Jernkontorets forskning

Rapport D 896

Computer vision, Condition-based monitoring, and Data fusion

Report by Nusyba Yesmin, Kalle Kurttio, Waleed Taria

Final report JK TO 60 2023-12-10

(ISSN 0280-249X)

Distribution

Denna rapport är öppen

Jernkontoret

UNIVERSITY
OF GÄVLE

FACULTY OF ENGINEERING AND SUSTAINABLE DEVELOPMENT
Department of Electrical Engineering, Mathematics and Science

Joint report for Jernkontoret

Computer vision, Condition-based monitoring, and Data fusion

Nusyba Yesmin, Kalle Kurttio, Waleed Tariq

2023

1 Table of contents

List of Figures

2 List of Tables

1 Introduction

Iba-systems is a widely integrated system in the steel industry. With a specific prominence on iba-systems, the members of Jernkontoret aim to achieve uniformity in software usage. Standardizing the software for data acquisition and analysis makes it easier to maintain knowledge and expertise within the company. As part of Jernkontorets digitization program TO 60 a vision of exploring the potential and future applications of already integrated modules resulted in a collaboration project consisting of three theses. Three case studies serve as the foundation for investigating the potential and future use of various iba-modules.

The thesis are related to computer vision, data fusion and condition-based monitoring.

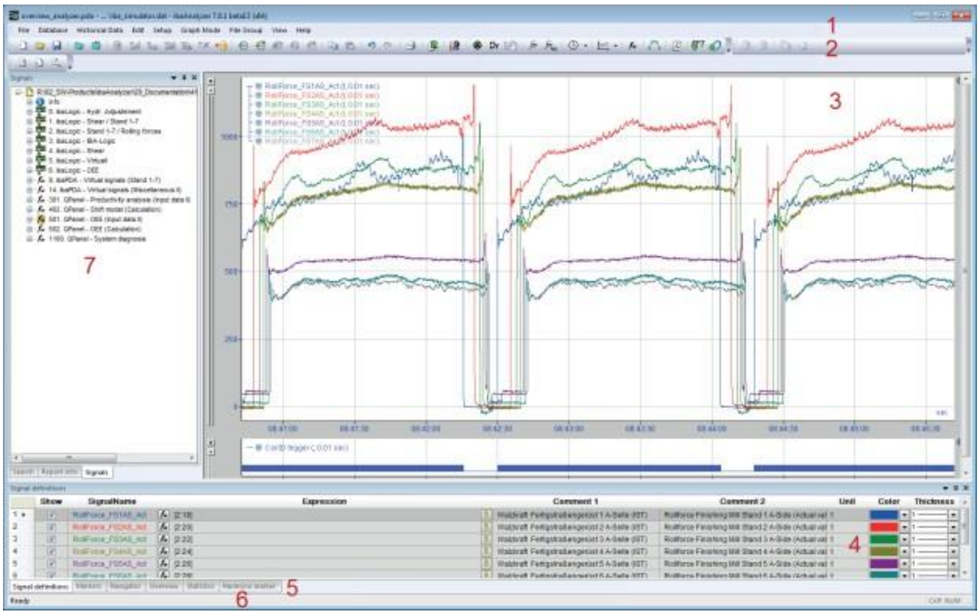
2 Theory

2.1 ibaAnalyzer

ibaAnalyzer is a module consisting of a large set of tools for analyzing historical data. The module offers time and frequency domain analysis. Expressions can be used to modify data series, calculate and evaluate inherent data. For more complex expressions, macros can be used. Essentially macros are corresponding to functions in conventional programming languages. This enables, what would otherwise, several expressions to be embedded within the same macro. As it is possible to reuse macros, it provides for a modular analysis.

Data to be analyzed is commonly stored in .dat files, .pdo files or in databases. ibaAnalyzer facilitates database connectivity, enabling data extraction within the same module. There are a number of other built-in functions available in ibaAnalyzer which aids in in-depth analysis of the complex calculations. The common add-ons that have been used in these projects are report generator, macros, expression builder, and data extractor. However, it has the potential to handle complex large amount of data.

In Figure 1 there is a menu bar at the top, tool bar right below menu bar, then there is signal window with the display of all signals under operation, there is signal definitions window which displays multiple functions for analysis and display. Additionally, there is a status bar which displays the status of the current operation, and in the end, there is a window which displays the signal tree, search function, report information, and analysis files. All the terms- have been mentioned in the Table 1 as well.



