Measurement of the speed of a rotating shaft through smartphone video acquisition exploiting the rolling shutter effect

T. Verwimp [1,2], **A. Mauricio** [1,2], **K. Gryllias** [1,2]
[1] KU Leuven, Department of Mechanical Engineering,
Celestijnenlaan 300, B-3001, Heverlee, Belgium
e-mail: **toby.verwimp@kuleuven.be**

[2] DMMS Core lab, Flanders Make, Belgium

Abstract
This article investigates the viability of using a smartphone's rolling shutter camera as a cost-effective tool for rotational speed measurement. The smartphone is pointed to the side of the rotating shaft, on which a zebra pattern is pasted. As a consequence of the rolling shutter effect, the zebra pattern is deformed. This deformation depends on the pose of the shaft with respect to the camera, the intrinsic camera parameters and the rotational speed of the shaft. A novel method is proposed to measure this speed using an extended perspective projection model and an optimization algorithm. This method is designed to measure constant and varying speeds, even above the Nyquist limit related to the camera's frame rate, and to be robust against variations in the pose of the shaft with respect to the camera. The method is validated using simulations and on an in-house test rig and shows promising results making it possible to develop a smartphone application to measure the rotational speed of a shaft in the close future.

1 Introduction

Condition monitoring of rotating machinery gains importance in a number of industrial applications including wind turbines and automotive, in order to optimally schedule maintenance and to guarantee operation safety and production efficiency. Diagnostic indicators serve as tools to evaluate the condition of a rotating component and are often based on vibration analysis [1]. However, many rotating machines operate under varying speed conditions, which results in varying vibration characteristics and thus complicates vibration analysis and the proper assessment of their components' health. Nevertheless, a direct measurement or an indirect estimation of the rotational speed can simplify the monitoring process [2, 3]. Other monitoring techniques are based on the rotational speed signals themselves [4, 5, 6].
Several speed measurement sensors are available on the market, like optical encoders, magnetic encoders and Hall sensors [7, 8]. However, the installation of such sensors is often expensive and requires additional space for data acquisition systems and wires. Another sensor that can be used for speed measurements is a high-speed camera. In [9, 10, 11], varying-density fringe patterns are pasted around the rotating shaft and a high-speed camera is used to capture video frames, after which the angular position is extracted using the pattern's density. Also low-speed (and low-cost) cameras have been used to extract the rotational speed: in [12, 13], differences in the shaft's texture are exploited and the similarity between consecutive video frames is evaluated. Subsequently, the rotational speed is extracted by a spectral analysis of the resulting similarity signal. However, low-cost cameras are less effective for these methods as the measurable speed is limited by the Nyquist theorem to half the frame rate, so the expensive high-speed cameras outperform the others. Still, they are more expensive and must process much more data, making them memory- and time-consuming speed measurement tools.
A smartphone also contains at least one camera and could be a very cost-effective tool as the sensor, the

acquisition system and possibly the processing system are all contained into one device. Furthermore, due to its daily use, the training to work with a possible application for speed measurement could be very limited. The camera embedded in a smartphone is a low-speed and low-cost camera and is therefore expected to have a limited performance. However, previous research [14, 15] has shown that the rolling shutter effect inherent to smartphone cameras allows for the improvement of this limited performance. Instead of acquiring all pixel lines at once (as with a global shutter camera), a rolling shutter camera acquires pixel lines subsequently. As a consequence, when there is a relative motion between the captured scene and the smartphone camera, geometrical distortions occur on the images. This effect is prominent for high-speed motions [16]. In [14], a methodology has been developed to extract the rotational speed of the shaft by modeling the geometrical image deformations of a zebra tape due to the rolling shutter effect as a function of the rotational speed. A possible misalignment in the image plane between the rolling shutter direction and the shaft's longitudinal axis is taken into account, as this misalignment influences the deformation of the zebra tape. However, the proposed methodology does not consider misalignments out of the image plane.
In [17, 18] the pose and velocity of an object is estimated using a single view from a rolling shutter camera. The general perspective projection model [19] only holds for global shutter cameras but needs to be extended for rolling shutter cameras when acquiring images of moving objects. This extended model is used in [17] to estimate the speed of an object by using point features and in [18] by using line features. In both studies, the correspondences between the 3D points (or lines) and their 2D projections are assumed to be matched beforehand in a supervised way. Matching these correspondences for a video with hundreds of images could consume a lot of time and effort. The algorithm using lines performs better than using points as lines contain more (redundant) information beneficial for the proposed optimization procedure. Therefore the proposed methodology in this paper extends the approach in [18] to an unsupervised method for the speed estimation of a rotating shaft.
The paper starts with an extensive review of the model developed in [18] together with an extension for the application of interest in Section 2. Subsequently, Section 3 elaborates the practical methodology after which it is experimentally validated in Section 4. Finally, Section 5 draws some conclusions and collects key results and findings.

2 Mathematical model

2.1 Rolling shutter camera projection model

In a rolling shutter camera of a smartphone, the sensor pixels are exposed sequentially starting at the right and proceeding column by column to the left. All pixels have the same integration time, but the start of integration is delayed with respect to the column to the right with a time equal to the rolling shutter period τ. Figure 1 shows a time diagram of the rolling shutter mechanism in a smartphone. After a reset signal, all pixels in a column are exposed during an exposure time t_e. Between the reset signals (and also the readout times) of subsequent columns (starting from the right) there is a delay equal to the rolling shutter period τ. The readout time t_r is the time needed to read out all pixel columns of the smartphone camera and therefore equals $W\tau$, with W the width (i.e. the number of pixel columns) of the images. From the time diagram, the following relation can be deduced:

$$\frac{1}{f} = t_r + t_d = W\tau + t_d \tag{1}$$

with f being the camera's frame rate and t_d the delay between readouts of successive video frames. The geometrical distortions due to the rolling shutter effect increases with the increasing rolling shutter period τ. This parameter is a (fixed) characteristic of the camera, but is unknown a priori. However, previous research has proposed a methodology to estimate this parameter from experiments by using a LED light flashing at a high frequency [20, 21]. In [14], this method is extended to an algorithm using all frames of a video and automatically returning the rolling shutter period τ. Furthermore, extra material is used between the smartphone's camera and the LED to diffuse the light.

