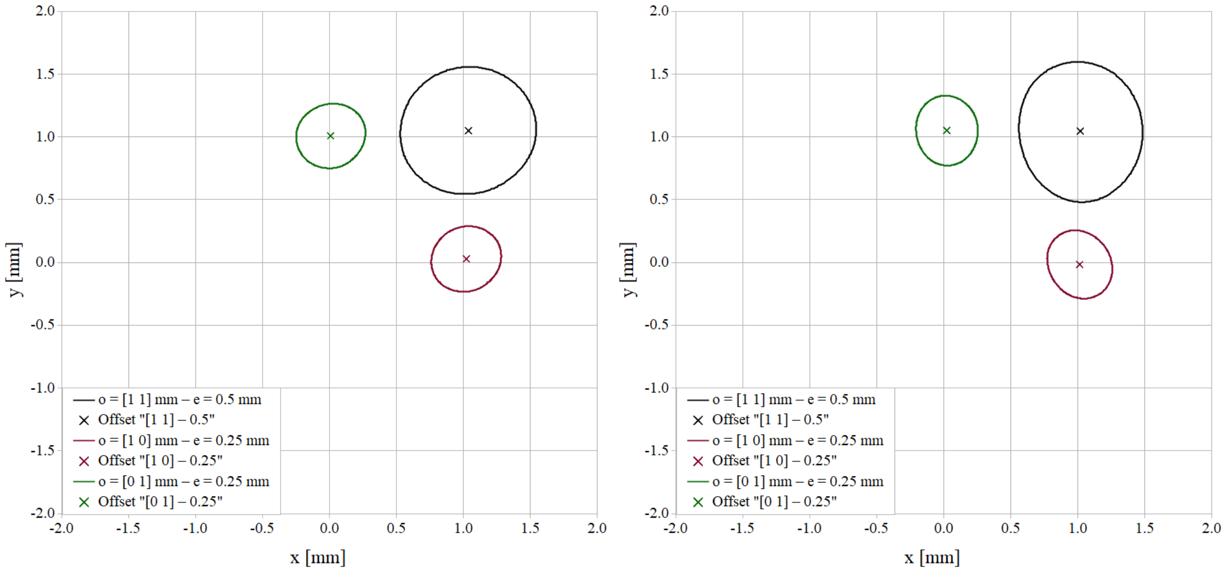
(b) Unadjusted setup

Figure 7.8: LabView results for a variation of eccentricity

Offset: In the previous paragraph the shaft was orbiting the center of the casing [0 0] mm. Now, the effect of sensor alignment on the approximation of the shaft's path with eccentricity and stationary offset can be shown. As well as before, this is done for adjusted and unadjusted sensor setup in order to evaluate the importance of alignment. For this purpose the following parameters have been set to be constant within the simple blade routine:

- Blade Count: 17
- Radius Casing: 250 mm
- Radius Shaft: 50 mm
- Blade Length: 198 mm
- $\triangle\theta$: 5000

Figure 7.9 (a) displays the processing results for the adjusted sensor mounting. As has been the case at the variation of eccentricity, the results represent the circular orbits of the shaft much closer than the orbits displayed in figure 7.9 (b) for unadjusted sensors. The quality is greatly increased by adjusting the sensors at a cascade simulated with 17 blades. Note, that the radius of the orbits in both cases exceed the setup magnitude by around 0.2 mm. In case of the adjusted sensors this can be corrected by more precise calibration unlike with the unadjusted setup, where the elliptical shape of the orbits would remain nonetheless.



(a) Adjusted setup

(b) Unadjusted setup

Figure 7.9: LabView results for a variation of offset and eccentricity

Blade Count: Since a blade count of 17 blades does not resemble reality in most cases, the effect of increasing the number of blades shall now be shown. This will answer the question of whether adjusting the sensors still influences accuracy by the same amount as with less blades. For this purpose the simulation for an eccentricity of 1.5 mm is repeated with 79 blades and then compared to the results from before. The simulation of the signal is done with:

- Blade Count: 79
- Offset: [0 0] mm
- Radius Casing: 250 mm
- Radius Shaft: 50 mm
- Blade Length: 198 mm
- $\triangle\theta$: 5000

When looking at figure 7.10 for the adjusted setup the difference between 17 blades (red) and 79 blades (blue) appears to be neglectable. The true orbit (black) has been inserted into the plot in order to give a reference for the ideal result. Note, that the shape of the orbit is not significantly improved, nor is the accuracy in terms of average radius. There seems to still be a slight overestimation of the eccentricity, what could be corrected by adjusting the calibration curve.

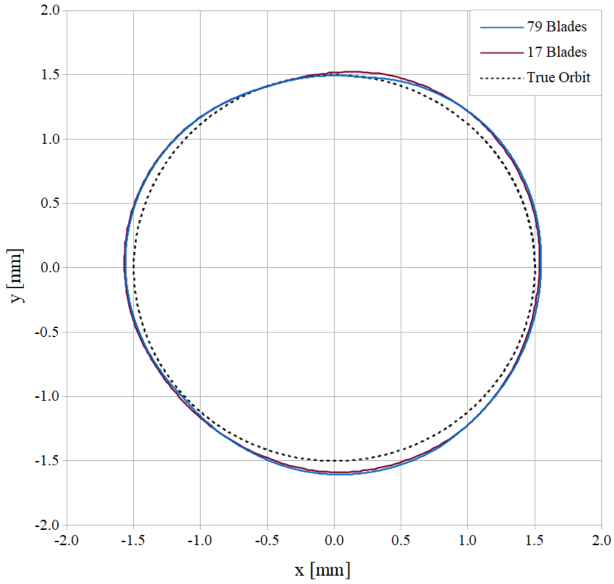
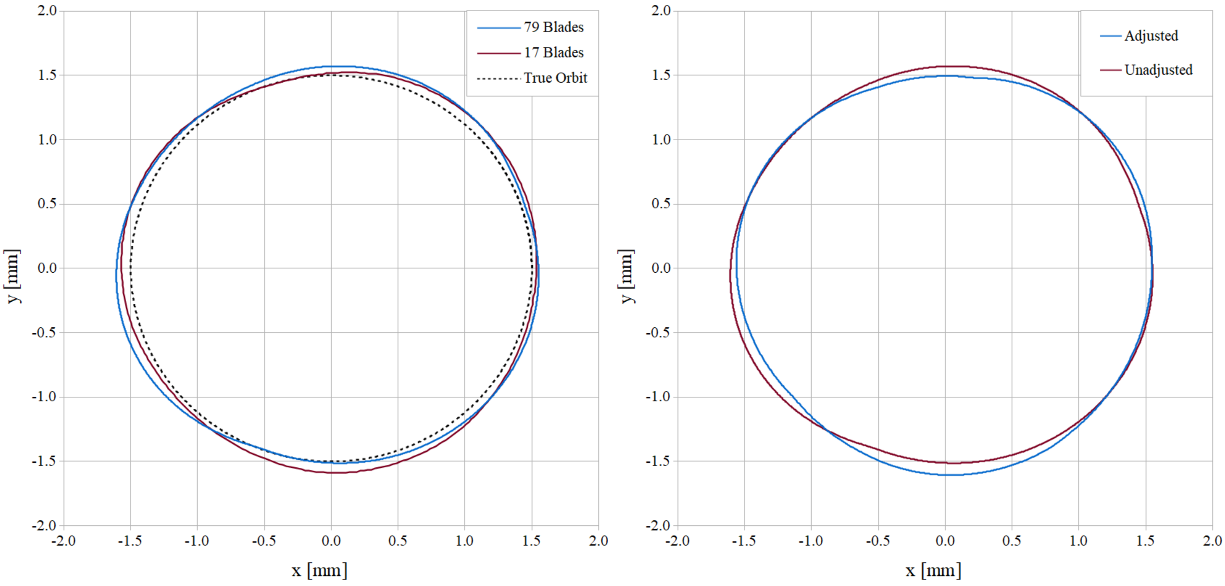


Figure 7.10: LabView results for a varying amount of blades at adjusted sensors

When it comes to unadjusted sensors, the blade count seems to matter significantly. The new shaft orbit for 79 blades (blue), compared to the result for fewer blades (red) from before, shows increased conservation of shape, as is visible in figure 7.11 (a). Figure 7.11 (b) compares the results for adjusted and unadjusted setup with 79 blades directly. The quality of the result for unadjusted setup has improved to the quality of the adjusted setup, which leads to the conclusion that adjusting the sensors to the blades is irrelevant, when measuring at cascades containing enough blades. The explanation for this result is obvious, since with increased amount of blades, the amount of reference points for the solution of the circle's system of equations increases equally. It can be stated that the effects are diminishing for high blade counts. The differentiation between adjusted and unadjusted sensor setups is neglected henceforward and only the adjusted result is computed.