	Show	SignalName	Expression	Comment 1	Comment 2	Unit	Color	Thickness
1	☑	RollForce_FS1AS_Act	[2:18]	Walzkraft Fertigstraßengerüst 1 A-Seite (IST)	Rollforce Finishing Mill Stand 1 A-Side (Actual val	t	🟦	1
2	☑	RollForce_FS2AS_Act	[2:20]	Walzkraft Fertigstraßengerüst 2 A-Seite (IST)	Rollforce Finishing Mill Stand 2 A-Side (Actual val	t	🟥	1
3	☑	RollForce_FS3AS_Act	[2:22]	Walzkraft Fertigstraßengerüst 3 A-Seite (IST)	Rollforce Finishing Mill Stand 3 A-Side (Actual val	t	🟩	1
4	☑	RollForce_FS4AS_Act	[2:24]	Walzkraft Fertigstraßengerüst 4 A-Seite (IST)	Rollforce Finishing Mill Stand 4 A-Side (Actual val	t	🟫	1
5	☑	RollForce_FS5AS_Act	[2:26]	Walzkraft Fertigstraßengerüst 5 A-Seite (IST)	Rollforce Finishing Mill Stand 5 A-Side (Actual val	t	🟪	1
6	☑	RollForce_FS6AS_Act	[2:28]	Walzkraft Fertigstraßengerüst 6 A-Seite (IST)	Rollforce Finishing Mill Stand 6 A-Side (Actual val	t	🟦	1

FIGURE 1 STANDARD SCREEN DISPLAY

(1)	Menu Bar
(2)	Tool Bar
(3)	Recorder window/signal window
(4)	Signal table (Signal definition, Markers, Statistics) + Navigator + Harmonic markers + Overview of trend query
(5)	View tab of (4)
(6)	Satus bar
(7)	Signal tree + search function + report information + analysis files

TABLE 1 SCREEN TABS

2.2 ibaPDA

ibaPDA (Process Data Acquisition) is one of the central components of the iba-system. It is used in the manufacturing process to record and acquire high resolution processed data in an automated way. The variable recording options, auto-detect function, and client-server architecture are some key features of ibaPDA. It allows several signal recordings at the same time and individual signal configuration of live display. Figure 2 represents the general view of ibaPDA interface.

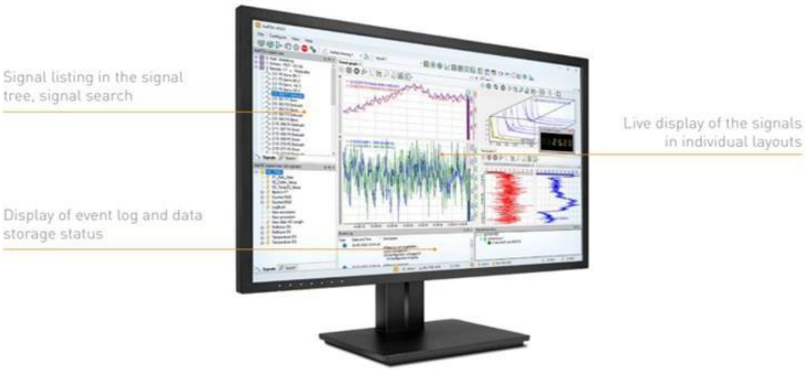

Signal listing in the signal tree, signal search
Live display of the signals in individual layouts
Display of event log and data storage status

FIGURE 2 IBAPDA INTERFACE.

ibaPDA offers Modular product concept that enables flexible configuration options and scalability. The data of ibaPDA could be analog and digital I/O signals, different sensory data, production data, product characteristic values etc. It also allows text signal acquisition for additional descriptive information. ibaPDA can be used in different applications such as troubleshooting, process data analysis, quality documentation, condition monitoring, test bench automation, and retrofit.

2.3 ibaCapture

ibaCapture is the video recording tool of the iba-system Figure 3. It allows continuous or event-triggered recording. In the iba-system, ibaCapture can be synchronized with ibaPDA to combine time-synchronized measured data and video data analysis. ibaCaptute can also integrate virtual cameras resulting from image processing by ibaVision.