The rolling shutter camera projection model elaborated in [17, 18] can be simplified by applying the model on the application of interest, i.e. a rotating shaft (with zebra tape) captured by the rolling shutter camera of

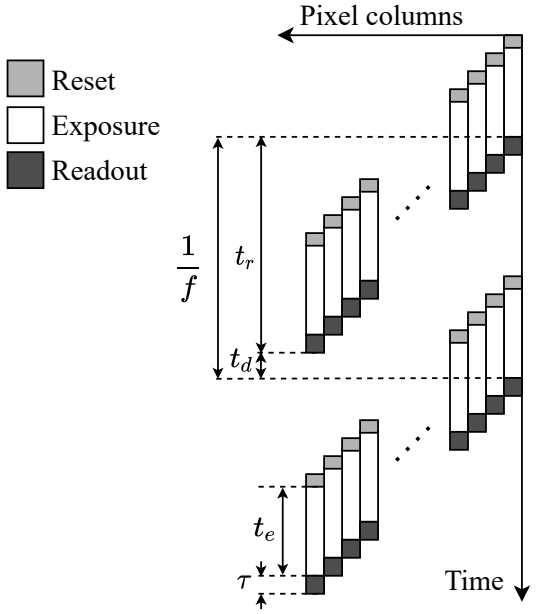


Figure 1: Rolling shutter time diagram.

a stationary smartphone, as shown in figure 2a. The shaft containing a point $\vec{P}_i = [x_i, y_i, z_i]^T$ on its surface moves with rotational speed Ω around a unit vector $\vec{a} = [1, 0, 0]^T$ (i.e. the x-axis). As the smartphone is assumed to be stationary and the shaft is assumed to have only one degree of freedom (i.e. rotation around the x-axis), the translational speed is $\vec{V} = [0, 0, 0]^T$.
The shaft is captured by the smartphone's rolling shutter camera at time t_0. Matrix $\mathbf{R}$ represents the orientation of the world coordinate system (X, Y, Z) with respect to the camera coordinate system (X_C, Y_C, Z_C) and vector $\vec{T}$ represents the position of the world coordinate system (X, Y, Z) with respect to and expressed in the camera coordinate system (X_C, Y_C, Z_C). The dotted straight line along the Z_C-axis represents the optical axis of the smartphone's camera. This optical axis intersects the image plane, which is represented by the 2D coordinate system (x, y) parallel to the camera coordinate system (X_C, Y_C, Z_C), in the principal point $[x_0, y_0]^T$.
In [17, 18], it is demonstrated that the 2D projection $\vec{p}_i = [x_i, y_i]^T$ of $\vec{P}_i$ on the image plane can be expressed as follows:

$$\rho \begin{bmatrix} \vec{p}_i \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{R}\delta\mathbf{R}_i \quad \mathbf{T} + \delta\mathbf{T}_i] \begin{bmatrix} \vec{P}_i \\ 1 \end{bmatrix} \tag{2}$$

with:

$$\mathbf{K} = \begin{bmatrix} k_x & s & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$\delta\mathbf{R}_i = \vec{a}\vec{a}^T(1 - \cos(\tau_i\Omega)) + \mathbf{I}\cos(\tau_i\Omega) + [\vec{a}]_\times \sin(\tau_i\Omega) \tag{4}$$

$$\delta\mathbf{T}_i = \tau_i\Omega\vec{V} = \vec{0} \tag{5}$$

$$\tau_i = \tau(W - x_i)\Omega \tag{6}$$

and ρ an arbitrary scale factor, $\mathbf{K}$ the camera calibration matrix, (k_x, k_y) the number of pixels per unit length horizontally and vertically, τ_i the time after the start of the frame acquisition before $\vec{p}_i$ is captured, $\mathbf{I}$ a 3×3 identity matrix and $[\vec{a}]_\times$ the skew-symmetric matrix related to $\vec{a}$.
Equation 2 expresses the projection of a 3D point on a rotating shaft using a rolling shutter camera with respect to the shaft's orientation and position (with respect to the camera), the rotational speed Ω and the rolling shutter period τ. As a consequence of the motion during the image acquisition, the coordinates of the

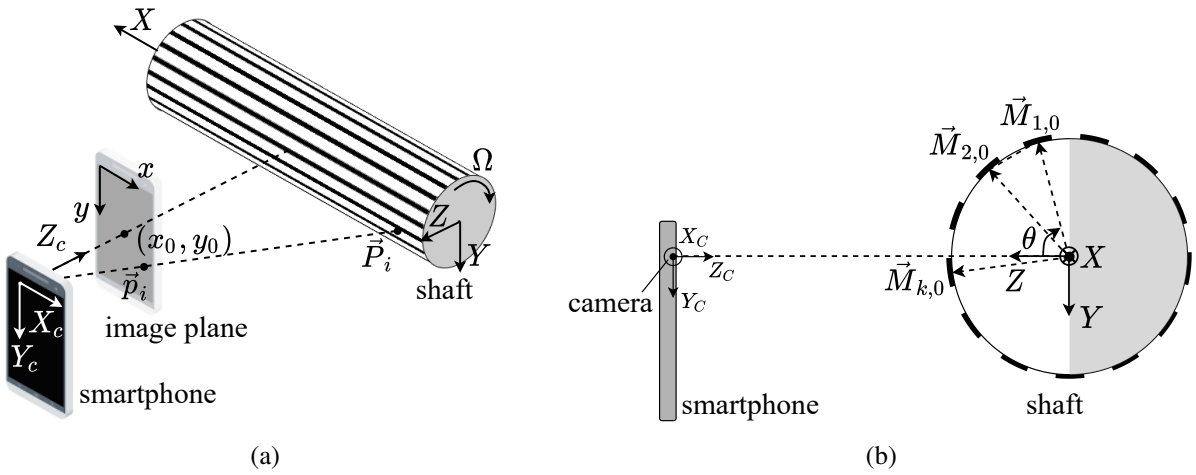


Figure 2: Illustration of a smartphone pointed to a rotating shaft: (a) 3D view (b) from an axial viewpoint (2D).

projected point $\vec{p}_i$ in the image plane depend on both the shaft's motion and the rolling shutter speed, and therefore x_i appears on both sides of equation 2.

2.2 Rotational speed estimation with lines

Using the adjusted rolling shutter camera projection model from section 2.1, a similar model for lines can be developed as in [18]. Assume that a zebra tape consisting of S stripes with length L is pasted around a shaft, as shown in figure 2a. The stripes are parallel to the rotation vector $\vec{a}$ and therefore have direction vectors $\vec{L}_k = [L, 0, 0]^T$ with $k = 1, ..., S$. Then, a point on stripe k can be expressed as:

$$\vec{M}_{k,i} = \vec{M}_{k,0} + \sigma_{k,i}\vec{L}_k \tag{7}$$

with $\vec{M}_{k,0}$ an arbitrary point on stripe k and $\sigma_{k,i}$ a dimensionless factor. In this approach, the points $\vec{M}_{k,0}$ are chosen to be at the right end of the corresponding stripe, i.e. in the YZ-plane, which results in $\sigma_{k,i}$ to be between 0 and 1. As the stripes are equally spread over the shaft's circumference, their position can be characterized by only one unknown parameter, the angle θ as shown in figure 2b:

$$\vec{M}_{k,0} = \frac{D}{2}\begin{bmatrix} 0 \\ -\sin\left(\theta - (k-1)\frac{2\pi}{S}\right) \\ \cos\left(\theta - (k-1)\frac{2\pi}{S}\right) \end{bmatrix} \tag{8}$$