(a) Unadjusted sensor setup

(b) Comparison of adjusted and unadjusted setup

Figure 7.11: Comparison of improvements for unadjusted sensor setup with increased blade count

7.3.2 Angular Sensor Positioning

The deformation of the orbit shape as a result of unadjusted sensor setup and low blade numbers, as has been apparent in the previous sections, can be seen as an extreme scenario. It has been shown that increasing the blade count leads to improvement in shape containment. Nevertheless, since even adjusted setup orbits show deflection from the ideal path to a certain degree, the question whether a certain sensor distribution influences the deformation process remains. By simulating only 17 blades with varying sensor placement, the effect of this placement is amplified. Therefore following settings are made:

- Blade Count: 17
- Offset: [0 0] mm
- Eccentricity: 1.5 mm
- Radius Casing: 250 mm
- Radius Shaft: 50 mm
- Blade Length: 198 mm
- $\triangle\theta$: 5000

The results for the variation of the sensor angles can be seen in figure 7.12. Four different setups are simulated and compared to the true orbit (black):

1. 0° - 22.5° - 45° (green)
2. 0° - 45° - 90° (red)
3. 0° - 90° - 180° (dark blue)
4. 0° - 120° - 240° (yellow)
5. 0° - 42° - 212° (light blue)

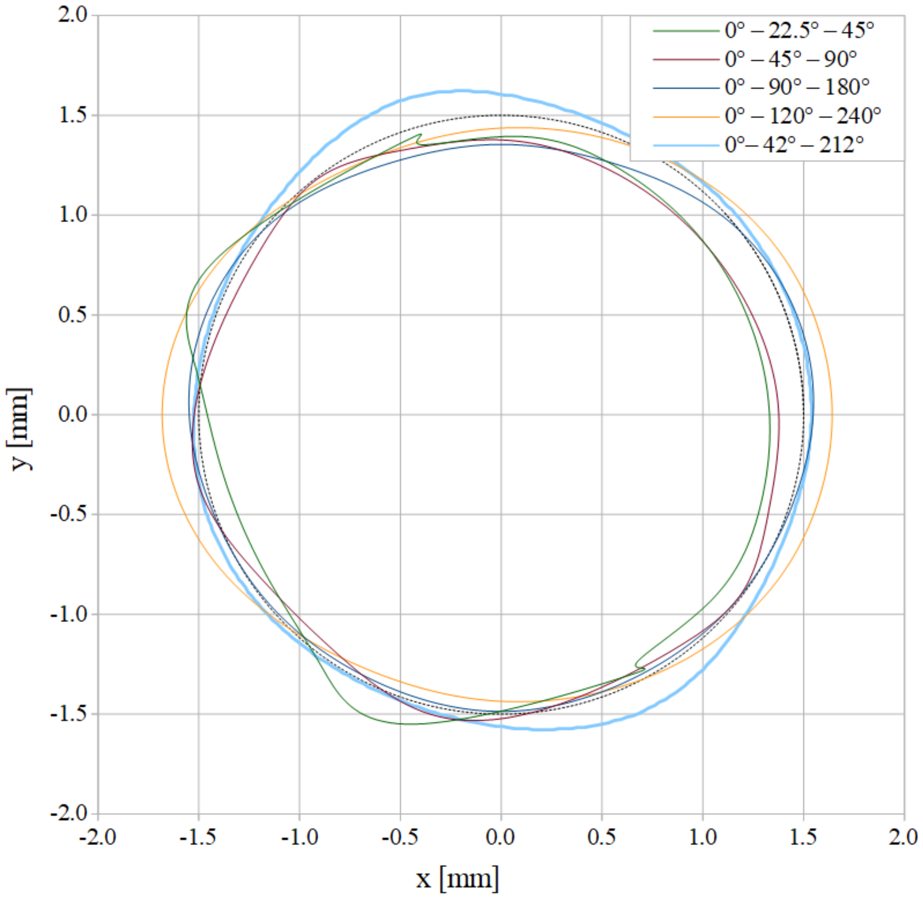


Figure 7.12: LabView results for a variation of sensor distribution

Setup (1) proves to acquire the worst resemblance with the true orbit, with various portions of the orbit showing chaotic behavior, leading to possible misjudgment of the shaft's movement state. Orbit (2), with sensors spaced out over the first 90 deg, shows most deflection in the area of the sensors being present. This is unlike setup (3), where deviation from the true orbit mainly occurs around the second sensor. One of the worst replications of the target orbit was achieved when using unequally spaced out sensors (5). Overall, sensor setup (4) shows the most average under- and over-approximation of the true path of the shaft. In general spacing out the sensors over the entire 360 deg range of the casing seems to improve the accuracy, with the highest angular distances between the sensors being the most effective in conserving average precision throughout one revolution of the shaft.

7.4 Imperfections

Until now the signal has been idealized, which does not represent real-world conditions. In order to show the impact geometric and measurement imperfections have on the approximation of the shaft's location, this section shows the processing of signals from rotors with variable blade length and signals containing noise. As a first step, the signal quality will not be enhanced. Therefore, the results presented in the following pages give an idea about the true impact of these imperfections and serve as negative examples that have to be improved. This will be done in the next step, presented in the following section.

7.4.1 Blade Length

To simulate the condition of variations in blade length a random length deviation has been specified for each blade individually within Matlab. The problem that arises is based on the fact that the calculation of the shaft's location does assume perfect circularity of the circle that is inscribed into the blade tips. In case of variable length this is not accurate anymore. The goal is to highlight the effect various blade length imperfections have on the accuracy of the predicted orbit and therefore on any derived information.
Before that, the matter of blade-sensor clearance measurement accuracy has to be discussed. In table 7.3 the processing results for a rotor with a blade length of 198 ± 0.25 mm is showcased. At a shaft radius of 50 mm and a casing radius of 250 mm, the resulting clearance varies around the mean value $\sigma = 2.02017$ mm. As can be seen in column three and four, the true and the measured clearance show differences in magnitudes of less than 0.01 mm.

Table 7.3: Clearance results at a blade length variation of ± 0.25 mm (centered shaft)

Blade	Length [mm]	True Clearance [mm]	Measured Clearance [mm]	Error [%]
1	197.90546	2.09454	2.09531	0.03691
2	197.92886	2.07114	2.07183	0.03343
3	197.92706	2.07294	2.07364	0.03379
4	197.77144	2.22856	2.22806	0.02234
5	197.91430	2.08570	2.08645	0.03580
6	198.09055	1.90945	1.91019	0.03863
7	197.92092	2.07908	2.07980	0.03483
8	198.01141	1.98859	1.98866	0.00343

Blade	Length [mm]	True Clearance [mm]	Measured Clearance [mm]	Error [%]
9	198.04434	1.95566	1.95613	0.02418
10	198.11302	1.88698	1.88773	0.03955
11	197.95646	2.04354	2.04402	0.02345
12	198.02587	1.97413	1.97440	0.01372
13	198.04779	1.95221	1.95272	0.02588
14	197.94991	2.05009	2.05063	0.02639
15	197.92996	2.07004	2.07072	0.03290
16	198.05483	1.94517	1.94573	0.02890
17	198.06490	1.93510	1.93574	0.03299
σ	**197.97983**	**2.02017**	**2.02069**	**0.02865**

The results presented in table 7.3 only verify the precision of the clearance measurement as a result of peak detection. Even though the accuracy is high, the results for orbit detection can shift towards misleading paths, as is displayed in figure 7.13. The signal is simulated with the following setup and with various settings for blade length:

- Blade Count: 17
- Offset: [0 0] mm
- Eccentricity: 1.5 mm
- Radius Casing: 250 mm
- Radius Shaft: 50 mm
- Blade Base Length: 198 mm
- $\triangle\theta$: 5000

While the orbit at a blade length variation of ± 0.1 mm (red) is still representable of the orbit with constant blade lengths (blue), the path of the shaft for a variation of ± 0.5 mm already yields deformed results (yellow). As mentioned above, this is not due to poor accuracy on the side of peak detection but rather explained by the fact that clearance information alone is used to detect the shaft center location. Since the clearance is no longer only a function of the shaft's dynamics, but is as well influenced by the magnitude of individual blade length variation, the effects on accuracy are apparent. This systematic error could be addressed by implementing a sub-routine within LabView that corrects the clearance data by the length offset from the mean blade length for each blade.

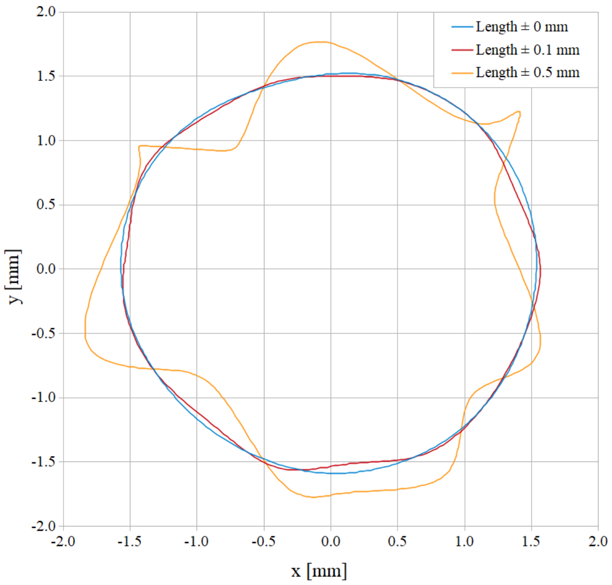


Figure 7.13: LabView results for different blade length imperfections (eccentricity = 1.5 mm)

Note, that even though the present orbit for blade length ± 0.5 mm (yellow) resembles the chaotic orbit shown in section 7.1 in figure 7.6 (b) for undersampled signals, the results are stationary in the present case. This is due to the fact that the blade length variation is constant throughout the simulation process and so is the resulting signal. This means, that the approximated min. rotor-casing clearance is equally steady. Unfortunately, this does not make the location and magnitude correct. The problem remains that the circle inscribed into the rotor's circumference by the means of three arbitrary clearance signals assumes perfect circularity, which is not given due to the variation in blade length and unpredictable error's are to be expected.