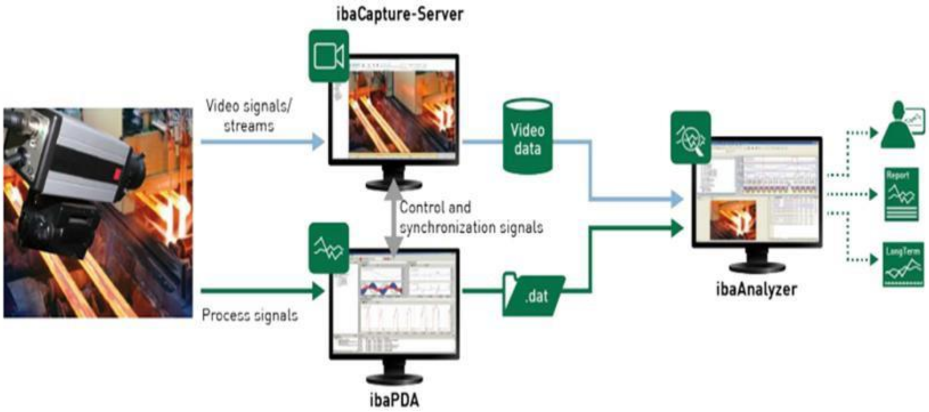


FIGURE 3 IBACAPTURE SERVER CONNECTION.

The measured data received from ibaPDA and video signal captured by ibaCapture can be read by ibaAnalyzer for process monitoring and analysis.

2.4 ibaVision

ibaVision is the image processing toolkit for the iba-system. The ibaVision program can convert visual data into numerical or logical values. The output of ibaVision can be a new image/video or logical data (numeric/text). The input can be an analog/digital signal from ibaPDA or image/video data from ibaCapture. The resultant data can be sent to ibaPDA while the image or video output can be displayed through ibaCapturer virtual cameras. Both signal and image information can be used bi-directionally, as can be seen in Figure 4.

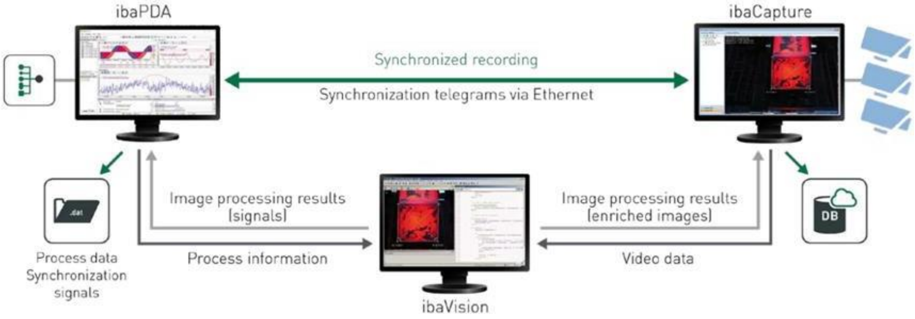

ibaPDA
ibaCapture
Synchronized recording
Synchronization telegrams via Ethernet
Image processing results (signals)
Image processing results (enriched images)
Process information
Video data
Process data Synchronization signals
DB
ibaVision

FIGURE 4 IBAVISION INTERFACE CONNECTION.

ibaVision supports both online and offline image processing. Either case requires an iba-module as a source to the ibaVision program. The online mode is intended for a 24/7 image processing task. Live image streaming can be evaluated with machine vision algorithms. In online mode the image processing must finish before the next frame is available. Offline processing does not support 24/7 image processing. It runs depending on the duration of the source file. The resultant data can be sent back to other iba analyzing tools. For image processing, ibaVision supports any of the three plug-in architectures mentioned below.

1. MVTechHalcon
2. Python 3.8
3. Custom plug-in in the .NET framework

ibaVision program runs under three fundamental procedures: Initialization, Main and Cleanup. The program starts with the Initialization procedure once the process begins. Initialization opens the handles of the process, for instance to Internal software, external hardware models, etc. The Main procedure is the procedure where the actual image processing takes place. After every execution ibaVision runs the Main procedure repeatedly to retrieve the result. Finally, the cleanup procedure is used to close the handles that are opened in the initialization procedure.