Applying the rolling shutter camera projection model (equation 2) to the stripes results in the following model:

$$\rho\begin{bmatrix}\vec{m}_{k,i} \\ 1\end{bmatrix} = \mathbf{K}[\mathbf{R}\delta\mathbf{R}_i \quad \mathbf{T}]\begin{bmatrix}\vec{M}_{k,0} + \sigma_{k,i}\vec{L}_k \\ 1\end{bmatrix} \tag{9}$$

with $\vec{m}_{k,i}$ the projection of point $\vec{M}_{k,i}$ on the image plane. This results in a pair of constraints for each pixel along the captured stripes:

$$\begin{aligned} x_{k,i} &= k_x \frac{\vec{R}_i^{(1)}\left(\vec{M}_{k,0}+\sigma_{k,i}\vec{L}_k\right)+T^{(x)}}{\vec{R}_i^{(3)}\left(\vec{M}_{k,0}+\sigma_{k,i}\vec{L}_k\right)+T^{(z)}} + x_0 \\ y_{k,i} &= k_y \frac{\vec{R}_i^{(2)}\left(\vec{M}_{k,0}+\sigma_{k,i}\vec{L}_k\right)+T^{(y)}}{\vec{R}_i^{(3)}\left(\vec{M}_{k,0}+\sigma_{k,i}\vec{L}_k\right)+T^{(z)}} + y_0 \end{aligned} \tag{10}$$

with $\vec{R}_i^{(j)}$ the j^{th} row of $\mathbf{R}\delta\mathbf{R}_i$ and $T^{(x,y,z)}$ the components of $\vec{T}$. The above constraints are expressed as a

function of the unknown parameters:

$$\vec{m}_{k,i} = \begin{bmatrix} x_{k,i} \\ y_{k,i} \end{bmatrix} = \begin{bmatrix} \xi_{x,k,i}\left(\mathbf{R}, \vec{T}, \theta, \Omega, \vec{\Sigma}\right) \\ \xi_{y,k,i}\left(\mathbf{R}, \vec{T}, \theta, \Omega, \vec{\Sigma}\right) \end{bmatrix} \tag{11}$$

with $\vec{\Sigma}$ the vector containing all the parameters $\sigma_{k,i}$. In order to have a vector of optimization variables, matrix $\mathbf{R}$ is represented by a vector of XYZ Euler angles, i.e. $\vec{\phi} = [\phi_X, \phi_Y, \phi_Z]^T$. As the deformation of the stripes in the image plane depends on the rotational speed [14], the number of stripes that the camera captures is not known a priori. Furthermore, the camera can only see almost half the shaft's circumference. Section 3 proposes a methodology to deal with this.
For a captured frame, each one of the Q deformed stripes with N observed pixels $\hat{\vec{m}}_{k,i} = [\hat{x}_{k,i}, \hat{y}_{k,i}]^T$ matches a stripe on the shaft's surface. These observed pixels can be compared with the theoretical projections using equation 11, which results in a $Q \times N \times 2$ equation system representing the reprojection error:

$$\vec{\epsilon}_{k,i} = \hat{\vec{m}}_{k,i} - \vec{m}_{k,i} = \begin{bmatrix} \hat{x}_{k,i} - \xi_{x,k,i}\left(\mathbf{R}, \vec{T}, \theta, \Omega, \vec{\Sigma}\right) \\ \hat{y}_{k,i} - \xi_{y,k,i}\left(\mathbf{R}, \vec{T}, \theta, \Omega, \vec{\Sigma}\right) \end{bmatrix} \tag{12}$$

The magnitude of the reprojection error should be minimized for all pixels resulting in the following objective function:

$$\epsilon = \sum_{k=1}^{Q}\sum_{i=1}^{N} \vec{\epsilon}_{k,i}^T \vec{\epsilon}_{k,i} = \sum_{k=1}^{Q}\sum_{i=1}^{N} \left[\hat{x}_{k,i} - \xi_{x,k,i}\left(\mathbf{R}, \vec{T}, \theta, \Omega, \vec{\Sigma}\right)\right]^2 + \left[\hat{y}_{k,i} - \xi_{y,k,i}\left(\mathbf{R}, \vec{T}, \theta, \Omega, \vec{\Sigma}\right)^2\right] \tag{13}$$

This minimization problem is a non-linear least squares problem which is discussed in section 3.3.

3 Methodology

Figure 3 shows the entire process to go from a video of a rotating shaft from a radial viewpoint to the rotational speed estimation ω. The algorithms to calibrate the rolling shutter period τ, to extract the shaft from the video frames and to calibrate the smartphone's camera are discussed in [14]. The main difference is that the distance between the camera and the shaft is part of the result of the optimization and will not be extracted from the camera calibration results. The following sections further elaborate the methodology to convert the extracted shaft in all video frames to an estimation of the rotational speed. The methods have been implemented in MATLAB [22].

3.1 Speed estimation

The speed estimation process is shown in figure 4, in which the right block is the enlarged version of the purple box in the left block and the left block corresponds to the blue box in figure 3. This process is applied to all video frames, to eventually determine the shaft's rotational speed for all time steps.
Once the shaft's mask is extracted, it is used to extract the shaft in all gray-scale images of the video frames, which are first corrected for radial and tangential distortion. As this mask is fixed, the smartphone should be kept stationary with respect to the shaft. This can be realised by using a tripod stand. As only grayscale is needed, the settings of the smartphone's camera can be tuned to serve as a monochrome camera, which saves space on the operator's device.
Subsequently, the image is cropped to a Region Of Interest (ROI) such that artefacts at the shaft's edges are not influencing the speed estimations. Next, the image is binarized (shown in figure 5b) using Otsu's algorithm to automatically determine the intensity threshold [23]. Briefly, this method divides the intensity histogram into two classes by maximizing the variance between the classes. The boundary between these classes is the intensity threshold to convert gray-scale images to binary images. In this way, variations in

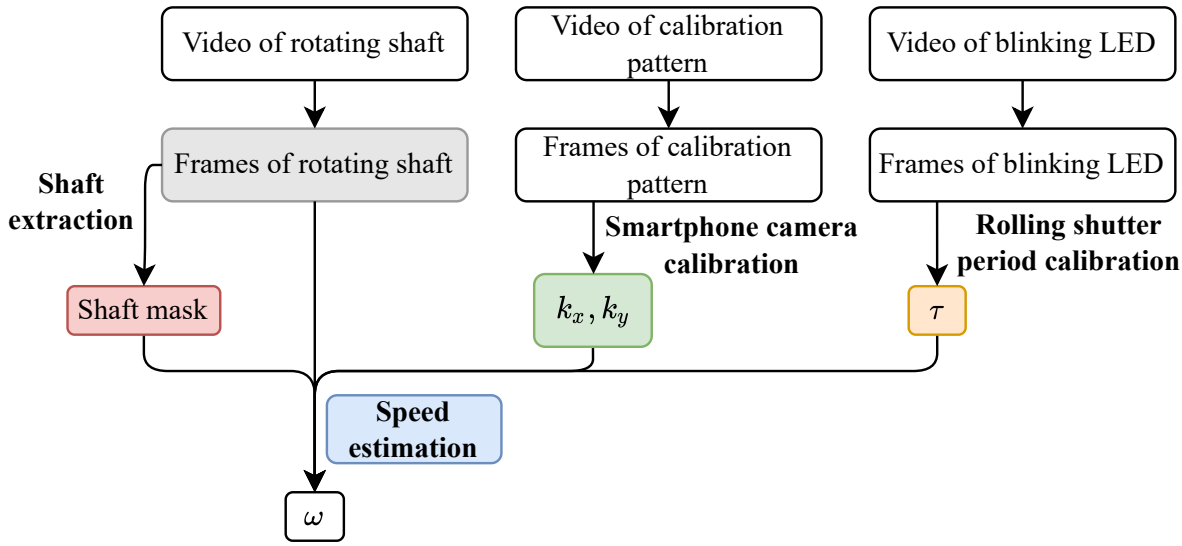


Figure 3: Flowchart of the methodology to extract the speed from a radial viewpoint.

brightness over time can be accounted for.
The next step is extracting N pixels on each of the Q observed stripes in the binary image, which is elaborated in section 3.2. Finally, the observed pixels $\hat{\vec{m}}_{k,i}$ serve as an input for the optimization algorithm discussed in sections 2.2 and 3.3.