7.4.2 Noise

Up to this point the signal only consisted of clean information of the simulated machine provided by the signal generation unit. In reality, unwanted signal components like deflections caused by fringing of field lines or noise are present and have to be removed. For now, only the effect of untreated noise shall be displayed. For this purpose, the Matlab routines are altered in a way that, after the blade signal calculation, a random value within the range of ± noise is added. The function 'rand()' is used to compute the altered signal according to

$$Signal_{output} = Signal_{blade} + rand(noise) - rand(noise) \tag{7.11}$$

This leads to the desired effect of accumulating different signal components on top of the initial signal. The amplitude of noise and the amplitude of the maximum initial signal results in the indicator SNR, defined by equation 3.6

$$SNR = \frac{max(Signal)}{Noise}$$

The goal of this section is to show the effects of different SNRs on the signal processing quality. For peak detection, the results are comparable to the blade length variation. The noise increases or decreases the peaks of the signal and therefor affects clearance results. Unfortunately, this time the effect is not systematic and can therefore not be accounted for by calibration. Note, that in order to be able to detect the peaks the 'width' setting of the peak detection routine has to be increased. This is done in order not to detect random noise spikes in amplitude as individual peaks and thereby interfering with the referencing of the blades.

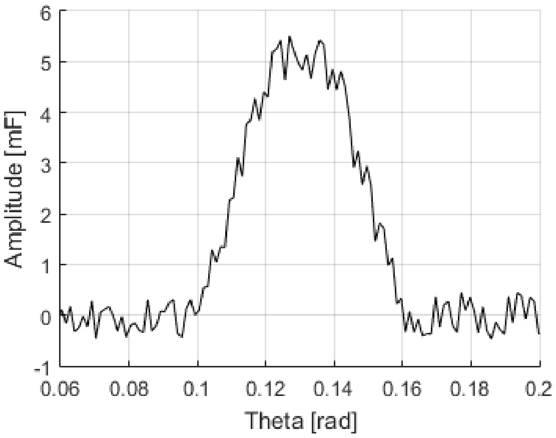


Figure 7.14: Clearance signal at a SNR of 10

The processing results for various simulated SNRs at a centered shaft with a constant clearance of 2 mm can be seen in table 7.4. The clearance for noise-free signals is represented by a SNR of ∞. Already at a rather high SNR of 10, as is displayed in figure 7.14, the clearance results are rendered useless due to the lack of remaining information about the true clearance. The average clearance information might be considered to be contained, but not much information can be drawn from this value, apart from approximating the stationary offset.

Table 7.4: Clearance results for various SNRs at a centered shaft with ’width’ = 30

SNR	Min. Clearance [mm]	Avg. Clearance [mm]	Max. Clearance [mm]
∞	1.99981	1.99981	1.99981
10	1.75294	1.91148	2.43119
7.5	1.70402	1.92770	2.50224
5	1.58508	1.93731	2.54716

Using the results from noise-heavy signals to calculate the shaft orbit leads to similar results as the blade length variation study showed. Again, the problem being that in the current case of random signal variation, no systematic corrections can be made to correct the offset and therefore unfiltered signals prove to be impossible to use accurately for direct calculation of the orbit, as can be taken from figure 7.15. Consequently, this uncertainty in orbit prediction leads to alternating and inaccurate information concerning the location of the smallest available clearance margin.

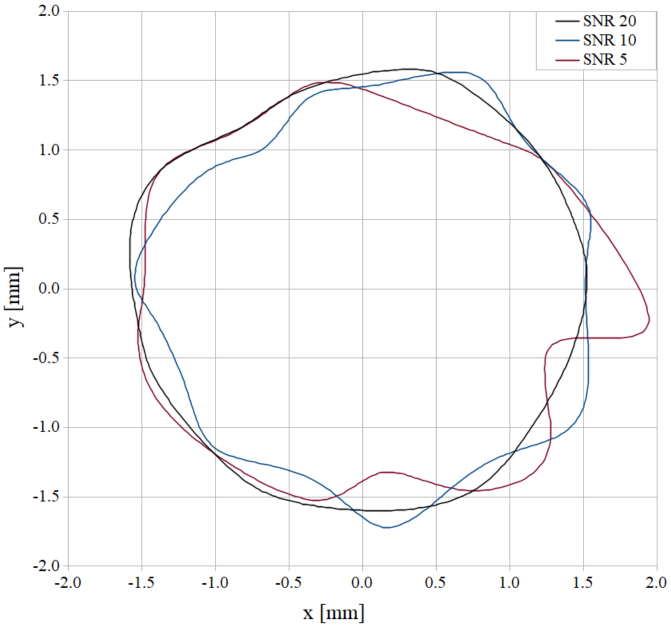


Figure 7.15: LabView results for different SNRs (eccentricity = 1.5 mm)

7.5 LabView Settings

Since the last section dealt with decreasing signal quality, this chapter focuses on the possible improvements based on settings available within the processing routines in LabView. Until now only the parameter 'width' has been changed for the peak detection sub-routine, in order to be able to detect blade peaks in noise-heavy signals. The possible signal improvements based on the curve-fitting-induced smoothing, mentioned previously in section 6.1, shall be analyzed. Apart from that, the already mentioned low-pass filter is available for the user to improve the results. When processing signals with high noise levels, the options range from setting a low cut-off frequency at the low-pass filter, effectively smoothing the curve so that peak detection can be performed very accurately even at low 'width', or initializing a high cut-off frequency, that leads to noise being left over but requires a higher 'width' setting. This tweaking process is highly situational and the effects on performance are outlined in the following pages.

7.5.1 Peak Detection

As has been stated before, LabView offers a built in sub-routine to handle peak detection that follows the described curve fitting methods in order to obtain peak amplitude and location. The algorithm fits second order polynomials into groups of 'width' data points and checks for maxima and minima by differentiating twice. If a found maximum exceeds a predefined threshold value, it is returned as peak, together with the index within the array. To obtain the most accurate result, 'width' has to be set to the lowest possible value, but as high as necessary. The lower limit is related to the amount of noise. 'Width' has to exceed the value, for which the detection algorithm reacts to local amplitude spikes caused by noise. On the other side, increasing 'width' too far leads to underestimating peaks or even missing them. Therefore, as a rule of thumb, 'width' should not be greater than half the width of a blade peak.

In order to showcase the effects this isolated setting has on the quality of the result, table 7.5 shows results of a 'width' variation at a centered shaft with a simulated clearance of 2 mm. A SNR of ten is compared to a noise-free signal. The minimum value for 'width' of 25 is due to additional peaks being detected when decreasing the value any further. By increasing 'width' step by step, no real tendency can be observed apart from the constant loss of accuracy.

Table 7.5: Clearance results at altering 'width' settings at a centered shaft

SNR	Width	Min. Clearance [mm]	Avg. Clearance [mm]	Max. Clearance [mm]
∞	3	1.99981	1.99981	1.99981
10	25 (min)	1.74792	1.89707	2.06307
10	26	1.73574	1.87988	2.07977
10	27	1.71306	1.88662	2.08541
10	28	1.75609	1.89804	1.98934
10	29	1.79387	1.88216	2.03467
10	30	1.67532	1.86640	1.98421
10	32	1.68985	1.85316	2.00770
10	35	1.76318	1.89642	2.04893
10	37	1.83046	1.89987	2.00306
10	40	1.8339	1.93639	2.09143

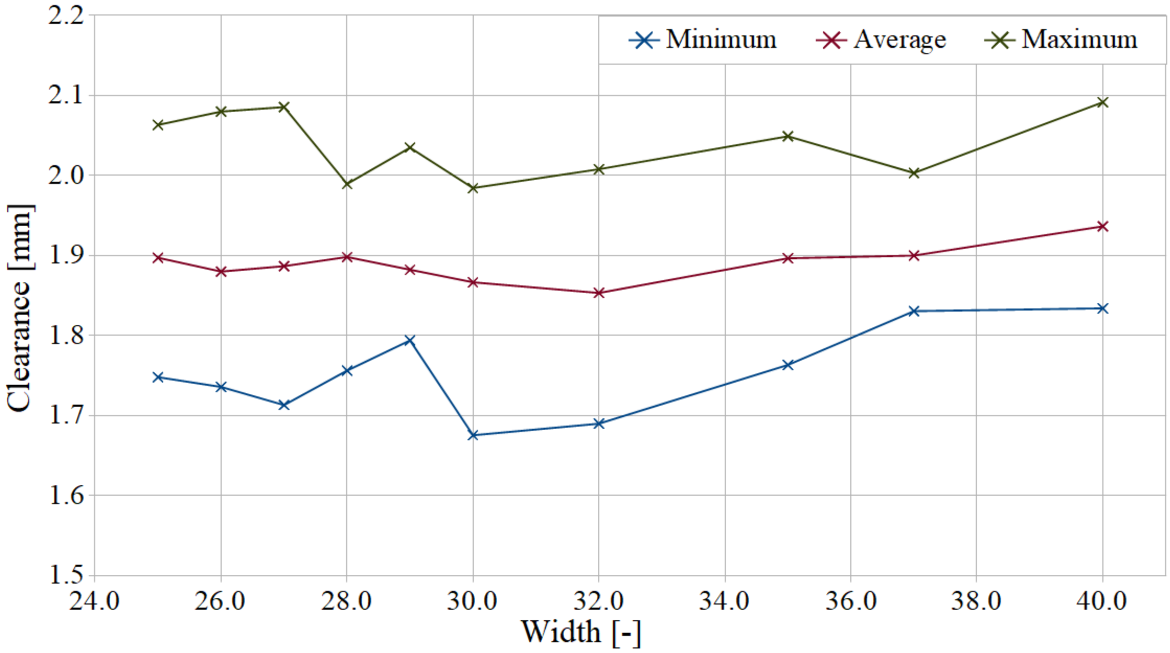


Figure 7.16: Minimum, average and maximum clearance detected at a SNR of 10 with various 'width' settings (centered shaft)