2.5 ibaDavis

A web-based platform utilized for displaying data recorded with the help of iba-systems and even it can display data recorded from any external database. It can help in identifying weak links with a number of different types of tiles available. These tiles then can be used further to have comparisons and detect the anomalies. The integration is with the iba-systems and has access to ibaHD-Server data. The data could also be displayed by building a connection with external databases.

Time and length-based signals from a measurement file in dat format could also be visualized in trends. In the following Figure 5 it can be observed that there are different routes to ibaDavis form different modules either it could be connected via ibaPDA or it could be connected via ibaHD-Server by using KPI’s. Additionally, there is another path of making connection in which offline events are analyzed with the help of ibaAnalyzer, automated with the help of ibaDatcoordinator and uploaded in databases with the help of data extractor into ibaDavis for long term analysis.

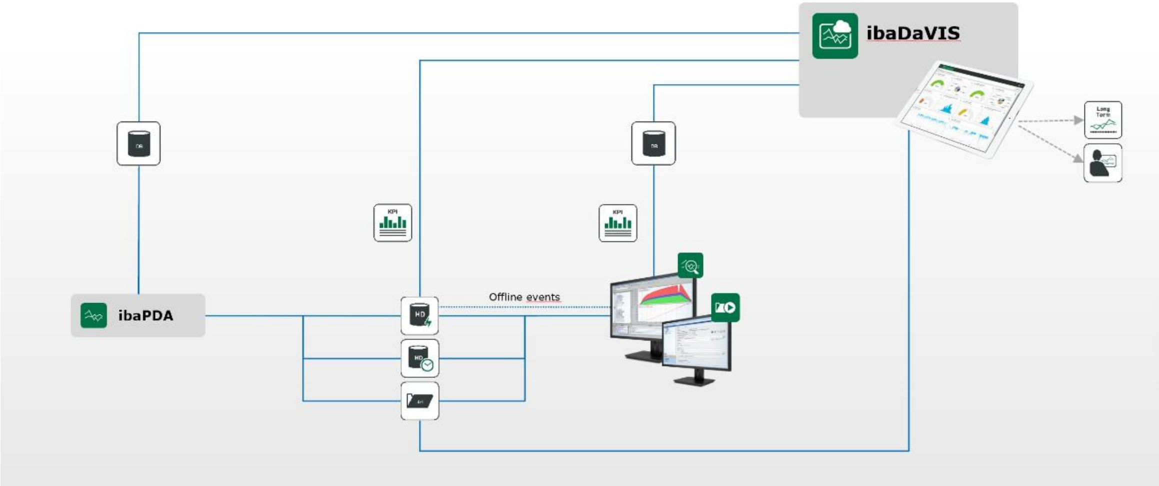


FIGURE 5 ROUTES TO IBADAVIS

3 Process and results

Three case studies are hereby presented. Full reports can be found in diva homepage [1].

3.1 Case study 1 – Can iba detect the next compressor failure?

3.1.1 Background

Production of steel powder is done by atomization of molten steel stream to rapidly cool and fragment the steel flow into powder. Nitrogen gas is used for the atomization process. The nitrogen gas is compressed to desired pressure and injected via nozzles to intersect the molten steel flow. As nitrogen gas is an inert gas it can be recycled by cooling and filtering the gas before it is routed back to the inlet side of the compressor. Thus, the compressor constitutes a vital part in the production of steel powder, as it allows for gas recycling and controllable gas pressure.

In 2021 a major compressor failure occurred at Erasteel Kloster AB, located in Söderfors, which caused large production losses. Therefore, this thesis investigates what modules and functionality iba-systems have to offer. Process and machine signals are studied to assess both their utility in predicting machine failure and relevant iba-modules for the predictive maintenance purposes, based on a literature review.

3.1.2 Process and results

Analyses are done in ibaAnalyzer and Matlab.

Production of steel powder is done under various process conditions. To limit the scope of this thesis and to uphold secrecy by not extracting more data than necessary, only selected production cycles were chosen for analysis. ibaAnalyzer offers the possibility to make a database query with conditions. By specifying the conditions, only desired data series are extracted from the database.