3.2 Stripe pixel extraction

The next step is extracting the pixel coordinates $\hat{\vec{m}}_{k,i}$ from the binary image. The first steps until the extraction of the stripe edges are the same as discussed in [14]. Next, each one of the Q stripes is represented by N equally spaced pixel coordinates of the corresponding detected upper edges. The $Q \times N \times 2$ tensor containing these pixel coordinates is however not yet ready to be used in the optimization algorithm: equation 8 requires the stripes in that tensor to be adjacent, but this is not true by default as the stripes (found by using MATLAB's bwconncomp) are ordered according to columns and then rows.
In order to arrange the stripes to be adjacent, the center of the edges is calculated from the N coordinates, after which they are sorted according to the center's coordinate in the direction (x or y) with the most variation in the centers' coordinates. Next, the stripes and their N edge pixels are sorted from right to left in order to be compatible with the assumptions in section 2.2.
Last but not least, it is possible that more stripes are visible than there physically are on the tape ($Q > N$). Therefore, this stripe is merged with the first occurence of this stripe and the N pixels are redistributed over the merged stripes, to finally get the N pixel coordinates $\hat{\vec{m}}_{k,i}$ for all Q observed stripes.

3.3 Reprojection error minimization

In order to start the optimization process using equation 13, an initial guess is needed for the optimization variables $\mathbf{R}, \vec{T}, \theta, \Omega$ and $\vec{\Sigma}$. For $\mathbf{R}_0$ and $\vec{T}_0$ a rough initial guess can be made based on the setup or they can be estimated in the smartphone camera calibration process. The angular position θ_0 of the most right pixel of the first stripe is initialized to 0 rad. Next, the rotational speed Ω_0 is initialized to the middle of the speed range. Finally, the vector $\vec{\Sigma}_0$ containing the dimensionless parameters $\sigma_{k,i}$ is initialized automatically based on the projection of the pixel coordinates $\hat{\vec{m}}_{k,i}$ on the major direction of the cylinder in the image plane.
After the variables are optimized, the estimation for the orientation $\mathbf{R}$, the position $\vec{T}$ and the rotational speed $\hat{\Omega}$ can serve as an initial guess for the minimization of the reprojection error in the next video frame.

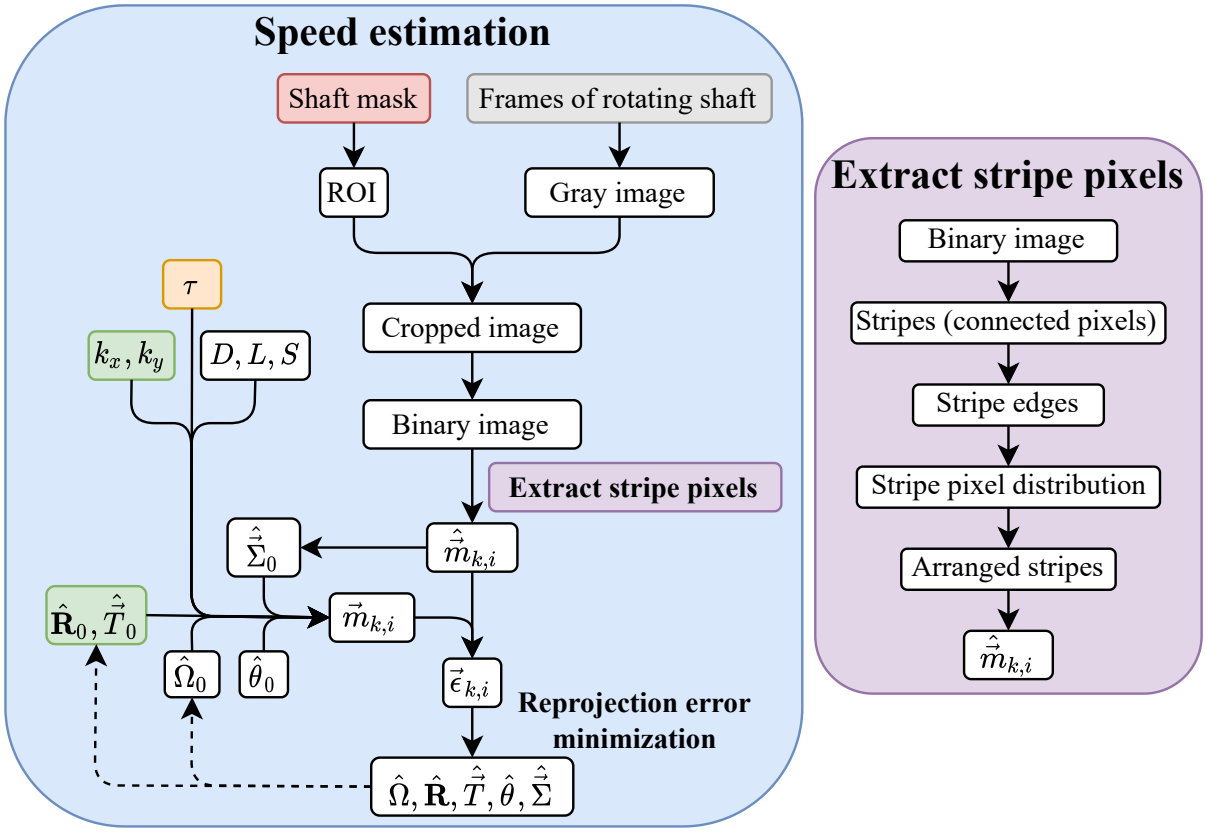


Figure 4: Flowchart of the methodology to estimate the speed and to extract the stripes' pixel coordinates.

However, to reduce the computation time, these interframe dependencies can be omitted such that the optimization can be computed in parallel using multiple cores of the machine.
The constraints in this problem are only lower and upper bounds for all optimization variables. Furthermore, the optimization variables are scaled in order to have about the same effect on the objective and constraint functions. In this way the scaled dimensionless optimization variables are obtained by dividing them with their expected maximum value (the upper bound) and the constraint functions are as follows:

$$\tilde{\Omega} = \frac{\hat{\Omega}}{\Omega_{max}}, \qquad 0 \leq \tilde{\Omega} \leq 1 \tag{14}$$