Figure 7.16 shows the same data as table 7.5 did. The increase and decrease in detected clearance with differences of around 0.1 mm between two consecutive 'width'-steps leads to the conclusion that smoothing noise with the curve fitting method proves to be inaccurate and can not be considered a valuable method to process noise-heavy data. As a result of these alternating clearance results, the resulting orbits resemble the orbits shown in section 7.4.2 dealing with untreated noise.

7.5.2 Lowpass Filtering

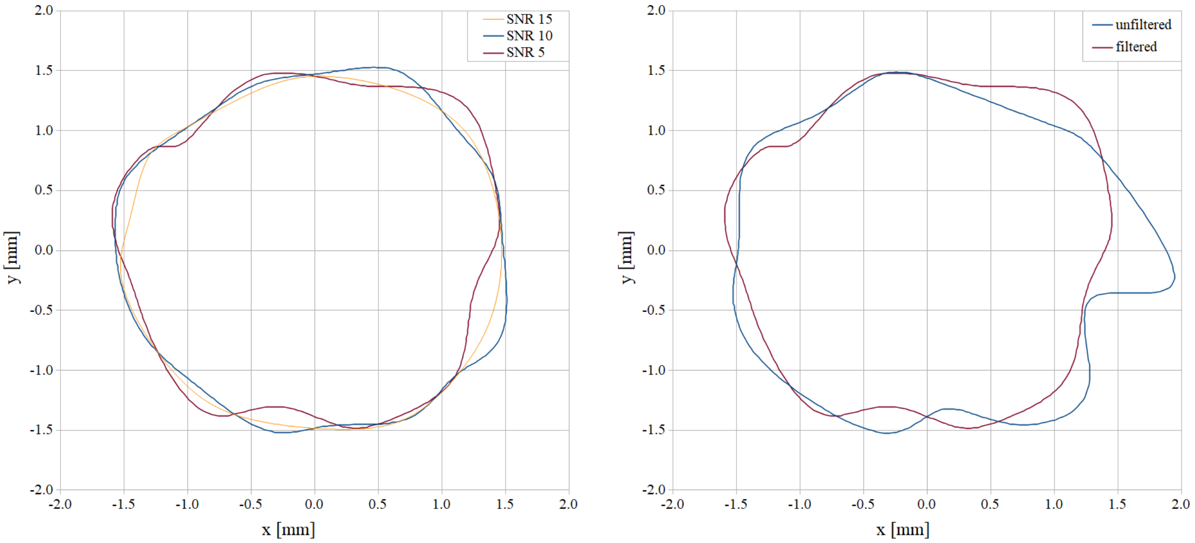
Due to the fact that 'width' has to be increased drastically to values of 25 and greater - even at moderate noise - in order to prevent additional peaks being detected, the necessity of preprocessing the data arises. An effective method to achieve this is forward-backwards low-pass filtering the data, as has been introduced in section 6.1. At cut-off frequencies f_c much higher than the blade-passing frequency, only the high frequency components are suppressed, while the blade peak amplitude does not suffer excessive reduction in amplitude. After that the peak detection sub-routine can be applied with a low 'width' setting, leading to more stationary results.

The effect of filtering on the detected clearance at a centered shaft with a constant clearance of 2 mm is an increase in measured clearance. The values range from a minimum of 2.20677 mm to a maximum of 2.31332 mm (average 2.23660 mm), which can be explained by the loss of magnitude of the capacitance peak due to the low-pass effect described in section 4. This reduction of amplitude leads to an increase in distance according to equation 5.3

$$Clearance \approx \frac{1}{Signal}$$

As can be seen in figure 7.17 (a), filtering does not greatly improve the accuracy of the approximated shaft orbit. The influence of the noise on the peak amplitude still leads to fluctuating results. Only when comparing the filtered result with a 'width' of 3 (red) directly to the unfiltered result with a 'width' of 25 (blue) in figure 7.17 (b)(SNR=5), the improvements become apparent.

Generally speaking, low pass filtering seems to behave more stable than curve fitting. Especially when recalling the effects shown previously concerning the variance in results for clearance due to an incremental change of 'width' (compare figure 7.16), filtering as a primary measure proves to achieves better results. Note, that since the damping of the signal due to low-pass filtering is systematic, calibration can achieve an increase in accuracy. This means that for decreasing SNRs the overall accuracy improves.



(a) Filtered signals (eccentricity = 1.5 mm & 'width' = 3)

(b) Comparison of filtered ('width' = 3) and unfiltered ('width' = 25) signal (SNR = 5)

Figure 7.17: Improvements in orbit results due to filtering at different SNR

7.5.3 Averaging

The reason for the remaining misshaped orbit is the fact that, even when filtering the signal components of the noise, some change in blade peak amplitude remains. Hence, the accuracy of clearance measurement suffers a great deal as soon as SNR reaches significant values, as has been discussed above. The nature of the noise applied to the signal in this theses is based on equation 7.11 from before:

$$Signal_{output} = Signal_{blade} + rand(noise) - rand(noise)$$

Note, that the mean value of ± noise for a high enough amount of sample strives towards zero. This fact can be used to improve the results.

The method implemented to make use of this statement has been mentioned before, while describing the post-processing loops in section 6.2. Instead of directly computing the orbit, a multidimensional blade-by-blade array is set up and data is recorded for multiple revolutions of the rotor. The underlying algorithm then takes each row, containing various clearance amplitudes for one individual blade and averages the clearances. This is done for all 'amount-of-blade' rows of the blade-by-blade array, after which the circle is inscribed into the corresponding sensor data sets. The result is a smoothed orbit that fluctuates less due to random noise. Two important things have to be respected:

1. If the noise is not distributed according to the pattern described before, the method loses accuracy. Fortunately, in case of noise being systematically induced, it becomes an error that can be addressed by averaging.
2. Due to reducing a set of data to a mean value, the shaft's orbit has to be as stationary as possible. Otherwise changing states of operation can be blurred and critical information might be missed.

As for the performance of the averaging method, figure 7.18 shows the effects on clearance results for the centered shaft with a predefined clearance of 2 mm.