By studying process and machine parameters the objective was to analyze the need for maintenance, i.e., condition-based monitoring. As each signal has features, the first step was to find significant signal features, i.e., features showing distinct correlation with machine failure or process anomalies. Signal features were assessed and ranked. A comparison can be seen in Figure 6, in which it is evident that standard deviation can distinguish two anomalies, whereas the maximum value of a data series does not provide useful information. Similar comparisons were done between numerous signals and their features.

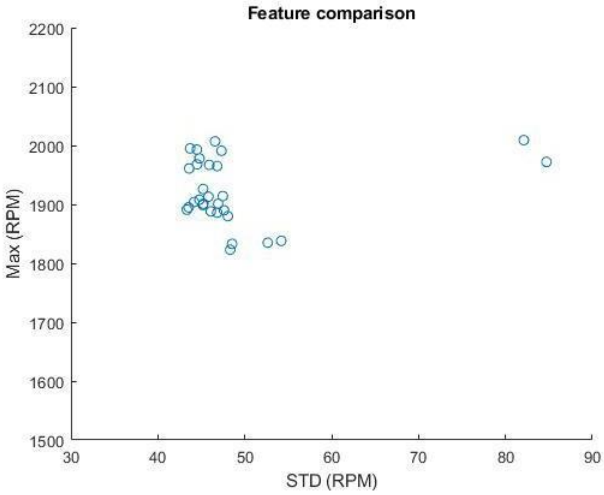


Figure 6 Feature comparison, standard deviation versus maximum value.

Based on selected signal features, anomaly detection is implemented in ibaAnalyzer.

Merging of multiple production cycles and feature calculation can be seen in Figure 7.

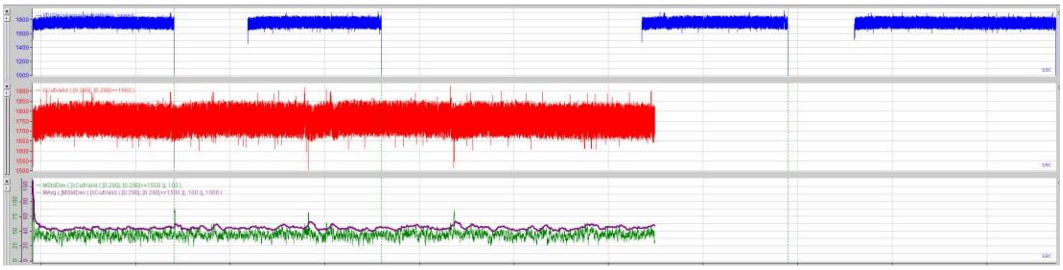

1800
1600
1400
1200
1000
XCutValid ([0:290], [0:290]>=1500)
1900
1850
1800
1750
1700
1650
1600
1550
1500
MStdDev ([XCutValid ([0:290], [0:290]>=1500)], 100)
MAvg ([MStdDev ([XCutValid ([0:290], [0:290]>=1500)], 100)], 1000)
100
75
50
25
0

Figure 7 Appended and concatenated data series and the moving standard deviation.

Production cycles throughout the span of 2.5 years were concatenated into one data series. Two anomalies were detected throughout the selected span. Anomalies can be seen in Figure 8. The first anomaly marked as 1 is related to an update of software in the compressor controller whereas the second anomaly, marked as 2, is due to faulty safety valves.

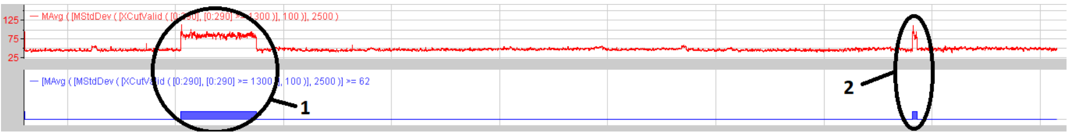

MAvg ([MStdDev ([XCutValid ([0:290], [0:290] >= 1300)], 100)], 2500)
125
75
25
[MAvg ([MStdDev ([XCutValid ([0:290], [0:290] >= 1300)], 100)], 2500)] >= 62
1
2

Figure 8 Anomaly detection and alarm generation.

Signal processing is done using expressions. More complex calculations are embedded within macros, i.e., corresponding to functions in conventional programming languages. Macros provides modularity for the analysis file, enabling reuse of created functions. This is useful when multiple mathematical and filtering operations are computed.