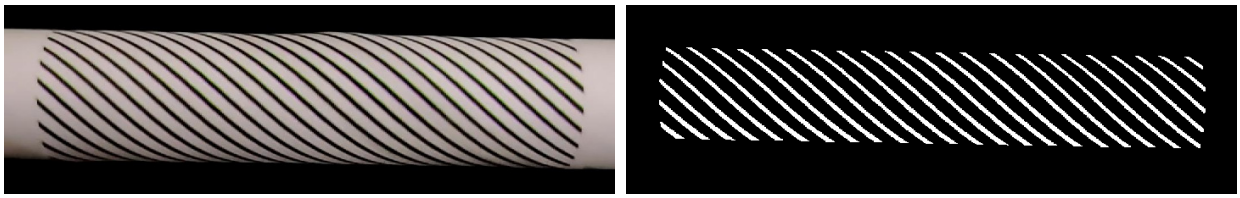
$$\tilde{\vec{\phi}} = \frac{\hat{\vec{\phi}}}{\vec{\phi}_{max}}, \qquad -\vec{1} \leq \tilde{\vec{\phi}} \leq \vec{1} \tag{15}$$

$$\tilde{\vec{T}} = \frac{\hat{\vec{T}}}{\vec{T}_{max}}, \qquad \tilde{\vec{T}}_{lb} \leq \tilde{\vec{T}} \leq \vec{1} \tag{16}$$

$$\tilde{\theta} = \frac{\hat{\theta}}{\theta_{max}}, \qquad -1 \leq \tilde{\theta} \leq 1 \tag{17}$$

$$\tilde{\vec{\Sigma}} = \hat{\vec{\Sigma}}, \qquad \vec{0} \leq \tilde{\vec{\Sigma}} \leq \vec{1} \tag{18}$$

in which "$\leq$" holds elementwise for vectors and $\vec{1}$ is a vector of ones with the same number of elements as the vector to which it is compared. $\tilde{\vec{T}}_{lb}$ is a dimensionless lower bound for the position of the smartphone's camera with respect to the shaft and should be chosen based on the specific setup.
As elaborated in section 4 of [18], the Jacobian matrix of this non-linear least squares problem is very sparse and has the structure shown in figure 6 for this application. Therefore, a trust-region-reflective least squares



(a) (b)

Figure 5: Image example: (a) A cropped video frame of a rotating shaft with a zebra pattern (b) Binary image of the ROI of the frame in figure 5a.

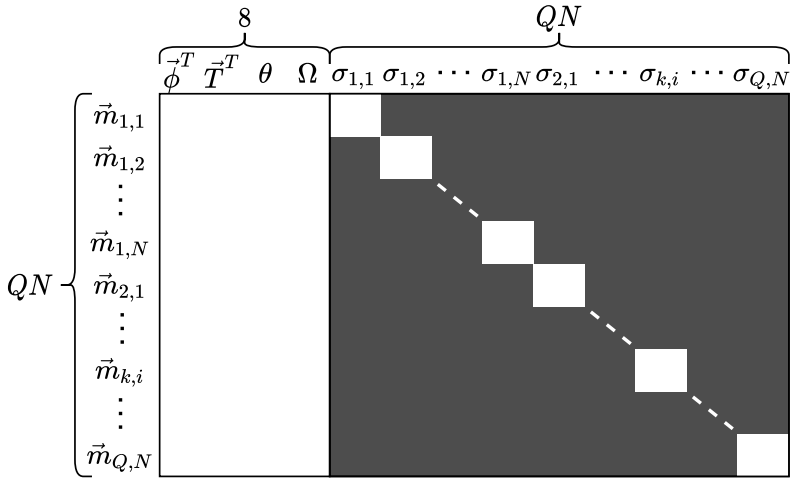


Figure 6: Structure of the Jacobian matrix.

algorithm is used as recommended by MATLAB [24]. Furthermore, this known sparse structure is given as an option in the algorithm in order to decrease the computation time.

4 Validation of the proposed methodology

This section validates the proposed methodology on simulations and on an in-house test rig. First, the following section describes the performed simulations with their results. Thereafter, a brief description of the test rig is provided, after which the results of the experiments performed on the test rig are discussed. For both simulations and experiments, the estimated speed signals using the proposed methodology are compared with a reference speed profile and with the speeds estimated using the methodology proposed in previous research [14].

4.1 Validation of the proposed methodology on simulations

Before experimentally validating the proposed methodology, a video of a rotating shaft can also be simulated. By integrating a speed profile Ω to an angle θ, i.e. the shaft's angular position, the position of the stripes on the shaft can be known for any time step. In order to generate the images, the rolling shutter time diagram (see figure 1) is used: the shaft's angular speed is integrated until a pixel column should be read out. At that moment, all visible stripes on the shaft are projected on the image plane and only the pixel values in the current exposed column are retained. Then the speed is further integrated until the next pixel column is read out. This process is repeated until the full image width is covered. Thereafter, the speed is further integrated until the start of the next video frame. In order to determine whether the camera has visual access to a point,

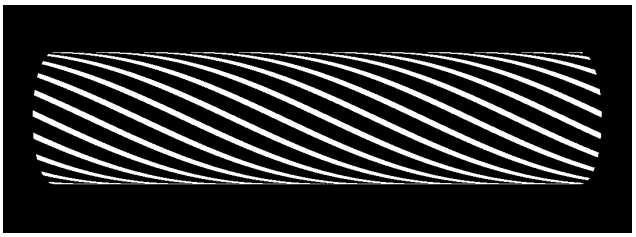


Figure 7: Example of a cropped video frame from a simulation.

the point's normal to the shaft's surface is projected on the projection vector, i.e. the vector from the camera center to the point of interest. If this projection (inproduct) is negative, then the point is visible to the camera. An example binary image (already cropped) is shown in figure 7. The shaft with zebra tape has the same dimensions as described in section 4.2. Furthermore the position of the camera and the camera intrinsics are about the same as in the experiments. The used speed profile corresponds to the measured reference speed profile of figure 10b but scaled down to 20 s (instead of 54 s) in order to reduce the simulation's computation time. The rolling shutter period τ is set to 25 µs. The speed in following simulations varies between 10 and 50 rps. Therefore, the maximum speed Ω_{max} defined in section 3.3 is set to 60 rps (120π rad/s). Furthermore, the maximum Euler angles $\vec{\phi}_{max}$ are set to $[\pi, \pi, \pi]^T$, the lower bound $\vec{T}_{lb}$ and the maximum $\vec{T}_{max}$ for the translation are respectively set to -0.30 m and 0.30 m in all three directions, and θ_{max} is set to π. Finally, the algorithm uses 150 pixels per stripe (i.e. N in section 2.2).

Three simulations are performed: the first one with the image plane aligned with the shaft and the camera at half the shaft's length, the second one with the camera at the left edge of the shaft and pointing to the middle of the shaft and a third one with the camera having the same orientation as in the first simulation, but translated down (i.e. the shaft appears near the top of the images). Finally, in a fourth simulation the camera is at the same position as in the first simulation but rotated 30° about the Z-axis of the world coordinate system. For simulations 1, 2 and 4, the camera's position is initialized at a distance of 0.16 m from the shaft's center and its orientation is initialized as in figure 2. This initial guess corresponds to the correct camera pose for only the first simulation. For the third simulation, the camera's position is also initialized correctly.
For all simulations, figure 8 shows the reference speed profile Ω (blue curve) on the top graphs, together with the speed estimation ω_1 based on the simulated frames using the previous methodology (red curve) and the speed estimation ω_2 based on the simulated frames using the methodology proposed in this article. On the bottom graphs, the estimation errors with respect to the reference speed are shown for both estimations.
From simulations 1, 2 and 4 (except for some outliers), the proposed methodology performs better than the previous one. However, simulation 3 shows worse results. The cause of the outliers and the worse speed estimation from the third simulation is probably due to the optimization behavior of the algorithm: as the optimization algorithm is only capable of finding local minima, there is no certainty that the found minima are also global minima. Therefore, a good initial guess of all optimization variables is recommended. Further research is needed to extend the proposed methodology to find global minima of the reprojection error. Nevertheless, the Normalized Root-Mean-Square Errors (NRMSE) of both estimations (see table 1) show promising improvements for three of the four simulations. Therefore, the next step is to test the proposed methodology on an in-house test rig, which is described in the following section.