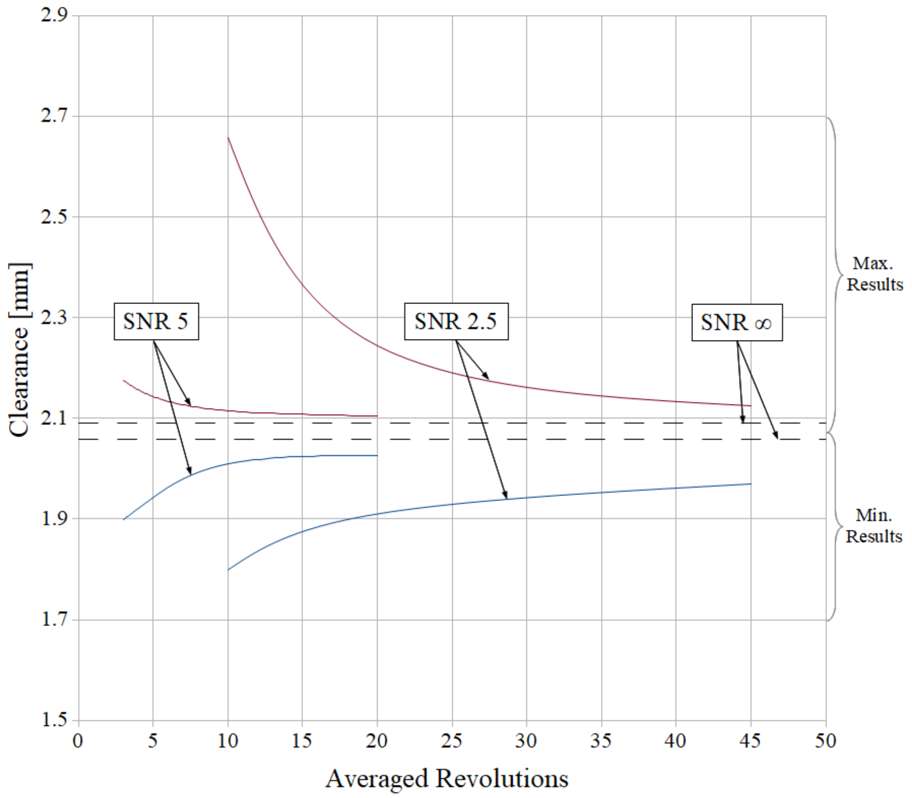
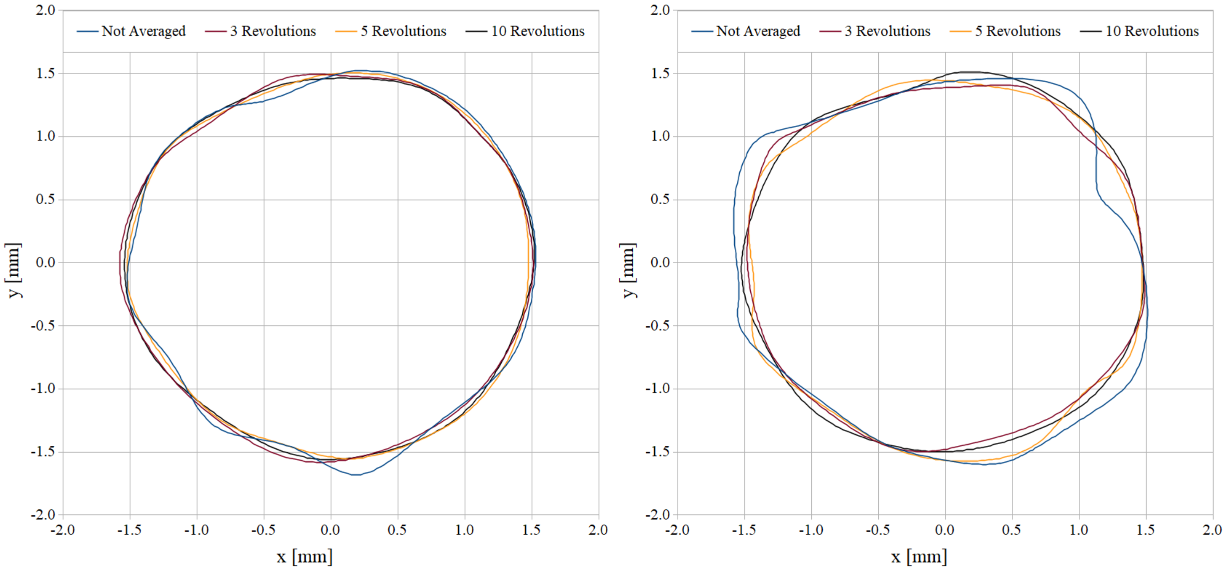


Figure 7.18: Effects of averaging on clearance results of two different SNR compared to a noise-free signal (always filtered)

The upper half of the figure shows the convergence of the maximum peak values detected towards the ideal, noise-free signal and the lower half shows the convergence towards the ideal minimum result. The ideal results with a SNR of ∞ are displayed as dashed lines. As

is apparent, the lower the SNR gets, the more rotations have to be used until the desired precision is achieved with averaging. For a SNR of 5 no real change in accuracy is achieved by increasing the revolutions to average above 10, while choosing a value below 30 revolutions at a SNR of 2.5 does not lead to improved results. Note the offset of both ideal results from a clearance of 2 mm upwards. This effect has been explained in section 7.5.2, where the loss of capacitance amplitude and resulting positive shift of distance for low-pass filtered signals has been mentioned. This offset can be addressed by calibration.
The same tendencies can be seen in figure 7.19 at the approximation of the shafts orbit. While a SNR of 10 (a) does not require averaging of more than three revolutions for better accuracy, at a SNR of 5 (b) the improvements for increased revolutions of up to 10 are apparent.



(a) SNR = 10

(b) SNR = 5

Figure 7.19: LabView results for two different SNRs with variable amounts of revolutions used for averaging (eccentricity = 1.5 mm & always low-pass filtered)

8 Conclusion

Finally, the aftermath of the statements made in chapter 7 can be summed up and the thesis can be concluded with an outlook over the range of possibilities given by the presented methods and routines.

8.1 Best Performance

Initially, the connections found in the previous chapter are highlighted once more before proceeding to apply all of them in one generalized test:

1. Calibration is a very situational operation that requires the entire system to be set up before being able to successfully link signal to physical value.
2. Spatial resolution influences the accuracy drastically and randomly. That is why oversampling in the domain of clearance measurement is a necessity and leads to an excessive setting for sample rate. As a rule of thumb, sampling frequencies of 30 or more times the blade-passing frequency are recommended.
3. The prediction of the smallest clearance between rotor and casing using approximated orbit information relies greatly on the accuracy of the latter. The magnitude of this error has been linked to the precision of the shaft's center position and the radius of the circle inscribed into the rotor's circumference. In certain cases the induced signal imperfections even lead to chaotic orbits and as a result of that to random location and magnitude of the minimal clearance. Therefore it can be stated, that every improvement made calculating the shaft's center directly applies to the accuracy of this information.
4. The positioning of the sensors on the circumference of the casing has to be as even as possible. Adjusting the sensors to the blades - in accordance with statements made in section 6.2.2 - can improve the accuracy of the outcome. The degree of the effect is linked to the number of blades in the blade row, where a higher count leads to less impact on accuracy, while a lower count shows deformation of the shaft's orbit. For common turbomachinery it can be stated, that sensor alignment is not a necessity since there is no clear improvement of processing accuracy. Note, that using more than

three sensors, computing the shaft's location for various constellations and averaging the data could potentially improve the result.

5. Variations in blade length influence the signal like noise. The positive aspect is the systematic nature of the deviation, that allows this effect to be countered by defining a correctional value for each individual blade. Note, that this is not a necessity for small variations (« 0.1 mm), since the impact on accuracy is neglectable.
6. Noise has a worsening effect on accuracy without exceptions. Unlike at the systematic blade length error, this type of error can not be addressed directly due to its random nature.
7. The setting 'width', that represents the amount of consecutive data points used for curve fitting within the peak detection sub-routine, is not suited to improve results on signals with a high Signal-to-Noise-Ratio. Its effect is rather unpredictable and very situational, which is why this value should be as close as possible to its minimum of three.
8. The better way to address noise-heavy signals is by low-pass filtering the data and then running the peak detection at low 'width' values. The dampening effect of the low-pass filter has to be addressed by a dedicated calibration process in order to ensure high accuracy.
9. While the filtering mainly serves the purpose of reducing the 'width' setting and smoothing the signal, the noise still changes the amplitudes of the peaks significantly. The presented method for removing noise effects at stationary machine conditions is averaging over multiple revolutions. The averaged results show great improvement in accuracy since they manage to even out fluctuations in peak amplitudes caused by the noise.

In accordance with these statements, the following simulation aims to display the best performance at various SNRs. The machine model is simulated with following parameters for simple blade configuration:

- Blade Count: 79
- Radius Casing: 250 mm
- Radius Shaft: 50 mm
- Offset = [0,0] mm
- Blade Length: 198 mm
- $\triangle\theta$: 5000
- Peak Detection 'Width': 3
- Low-pass Filtering

In chapter 7 individual calibration has been avoided due to the desire to have uniform testing conditions. Here, the goal is not to showcase effects of individual parameters, but to reach the highest accuracy possible. Therefore a new calibration curve for the filtered signals is acquired. This is done by simulating a centered shaft with known clearance within Matlab without any noise and, instead of using a maximum signal amplitude as before, accumulating clearance values for an entire revolution of the rotor. Since the shaft is centered and there is no noise present, the clearances should be equal at each blade and the results can be averaged to acquire one value within LabView. This is repeated for clearances from zero to five millimeters, leading to the desired calibration curve. Note, that the clearance signal at this point is highly influenced by the spatial resolution defined by $\triangle\theta$. This is due to the fact, that the composition of signal components changes, according to how much oversampling is done, which is why the low-pass filter produces different results for every sample rate. In case of changing circumferential velocity, for example due to the machine spinning up, this problem has to be addressed by more elaborate and transient filter settings to conserve the calibration.

First, the performance at clearance detection is evaluated by simulating one revolution of a centered shaft with a clearance of 2 mm at different noise-settings. $SNR = \infty$ is the case of no noise being present, a high number indicates low levels of noise and a low number implies the contrary. The results for detected clearances after one revolution of the rotor are displayed in table 8.1. In section 7.5.2 the result of filtering a noise free signal without adjusted calibration has been shown to result in a minimum of 2.20677 mm and a maximum of 2.31332 mm (average 2.23660 mm) of clearance detected. The overestimation of the clearance has been explained by loss of the signal amplitude due to the low-pass effects.

Table 8.1: Optimized results for different SNRs with centered shaft

SNR	Min Clearance [mm]	Avg Clearance [mm]	Max Clearance [mm]
∞	2.00251	2.00301	2.0035
10	1.82839	1.99296	2.1494
7.5	1.80695	1.98756	2.17432
5	1.6719	2.00366	2.3064
5 (avg)	1.91382	1.99451	2.09068

In present case with adjusted calibration, the results for a noise free signal (SNR=∞)

show increased accuracy, proving the statements made throughout the previous chapter concerning improvements on clearance detection due to better calibration. Comparing the results above to the ones in table 7.4 of section 7.4.2 dealing with the impact of untreated noise, the improvements for lower SNRs become obvious. Generally speaking, the deviation from the ideal, noise-free signal has been reduced by 0.2 mm by low-pass filtering and calibrating accordingly. The accuracy can be further improved by the mentioned averaging of clearance data of multiple revolutions. Accumulating data for 20 revolutions of the rotor and computing the mean clearance for each blade results in a minimum of 1.91382 mm and a maximum of 2.09068 mm (average 1.99451) at a SNR=5, displayed in table 8.1.

Furthermore, the impact on orbit prediction can be highlighted. In order to do so, an eccentricity of 1.5 mm is set to be able to compare the result to results shown in chapter 7. Additionally, the processing unit is set up to average over 60 revolutions of the rotor. When assuming a rotational speed of 6000 RPM, the simulated signal represent one second worth of data of that machine. At a spatial stepping rate of $\triangle\theta$ of 5000 this leads to a total number of $5000 \cdot 60 = 300000$ samples[1] being acquired and processed. The result for different SNRs at this setup can be seen in figure 8.1. Compared to the various results for the same orbit prediction in chapter 7, the improvements are apparent. At a SNR of 10 the circularity of the orbit is near perfect and the eccentricity of 1.5 mm is captured accurately. The reduction of quality for decreasing SNR is still present, nevertheless the effect has been reduced drastically.

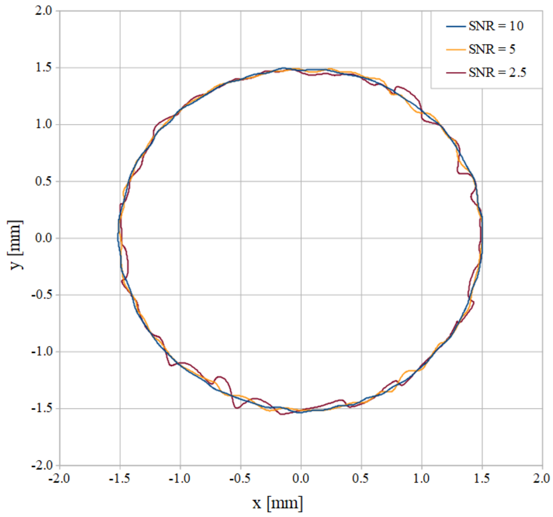


Figure 8.1: LabView results for various SNRs with optimized routines (eccentricity = 1.5 mm & 79 blades & 60 averaged rotations)

[1] The requirement for the Data-Aquisition device would be a sampling frequency of 300000 Hz.

8.2 Summary and Outlook

Throughout this thesis the model-based development of a radial rotor-casing clearance monitoring software for axial, thermal turbomachinery has been thoroughly described. Initially, the underlying fields of machinery and measurement were highlighted. This was used as a foundation for the reader to comprehend the design choices made by the author concerning the selected sensing principle and modeling processes. Subsequently the software development was described in detail in order to provide information for the successor of the project, responsible for the acquisition and the installation of the hardware. The model-based approach allowed for precise studying of individual aspects of the regimes, leading to clear statements concerning influential parameters.

The evaluation of these influences showed that the machine's parameters can be derived accurately by the developed software, as long as the stated premises are met. This was shown by studying various shaft movements with the help of the parameters eccentricity and offset at idealized signal quality. In any case, these values were derived from the signal with high precision in form of clearance and orbit data. Moreover the effect of blade length imperfections was discussed. It has been mentioned that, while influencing the orbit prediction comparable to noise, easy offset corrections can be made, also rendering this geometric factor little influential on the overall quality. Unlike the machine, all components of the measurement chain that were modeled have proven to have a huge impact on the performance. Throughout the concluding parameter variation the effects of different sensor setups and an answer to the question whether dedicated sensor alignment improves accuracy has been shown. Furthermore, multiple requirements for a measurement chain in turbomachinery application were derived. Apart from the selection of a viable sensing principle, providing an overview over possible cable types and showcasing basic circuit methods, the sample rates needed in this specific case could be determined and proven. Additionally, by choosing array-based data processing, the clearance monitoring system was designed to be able to deal with the arriving data of common data-acquisition devices.

Due to focusing on a wide range of different signal types, continuous and pulsing, the developed routines are designed to be applicable very generalized, which is enhanced even further by their expandable nature. Within the range of tested parameters and signal types, measurements of below 0.1 mm seem to be realistic, solely dependent on the quality of the input signal. This dependency also proves to be the biggest problem. The pure model-based approach lacks the testing required for anything else but the prototype stage of the created processing routines. The performance has been discussed extensively and has been verified as precisely as possible, but the range of possible input signals is nearly endless and the information contained is often influenced in unpredictable ways. Therefore the routines are very situational, leaving a lot of signal preprocessing work to be done by the designer of the real-world measurement chain.

Apart from the adaptation that is required to ensure the presented signal quality, further refinements of the routines can allow for additional monitoring precision and information. Generally speaking, two future development steps can be outlined:

1. Refining the mathematical principles and using more sensors.
2. Adapting the routines for other purposes.

The first step includes the extension of the shaft localization algorithms for shapes other than circles with perfect circularity. In the present case, three sensors sufficed in order to obtain the shaft's location within the casing. Nevertheless, if the shape of the inscribed line shall be anything other than a circle, the amount of reference points has to be altered. In case of an ellipse being fit into the tips, five sensors are needed, which requires for an adaptation of the underlying mathematical principle. This could potentially increase processing accuracy in case of imperfections of blade lengths or uneven deformations of the casing. Furthermore, additional sensors could as well be used for improving the accuracy of the present routines. The location of the shaft could be computed for every constellation of sensors and the results could be averaged, which could possibly lead to improvements in accuracy, especially at low signal quality. The effect each of the proposed alterations has on the accuracy of the measurement needs to be evaluated individually. The second step concerns the adaptation of the routines for other purposes than clearance measurement. At the present case, the focus was solely on clearance detection and the results that could be derived from this information. With the help of a few additional models regarding elastic blade behavior and processing of this information, the introduced routines can easily be used for developing a monitoring software for tip timing purposes.

Nomenclature

Abbreviations

AD Analog-to-Digital

BPF Blade-Passing-Frequency

DAQ Data-Aquisition

RPM Revolutions per Minute

SBR Sensor-to-Blade-Width-Ratio

SNR Signal-to-Noise-Ratio

TCP Transmission Control Protocol

Symbols

A	m^2	Area
c	$\frac{N}{m}$	Spring Stiffness
C	F	Capacitance
d	m	distance
e	mm	eccentricity
E	$\frac{N}{m^2}$	E-Modul
f	Hz	frequency
f_c	Hz	cutoff frequency
f_{sample}	Hz	sample frequency
F	N	Force
F’	N	Force per Unit Span

F_C	N	Centrifugal Force
F_S	N	Shaft's Spring Force
G	-	Transfer Function
I	A	Current
l, L	m	Length
L_B	mm	Blade Length
j	-	Imaginary Unit
m	kg	Mass
N(x)	N	Normal Force
o	mm	Stationary Offset of the Rotor from the Casing's Center
Q	C	Charge
R	Ω	Resistance
R_{Casing}	mm	Radius Casing
R_{Shaft}	mm	Radius Shaft
R_{Shroud}	mm	Radius Shroud
s	m	spatial Coordinate
T	°K	Temperature
theta	rad	Incremental Angle
u(x)	m	Displacement
U_{in}	V	Input Voltage
U_{out}	V	Output Voltage
U_∞	$\frac{m}{s}$	Velocity of the Potential Fluid Flow
v	$\frac{m}{s}$	Velocity
V	V	Voltage
W_B	mm	Blade Width
W_S	mm	Sensor Area Width
x	m	Horizontal Displacement Amplitude
X_C	Ω	Resistance of a Capacitor
y	m	Vertical Displacement Amplitude
z	m	Distance between Shaft's Centerline and Center of Mass

Greek

α_{th}	deg K^{-1}	thermal expansion coefficient
α	rad	Blade Distribution Angle
θ	rad	Incremental Angle
$\triangle\theta$	rad	Incremental Angle Step
δ	deg	Sensor Placement Angle
ϵ_0	$\frac{F}{m}$	Permittivity of Vacuum $\approx 8.85 \cdot 10^{-12}$
ϵ_r	$\frac{F}{m}$	relative Permittivity of a non-conductive Area
Γ	-	Circulation
ρ	$\frac{kg}{m^3}$	Density
σ	-	Mean Value
ω	$\frac{1}{s}$	Angular Velocity
ω_0	$\frac{1}{s}$	Natural Angular Velocity
ω_c	$\frac{1}{s}$	Angular Cutoff Frequency
Ω_s	$\frac{1}{s}$	Angular Velocity of a Shaft

Bibliography

[Aerossurance Website] Aerossurance Website: *Micro FOD: Cessna 208B Grand Caravan Engine Failure & Forced Landing.* http://aerossurance.com/safety-management/micro-fod/, Accessed: 2018-09-12

[AmBrSoft Website] AmBrSoft Website: *Circle Defined by 3 Arbitrary Points.* http://ambrsoft.com/TrigoCalc/Circle3D.htm, Accessed: 2018-07-07