Table 1: NRMSE for the simulation results shown in figure 8.

Simulation	**NRMSE 1 (%)**	**NRMSE 2 (%)**
1	7.21	0.10
2	18.27	0.18
3	11.97	18.07
4	7.03	5.12

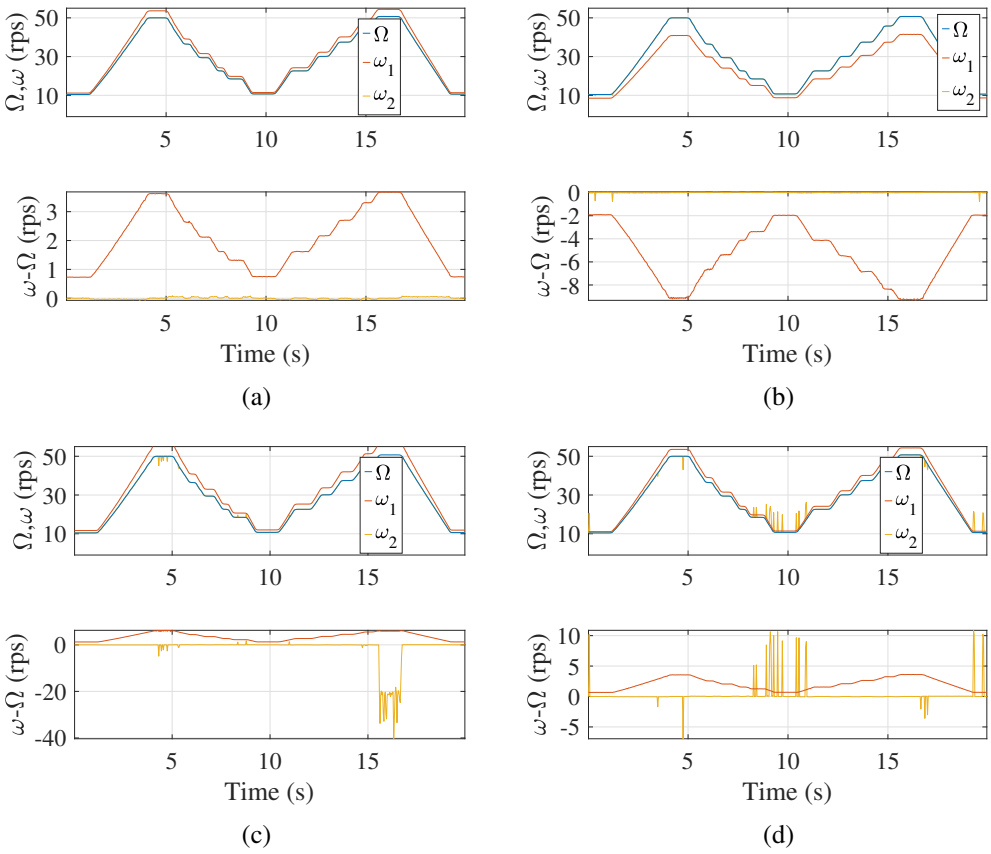


Figure 8: Speed estimations and errors from four (a,b,c,d) simulations.

4.2 Description of the experimental test rig

The test rig used for the experiments is presented in figure 9. The main setup consists of an induction motor[1] (in blue) with a control unit (not shown on the picture) to set a speed target for the rotating shaft with diameter $D = 0.025$ m. Two bearings (with black housings) hold the shaft, which is connected to the motor through a jaw/spider coupling. At the back of the motor an incremental rotary encoder[2] with a resolution of 1024 ppr measures the shaft's rotational speed which modulates a square wave at the frequency of the motor driver. Therefore, the speed can be extracted by low-pass filtering and spectral analysis, and is afterwards also resampled to the video frame rate to be able to use it as a reference speed Ω for the speed estimations $\omega = \hat{\Omega}$ using the proposed methodology.

The same zebra pattern is pasted on the shaft (between the two bearings) as in [14] and contains twenty stripes with a thickness of 1 mm and a length L of 10 mm. After the zebra pattern is pasted on the shaft (between the two bearings), the (mis)alignment of the motor with the rotating shaft is checked and is corrected by putting shims at the bases. Subsequently, the smartphone is mounted on a tripod for stabilization with its camera pointed to the shaft. Next, the smartphone's camera is calibrated using a checkerboard with square sizes of 11 mm.

Next to the smartphone, a tripod is positioned with a microphone, which is (together with the encoder) connected to a data acquisition system[3]. This is in turn connected to a laptop running Simcenter Testlab Signature Acquisition to collect and save the acquired measurements. The microphone serves as a trigger to start the data acquisition. If the sound pressure exceeds 2 Pa, which is equivalent to a sound pressure level of 100 dB, the acquisition is started. After starting the video recording, this pressure can be reached by

[1] Motor specifications are available at the website: https://ap.lc/ychbo

[2] Encoder specifications are available at the website: https://ap.lc/PkJzy

[3] Simcenter SCADAS XS: https://www.plm.automation.siemens.com/global/en/products/simcenter/scadas-xs.html

Figure 9: A picture of the test rig with additional DC lighting and a smartphone and microphone mounted on a tripod.

clapping hands or by snapping fingers near the microphone. A similar peak is then present in the audio signal of the recorded video, making it possible to synchronize the speed estimation ω through the video with the reference speed Ω measured by the encoder. As the sampling frequency of the smartphone audio signal is 48 kHz and thus much higher than the camera's frame rate f, it is assumed that the audio is recorded from the start of the video, while the time instant for the speed estimation in the first frame is equal to $1/(2f)$. This delay is taken into account in the synchronization process.
The videos are captured using the application ProCam X Lite which allows tuning camera settings (such as the exposure time and focus) for video recordings. This tuning is needed in order to acquire sharp images of the rotating shaft. The focus is set to manual and fixed mode to prevent the autofocus from changing focus during recording. However, the application is limited to recordings with a frame rate of 30 fps. To check whether it is possible to extract speeds way above the Nyquist limit of 15 rps (for a frame rate of 30 fps), the speed in following experiments varies between 10 and 50 rps. Therefore, the maximum speed Ω_{max} defined in section 3.3 is set to 60 rps (120π rad/s).
A final part of the test rig are two DC lights directed to the zebra pattern. As the exposure time is reduced to capture sharp images, less light reaches the camera sensor, so to increase the contrast between the white paper and the black stripes and between the shaft and its background, an additional light source is required. An AC light source cannot be used, as this causes a flickering effect on the captured video frames. The LED light of the smartphone could be used too, but this is not possible with the application used to capture the videos.