[Ameri et al. 1998] Ameri, A A. ; Corporation, A Y T. ; Park, Brook ; Steinthorsson, E ; Rigby, David L.: *Effect of Squealer Tip on Rotor Heat Transfer and Efficiency.* In: Journal of Turbomachinery 120 (1998). http://dx.doi.org/10.1115/1.2841786. – DOI 10.1115/1.2841786

[Azad et al. 2002] Azad, Salam ; Han, Je-Chin ; Bunker, Ronald S. ; Lee, Pang C.: *Effect of Squealer Geometry Arrangement on a Gas Turbine Blade Tip Heat Transfer.* In: Journal of Heat Transfer 124 (2002). http://dx.doi.org/10.1115/1.1471523. – DOI 10.1115/1.1471523

[Böge 2013] Böge, Alfred: *Handbuch Machinenbau.* 21. Editio. Springer Vieweg, 2013

[Denton 1993] Denton, J. D.: *Loss Mechanisms in Turbomachines.* In: Journal of Turbomachinery 115 (1993). http://dx.doi.org/10.1115/1.2929299. – DOI 10.1115/1.2929299

[Earl and Ramendra 1995] Earl, Logan J. ; Ramendra, Roy: *Handbook of Turbomachinery.* 2. Edition. Marcel Dekker, Inc., 1995

[Fabian et al. 2005] Fabian, Tibor ; Prinz, Friedrich B. ; Brasseur, Georg: *Capacitive Sensor for Active Tip Clearance Control in a Palm-Sized Gas Turbine Generator.* In: IEEE Transactions on Instrumentation and Measurement 54 (2005), Nr. 3. http://dx.doi.org/10.1109/TIM.2005.847233. – DOI 10.1109/TIM.2005.847233

[Flotow and Drumm] Flotow, Andreas ; Drumm, Michael J.: *Blade-Tip Monitoring with Through-the-Case Eddy Current Sensors.* http://archives.sensorsmag.com/articles/0604/28/main.shtml, Accessed: 2018-09-03

[Haase and Haase 2013] Haase, Wayne C. ; Haase, Zachary S.: *High-Speed, Capacitance-Based Tip Clearance Sensing*. In: IEEE Aerospace Conference Proceedings (2013). http://dx.doi.org/10.1109/AERO.2013.6496903. – DOI 10.1109/AERO.2013.6496903

[Jia and Zhang 2011] Jia, Binghui ; Zhang, Xiaodong: *An Optical Fiber Blade Tip Clearance Sensor for Active Clearance Control Applications*. In: Procedia Engineering 15 (2011). http://dx.doi.org/10.1016/j.proeng.2011.08.182. – DOI 10.1016/j.proeng.2011.08.182

[Khan Academy] Khan Academy: *Green's Theorem*. https://www.khanacademy.org/math/multivariable-calculus/greens-theorem-and-stokes-theorem, Accessed: 2018-07-07

[Marn 2014] Marn, Andreas: *Skriptum zur Vorlesung Rotordynamik*. 2014

[Miller 1992] Miller, S.J.: *The Method of Least Squares and Signal Analysis*. 1992

[National Instruments Website] National Instruments Website: *LabView Peak Detection*. http://www.ni.com/white-paper/3770/en/, Accessed: 2018-03-08

[Quora Website] Quora Website: *Why do airplanes use an axial flow jet engine instead of a more compact centrifugal jet engine? Helicopters use them and they work fine, so why not on a fixed wing aircraft?* https://www.quora.com/Why-do-airplanes-use-an-axial-flow-jet-engine-instead-of-a-more-compact-centrifugal-jet/engine-Helicopters-use-them-and-they-work-fine-so-why-not-on-a-fixed-wing-aircraft, Accessed: 2018-09-22

[Schwarz] Schwarz, Douglas M.: *Fast and Robust Curve Intersections*. https://de.mathworks.com/matlabcentral/fileexchange/11837-fast-and-robust-curve-intersections?focused=7711886{&}tab=function, Accessed: 2018-07-07

[Sheard et al. 1997] Sheard, A.G. ; O'Donell, S.G. ; Stringfellow, J.F.: *High Temperature Proximity Measurement in Aero and Industrial Turbomachinery*. In: Journal of Engineering for Gas Turbines and Power (1997). http://dx.doi.org/10.1115/97-GT-198. – DOI 10.1115/97–GT–198

[Szczepanik et al. 2012] Szczepanik, Ryszard ; Przysowa, Radosław ; Spychała, Jarosław: *Application of Blade-Tip Sensors to Blade-Vibration Monitoring in Gas Turbines*. In: Thermal Power Plants (2012). http://dx.doi.org/10.5772/29550. – DOI 10.5772/29550