4.3 Validation of the proposed methodology on experimental data, results and discussion

The algorithm uses the same initial guess as for simulations 1, 2 and 4, but with a slightly different distance (about 0.18 m) between the camera and the shaft's center. Another difference with the simulations is that the camera intrinsics are calibrated using images of a checkerboard pattern, while in the simulations the camera intrinsics used to generate the images are also used in the speed estimation process. Furthermore, in the simulations 95 % of the diameter is used, while in the experiments only 70 % could be used due to artefacts appearing near the shaft's edges.
Figure 10 shows the speed Ω measured by the encoder, the estimated speed ω_1 using the methodology from [14] and the estimated speed ω_2 using the methodology proposed in section 3. The estimates ω_2 are noisier than the original estimates ω_1, as shown by the estimation error (second row of figures).
Possible causes for these larger errors are the increased number of degrees of freedom and errors in the camera calibration: an error in the camera calibration result will influence the reprojection error $\vec{\epsilon}_{k,i}$ and allows the optimization variables to change in order to minimize the reprojection error. This could possibly result in worse speed estimations.

Figure 11 also shows the two estimates ω_1 and ω_2, and the reference speed Ω, but for different orientations of the smartphone with respect to the rotating shaft. The first estimate ω_1 is again less noisy than the second

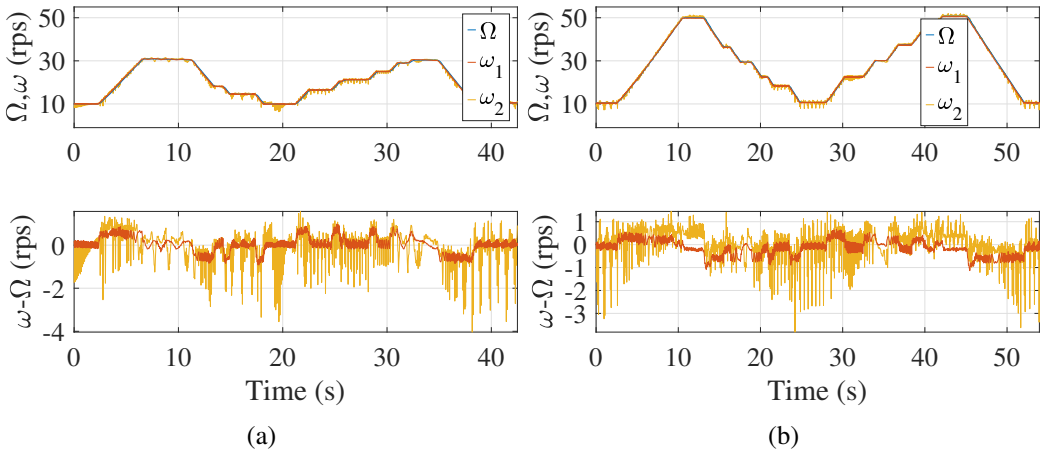


Figure 10: Speed estimations for speeds varying between: (a) 10 and 30 rps (b) 10 and 50 rps.

estimate ω_2, and for the latter, the noise increases with the tilt angle of the smartphone. In the original algorithm (estimate ω_1) this tilt angle is extracted and taken into account in the speed estimation process, while in the current algorithm (estimate ω_2) this tilt angle is part of the orientation matrix $\mathbf{R}$ which is also an optimization variable. Furthermore, when the camera is tilted more with respect to the shaft, the stripes span less pixels and therefore more stripes are filtered out as they consist of less than 150 (N) horizontally adjacent pixels. The NRMSEs for both methodologies in table 2 confirm the above observations.
As mentioned in section 3.3, the optimized variables have not been used as an initial guess in the subsequent frame, in order to reduce the computation time. However, this option has been also tested and almost the same results have been obtained.
Finally, the proposed algorithm is designed to estimate the speed irrespective of the position or orientation of the smartphone with respect to the rotating shaft, while the original algorithm can only take into account misalignments between the rolling shutter direction and the shaft in the image plane. Only experiments with these misalignments are shown, resulting in better results when using the original algorithm. Future experiments will validate both methodologies for other positions and orientations of the camera with respect to the shaft and, based on the simulations, it is expected that the proposed methodology will outperform the original one.

Table 2: NRMSE for the simulation results shown in figures 10 and 11.

Figure	**NRMSE 1 (%)**	**NRMSE 2 (%)**
10a	1.76	3.72
10b	1.16	2.17
11a	1.81	3.18
11b	1.87	31.00
11c	2.58	9.44
11d	4.07	14.44

5 Conclusion

This paper proposes a method to measure the speed of rotating machinery from a radial viewpoint and by exploiting the rolling shutter effect. A zebra pattern is pasted around a shaft and due to the rolling shutter effect, the black stripes deform in the image plane. This deformation depends on the pose of the shaft with respect to the camera, the intrinsic camera parameters and the rotational speed of the shaft. A novel method is proposed to measure this speed using an extended perspective projection model and an

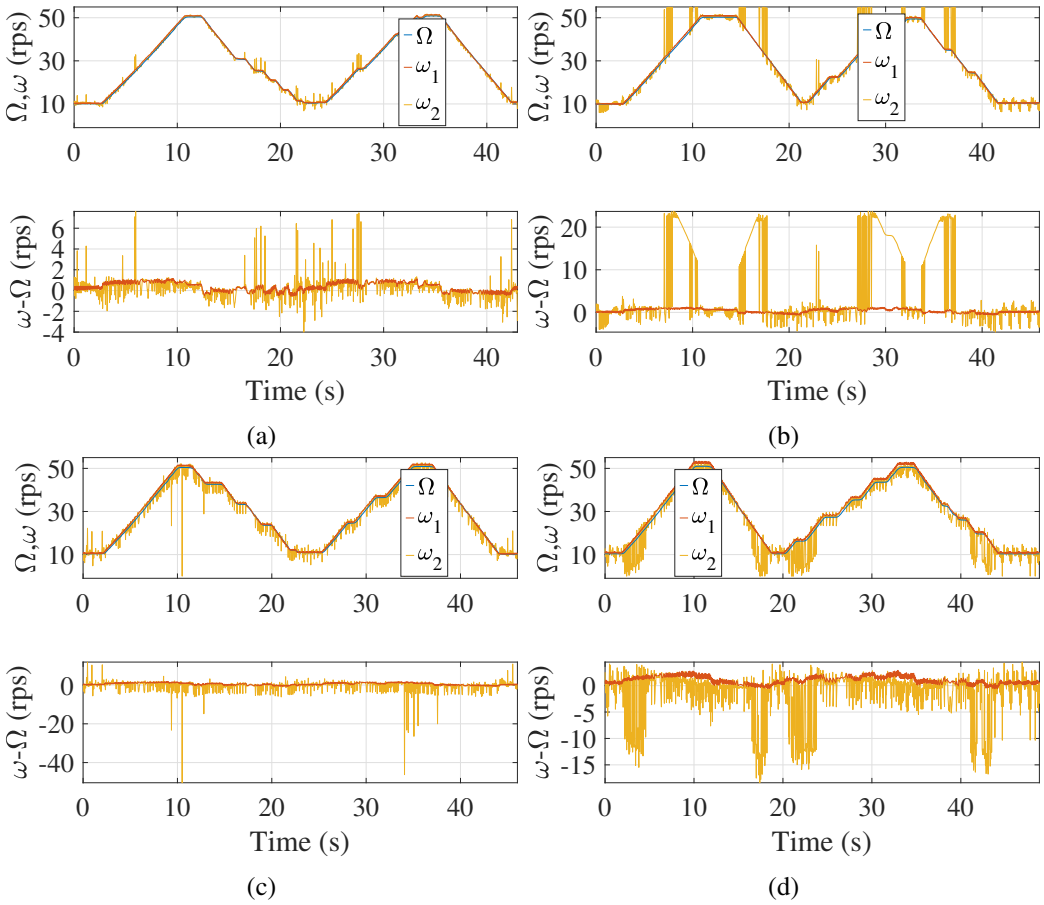


Figure 11: Speed estimations for speeds varying between 10 and 50 rps, for tilt angles of the smartphone of about: (a) 15°, (b) 30°, (c) 45°, (d) 60°.