Appendix

Intersections Algorithm by Schwarz

```matlab
function [x0,y0,iout,jout] = intersections(x1,y1,x2,y2,robust)
%INTERSECTIONS Intersections of curves.
%   Computes the (x,y) locations where two curves intersect.  The curves
%   can be broken with NaNs or have vertical segments.
%
% Example:
%   [X0,Y0] = intersections(X1,Y1,X2,Y2,ROBUST);
%
% where X1 and Y1 are equal-length vectors of at least two points and
% represent curve 1.  Similarly, X2 and Y2 represent curve 2.
% X0 and Y0 are column vectors containing the points at which the two
% curves intersect.
%
% ROBUST (optional) set to 1 or true means to use a slight variation of the
% algorithm that might return duplicates of some intersection points, and
% then remove those duplicates.  The default is true, but since the
% algorithm is slightly slower you can set it to false if you know that
% your curves don't intersect at any segment boundaries.  Also, the robust
% version properly handles parallel and overlapping segments.
%
% The algorithm can return two additional vectors that indicate which
% segment pairs contain intersections and where they are:
%
%   [X0,Y0,I,J] = intersections(X1,Y1,X2,Y2,ROBUST);
%
% For each element of the vector I, I(k) = (segment number of (X1,Y1)) +
% (how far along this segment the intersection is).  For example, if I(k) =
% 45.25 then the intersection lies a quarter of the way between the line
% segment connecting (X1(45),Y1(45)) and (X1(46),Y1(46)).  Similarly for
% the vector J and the segments in (X2,Y2).
%
% You can also get intersections of a curve with itself.  Simply pass in
% only one curve, i.e.,
%
%   [X0,Y0] = intersections(X1,Y1,ROBUST);
%
% where, as before, ROBUST is optional.

% Version: 2.0, 25 May 2017
% Author:  Douglas M. Schwarz
% Email:   dmschwarz=ieee*org, dmschwarz=urgrad*rochester*edu
% Real_email = regexprep(Email,{'=','*'},{'@','.'})


% Theory of operation:
%
% Given two line segments, L1 and L2,
%
%   L1 endpoints:  (x1(1),y1(1)) and (x1(2),y1(2))
%   L2 endpoints:  (x2(1),y2(1)) and (x2(2),y2(2))
%
% we can write four equations with four unknowns and then solve them.  The
% four unknowns are t1, t2, x0 and y0, where (x0,y0) is the intersection of
% L1 and L2, t1 is the distance from the starting point of L1 to the
% intersection relative to the length of L1 and t2 is the distance from the
% starting point of L2 to the intersection relative to the length of L2.
%
% So, the four equations are
%
%    (x1(2) - x1(1))*t1 = x0 - x1(1)
%    (x2(2) - x2(1))*t2 = x0 - x2(1)
%    (y1(2) - y1(1))*t1 = y0 - y1(1)
%    (y2(2) - y2(1))*t2 = y0 - y2(1)
%
% Rearranging and writing in matrix form,
%
%  [x1(2)-x1(1)       0       -1   0;      [t1;      [-x1(1);
%        0       x2(2)-x2(1)  -1   0;   *   t2;   =   -x2(1);
%   y1(2)-y1(1)       0        0  -1;       x0;       -y1(1);
%        0       y2(2)-y2(1)   0  -1]       y0]       -y2(1)]
%
% Let's call that A*T = B.  We can solve for T with T = A\B.
%
% Once we have our solution we just have to look at t1 and t2 to determine
% whether L1 and L2 intersect.  If 0 <= t1 < 1 and 0 <= t2 < 1 then the two
% line segments cross and we can include (x0,y0) in the output.
%
% In principle, we have to perform this computation on every pair of line
% segments in the input data.  This can be quite a large number of pairs so
% we will reduce it by doing a simple preliminary check to eliminate line
% segment pairs that could not possibly cross.  The check is to look at the
% smallest enclosing rectangles (with sides parallel to the axes) for each
% line segment pair and see if they overlap.  If they do then we have to
% compute t1 and t2 (via the A\B computation) to see if the line segments
% cross, but if they don't then the line segments cannot cross.  In a
% typical application, this technique will eliminate most of the potential
% line segment pairs.


% Input checks.
if verLessThan('matlab','7.13')
    error(nargchk(2,5,nargin)) %#ok<NCHKN>
else
    narginchk(2,5)
end

% Adjustments based on number of arguments.
switch nargin
    case 2
        robust = true;
        x2 = x1;
        y2 = y1;
        self_intersect = true;
    case 3
        robust = x2;
        x2 = x1;
        y2 = y1;
        self_intersect = true;
    case 4
        robust = true;
        self_intersect = false;
    case 5
        self_intersect = false;
end

% x1 and y1 must be vectors with same number of points (at least 2).
if sum(size(x1) > 1) ~= 1 || sum(size(y1) > 1) ~= 1 || ...
        length(x1) ~= length(y1)
    error('X1 and Y1 must be equal-length vectors of at least 2 points.')
end
```

```
% x2 and y2 must be vectors with same number of points (at least 2).
if sum(size(x2) > 1) ~= 1 || sum(size(y2) > 1) ~= 1 || ...
        length(x2) ~= length(y2)
    error('X2 and Y2 must be equal-length vectors of at least 2 points.')
end


% Force all inputs to be column vectors.
x1 = x1(:);
y1 = y1(:);
x2 = x2(:);
y2 = y2(:);

% Compute number of line segments in each curve and some differences we'll
% need later.
n1 = length(x1) - 1;
n2 = length(x2) - 1;
xy1 = [x1 y1];
xy2 = [x2 y2];
dxy1 = diff(xy1);
dxy2 = diff(xy2);


% Determine the combinations of i and j where the rectangle enclosing the
% i'th line segment of curve 1 overlaps with the rectangle enclosing the
% j'th line segment of curve 2.

% Original method that works in old MATLAB versions, but is slower than
% using binary singleton expansion (explicit or implicit).
% [i,j] = find( ...
%   repmat(mvmin(x1),1,n2) <= repmat(mvmax(x2).',n1,1) & ...
%   repmat(mvmax(x1),1,n2) >= repmat(mvmin(x2).',n1,1) & ...
%   repmat(mvmin(y1),1,n2) <= repmat(mvmax(y2).',n1,1) & ...
%   repmat(mvmax(y1),1,n2) >= repmat(mvmin(y2).',n1,1));

% Select an algorithm based on MATLAB version and number of line
% segments in each curve.  We want to avoid forming large matrices for
% large numbers of line segments.  If the matrices are not too large,
% choose the best method available for the MATLAB version.
if n1 > 1000 || n2 > 1000 || verLessThan('matlab','7.4')
    % Determine which curve has the most line segments.
    if n1 >= n2
        % Curve 1 has more segments, loop over segments of curve 2.
        ijc = cell(1,n2);
        min_x1 = mvmin(x1);
        max_x1 = mvmax(x1);
        min_y1 = mvmin(y1);
        max_y1 = mvmax(y1);
        for k = 1:n2
            k1 = k + 1;
            ijc{k} = find( ...
                min_x1 <= max(x2(k),x2(k1)) & max_x1 >= min(x2(k),x2(k1)) & ...
                min_y1 <= max(y2(k),y2(k1)) & max_y1 >= min(y2(k),y2(k1)));
            ijc{k}(:,2) = k;
        end
        ij = vertcat(ijc{:});
        i = ij(:,1);
        j = ij(:,2);
    else
        % Curve 2 has more segments, loop over segments of curve 1.
        ijc = cell(1,n1);
        min_x2 = mvmin(x2);
        max_x2 = mvmax(x2);
        min_y2 = mvmin(y2);
        max_y2 = mvmax(y2);
        for k = 1:n1
            k1 = k + 1;
            ijc{k}(:,2) = find( ...
                min_x2 <= max(x1(k),x1(k1)) & max_x2 >= min(x1(k),x1(k1)) & ...
                min_y2 <= max(y1(k),y1(k1)) & max_y2 >= min(y1(k),y1(k1)));
            ijc{k}(:,1) = k;
        end
        ij = vertcat(ijc{:});
        i = ij(:,1);
        j = ij(:,2);
    end

elseif verLessThan('matlab','9.1')
    % Use bsxfun.
    [i,j] = find( ...
        bsxfun(@le,mvmin(x1),mvmax(x2).') & ...
        bsxfun(@ge,mvmax(x1),mvmin(x2).') & ...
        bsxfun(@le,mvmin(y1),mvmax(y2).') & ...
        bsxfun(@ge,mvmax(y1),mvmin(y2).'));

else
    % Use implicit expansion.
    [i,j] = find( ...
        mvmin(x1) <= mvmax(x2).' & mvmax(x1) >= mvmin(x2).' & ...
        mvmin(y1) <= mvmax(y2).' & mvmax(y1) >= mvmin(y2).');

end


% Find segments pairs which have at least one vertex = NaN and remove them.
% This line is a fast way of finding such segment pairs.  We take
% advantage of the fact that NaNs propagate through calculations, in
% particular subtraction (in the calculation of dxy1 and dxy2, which we
% need anyway) and addition.
% At the same time we can remove redundant combinations of i and j in the
% case of finding intersections of a line with itself.
if self_intersect
    remove = isnan(sum(dxy1(i,:) + dxy2(j,:),2)) | j <= i + 1;
else
    remove = isnan(sum(dxy1(i,:) + dxy2(j,:),2));
end
i(remove) = [];
j(remove) = [];

% Initialize matrices.  We'll put the T's and B's in matrices and use them
% one column at a time.  AA is a 3-D extension of A where we'll use one
% plane at a time.
n = length(i);
T = zeros(4,n);
AA = zeros(4,4,n);
AA([1 2],3,:) = -1;
AA([3 4],4,:) = -1;
AA([1 3],1,:) = dxy1(i,:).';
AA([2 4],2,:) = dxy2(j,:).';
B = -[x1(i) x2(j) y1(i) y2(j)].';
```

```
% Loop through possibilities.  Trap singularity warning and then use
% lastwarn to see if that plane of AA is near singular.  Process any such
% segment pairs to determine if they are colinear (overlap) or merely
% parallel.  That test consists of checking to see if one of the endpoints
% of the curve 2 segment lies on the curve 1 segment.  This is done by
% checking the cross product
%
%   (x1(2),y1(2)) - (x1(1),y1(1)) x (x2(2),y2(2)) - (x1(1),y1(1)).
%
% If this is close to zero then the segments overlap.

% If the robust option is false then we assume no two segment pairs are
% parallel and just go ahead and do the computation.  If A is ever singular
% a warning will appear.  This is faster and obviously you should use it
% only when you know you will never have overlapping or parallel segment
% pairs.

if robust
    overlap = false(n,1);
    warning_state = warning('off','MATLAB:singularMatrix');
    % Use try-catch to guarantee original warning state is restored.
    try
        lastwarn('')
        for k = 1:n
            T(:,k) = AA(:,:,k)\B(:,k);
            [unused,last_warn] = lastwarn; %#ok<ASGLU>
            lastwarn('')
            if strcmp(last_warn,'MATLAB:singularMatrix')
                % Force in_range(k) to be false.
                T(1,k) = NaN;
                % Determine if these segments overlap or are just parallel.
                overlap(k) = rcond([dxy1(i(k),:);xy2(j(k),:) - xy1(i(k),:)]) < eps;
            end
        end
        warning(warning_state)
    catch err
        warning(warning_state)
        rethrow(err)
    end
    % Find where t1 and t2 are between 0 and 1 and return the corresponding
    % x0 and y0 values.
    in_range = (T(1,:) >= 0 & T(2,:) >= 0 & T(1,:) <= 1 & T(2,:) <= 1).';
    % For overlapping segment pairs the algorithm will return an
    % intersection point that is at the center of the overlapping region.
    if any(overlap)
        ia = i(overlap);
        ja = j(overlap);
        % set x0 and y0 to middle of overlapping region.
        T(3,overlap) = (max(min(x1(ia),x1(ia+1)),min(x2(ja),x2(ja+1))) + ...
            min(max(x1(ia),x1(ia+1)),max(x2(ja),x2(ja+1)))).'/2;
        T(4,overlap) = (max(min(y1(ia),y1(ia+1)),min(y2(ja),y2(ja+1))) + ...
            min(max(y1(ia),y1(ia+1)),max(y2(ja),y2(ja+1)))).'/2;
        selected = in_range | overlap;
    else
        selected = in_range;
    end
    xy0 = T(3:4,selected).';

    % Remove duplicate intersection points.
    [xy0,index] = unique(xy0,'rows');
    x0 = xy0(:,1);
    y0 = xy0(:,2);

    % Compute how far along each line segment the intersections are.
    if nargout > 2
        sel_index = find(selected);
        sel = sel_index(index);
        iout = i(sel) + T(1,sel).';
        jout = j(sel) + T(2,sel).';
    end
else % non-robust option
    for k = 1:n
        [L,U] = lu(AA(:,:,k));
        T(:,k) = U\(L\B(:,k));
    end

    % Find where t1 and t2 are between 0 and 1 and return the corresponding
    % x0 and y0 values.
    in_range = (T(1,:) >= 0 & T(2,:) >= 0 & T(1,:) < 1 & T(2,:) < 1).';
    x0 = T(3,in_range).';
    y0 = T(4,in_range).';

    % Compute how far along each line segment the intersections are.
    if nargout > 2
        iout = i(in_range) + T(1,in_range).';
        jout = j(in_range) + T(2,in_range).';
    end
end

% Plot the results (useful for debugging).
% plot(x1,y1,x2,y2,x0,y0,'ok');

function y = mvmin(x)
% Faster implementation of movmin(x,k) when k = 1.
y = min(x(1:end-1),x(2:end));

function y = mvmax(x)
% Faster implementation of movmax(x,k) when k = 1.
y = max(x(1:end-1),x(2:end));
```