optimization algorithm. This method is designed to measure constant and varying speeds, even above the Nyquist limit related to the camera's frame rate, and to be robust against variations in the pose of the shaft with respect to the camera. The method is validated on an in-house test rig and shows promising results making it possible to develop a smartphone application to measure the rotational speed of a shaft in the future. However, the measurement errors are still big, which results in the need for more accurate camera calibration and better optimization algorithms for future experiments.

References

[1] V. Sharma and A. Parey, "A review of gear fault diagnosis using various condition indicators," *Procedia Engineering*, vol. 144, pp. 253–263, 2016.

[2] A. Raad, J. Antoni, and M. Sidahmed, "Indicators of cyclostationarity: Theory and application to gear fault monitoring," *Mechanical systems and signal processing*, vol. 22, no. 3, pp. 574–587, 2008.

[3] A. Mauricio, L. Zhou, D. Mba, and K. Gryllias, "Vibration-based condition monitoring of helicopter gearboxes based on cyclostationary analysis," *Journal of engineering for gas turbines and power*, vol. 142, no. 3, 2020.

[4] H. André, D. Rémond, and A. Bourdon, "On the use of the instantaneous angular speed measurement in non-stationary mechanism monitoring," in *ASME 2011 International Design Engineering Technical Conferences*, ser. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. Volume 1: 23rd Biennial Conference on Mechanical Vibration and Noise, Parts A and B, Washington, United States, 2011, pp. 15–24.

[5] W. Moustafa, O. Cousinard, F. Bolaers, K. Sghir, and J. Dron, "Low speed bearings fault detection and size estimation using instantaneous angular speed," *Journal of vibration and control*, vol. 22, no. 15, pp. 3413–3425, 2016.

[6] M. Zhao, X. Jia, J. Lin, Y. Lei, and J. Lee, "Instantaneous speed jitter detection via encoder signal and its application for the diagnosis of planetary gearbox," *Mechanical systems and signal processing*, vol. 98, pp. 16–31, 2018.

[7] Y. Li, F. Gu, G. Harris, A. Ball, N. Bennett, and K. Travis, "The measurement of instantaneous angular speed," *Mechanical systems and signal processing*, vol. 19, no. 4, pp. 786–805, 2005.

[8] S. Lu, R. Yan, Y. Liu, and Q. Wang, "Tacholess speed estimation in order tracking: A review with application to rotating machine fault diagnosis," *IEEE transactions on instrumentation and measurement*, vol. 68, no. 7, pp. 2315–2332, 2019.

[9] J. Zhong, S. Zhong, Q. Zhang, Z. Peng, S. Liu, and Y. Yu, "Vision-based measurement system for instantaneous rotational speed monitoring using linearly varying-density fringe pattern," *IEEE transactions on instrumentation and measurement*, vol. 67, no. 6, pp. 1434–1445, 2018.

[10] J. Zhong, S. Zhong, Q. Zhang, and Z. Peng, "Measurement of instantaneous rotational speed using double-sine-varying-density fringe pattern," *Mechanical systems and signal processing*, vol. 103, pp. 117–130, 2018.

[11] J. Zhong, S. Zhong, Q. Zhang, N. Maia, Y. Shen, S. Liu, Y. Yu, and Z. Peng, "Vision-based system for simultaneous monitoring of shaft rotational speed and axial vibration using non-projection composite fringe pattern," *Mechanical systems and signal processing*, vol. 120, pp. 765–776, 2019.

[12] T. Wang, L. Wang, Y. Yan, and S. Zhang, "Rotational speed measurement using a low-cost imaging device and image processing algorithms," in *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. Houston, United States: IEEE, 2018, pp. 1–6.

[13] T. Wang, Y. Yan, L. Wang, Y. Hu, and S. Zhang, "Instantaneous rotational speed measurement using image correlation and periodicity determination algorithms," *IEEE transactions on instrumentation and measurement*, vol. 69, no. 6, pp. 2924–2937, 2020.

[14] T. Verwimp and K. Gryllias, "Rotating machinery speed extraction through smartphone video acquisition," Master's thesis, Leuven, Belgium, 2021.

[15] H. André, Q. Leclère, D. Anastasio, Y. Benaïcha, K. Billon, M. Birem, F. Bonnardot, Z. Chin, F. Combet, P. Daems, A. Daga, R. De Geest, B. Elyousfi, J. Griffaton, K. Gryllias, Y. Hawwari, J. Helsen, F. Lacaze, L. Laroche, X. Li, C. Liu, A. Mauricio, A. Melot, A. Ompusunggu, G. Paillot, S. Passos, C. Peeters, M. Perez, J. Qi, E. Sierra-Alonso, W. Smith, and X. Thomas, "Using a smartphone camera to analyse rotating and vibrating systems: Feedback on the survishno 2019 contest," *Mechanical systems and signal processing*, vol. 154, p. 107553, 2021.

[16] C.-K. Liang, L.-W. Chang, and H. Chen, "Analysis and compensation of rolling shutter effect," *IEEE transactions on image processing*, vol. 17, no. 8, pp. 1323–1330, 2008.

[17] O. Ait-Aider, N. Andreff, J. M. Lavest, and P. Martinet, "Simultaneous object pose and velocity computation using a single view from a rolling shutter camera," in *Computer Vision – ECCV 2006*, ser. Lecture Notes in Computer Science, vol. 3952. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 56–68.

[18] O. Ait-Aider, A. Bartoli, and N. Andreff, "Kinematics from lines in a single rolling shutter image," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. Minneapolis, United States: IEEE, 2007, pp. 1–6.

[19] L. Van Gool, T. Tuytelaars, G. Szekely, and V. Ferrari, *Pattern Recognition and Image Interpretation: Part 2: Image processing and Image interpretation*, 2016-2017.

[20] M. Meingast, C. Geyer, and S. Sastry, "Geometric models of rolling-shutter cameras," *Computing Research Repository (CoRR)*, 2005.

[21] E. Ringaby and P.-E. Forssén, "Efficient video rectification and stabilisation for cell-phones," *International journal of computer vision*, vol. 96, no. 3, pp. 335–352, 2012.

[22] (2022) Matlab - overview - math. graphics. programming. The MathWorks, Inc. [Online]. Available: https://nl.mathworks.com/products/matlab.html

[23] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[24] (2022) Optimization toolbox - choosing the algorithm. The MathWorks, Inc. [Online]. Available: https://nl.mathworks.com/help/optim/ug/choosing-the-algorithm.html#bsbw9tv