3.1.3 Conclusion and future improvements

It is possible to detect anomalies using ibaAnalyzer. Although, anomalies detected in this thesis correspond to extreme conditions. For more sophisticated anomaly detection and remaining useful life estimators, complementary measurements are needed. Suggested improvements focus on vibration measurements as this is a well-documented and proven method. Additionally, lubrication oil measurements using ferrography is suggested to detect machine degradation.

The need for metadata should be highlighted, as conditioned queries require distinct events for start and stop triggers of data extraction from the ibaHD server. By being able to sort data based on production recipes, guaranteeing that correct data is extracted, analysis would be simplified.

If vibration sensors are installed to monitor the compressor, ibaInSpectra is recommended as an analysis tool. The module has the option for preprocessing signals, calculating standard statistical features and configuring alarms. The ability to design low-, high-, bandpass and stop filters makes this module useful for vibration measurements which are analyzed in the frequency domain.

ibaInCycle was initially not considered for this thesis since an entire production cycle has to finish before ibaInCycle generates a response. If machine degradation occurs over a long period of time, i.e., many production cycles, ibaInCycle is suggested as a tool for future anomaly detection as the process is cyclic.

3.2 Case study 2 – Time efficient harvesting of data to find an optimum production window with the help of iba sofrware

3.2.1 Background

There is production of tubes being done in the Tube mill at Ovako AB and there is immense amount of data available that is hard to utilize in a time-efficient way. The tube manufacturing process goes through a number of steps, the step that is under consideration for this project is the third step in the manufacturing process with the name Piercing Mill which basically records the values of moment for each tube.

Currently the statistical properties of these tubes are calculated with the help of MS excel and their graphs are being plotted also in MS excel and then all the analysis is being carried out with the help of MS excel. The data is exported from ibaAnalyzer via ibaHDserver into MS excel and the analyzed accordingly manually in the excel which is quite time-consuming task and not time efficient.

3.2.2 Process and results

There are a number of steps that are being carried out in order to complete the process from retrieving data from HD-server, analyzing it, then encapsulating that data in the form of macros, then uploading that data into the database with the help of data extractor and in the end displaying that data with the help of ibaDavis.

The tubes have been extracted out individually with the help of respective expressions built in expression builder. In,

Figure 9 after snipping out all the tubes as a separate signal, the tubes are basically overlapped on top of each other in order to have a comparison. In this figure there are total 5 tubes that have been extracted out to make comparison. The signals are time-based signals where intervals are measured in seconds.

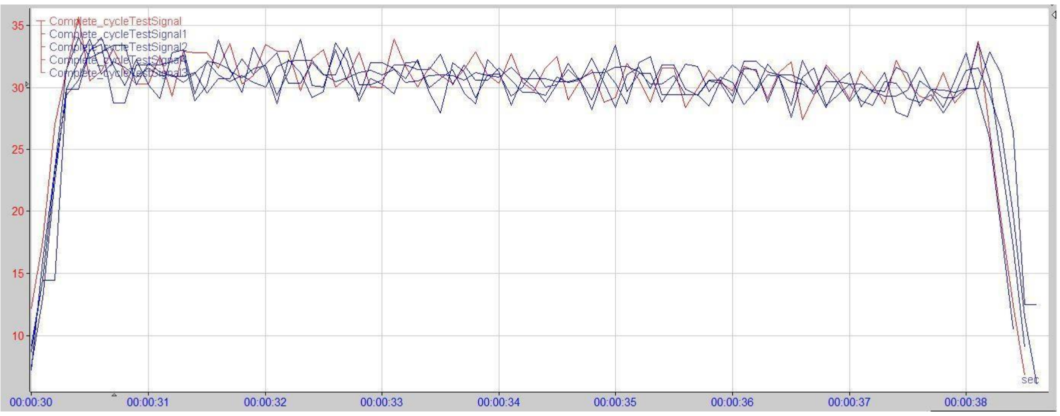


FIGURE 9 COMBINED TUBES COMPLETE CYCLES

Additionally, in the same way all the signals will be calculated for the stats cycle in the Figure 10, then overlapping them on top of each other for comparison purposes, and in the end, it will be used to calculate the statistical properties. The signals have been extracted here with a gap of 1 sec after the peak starts and before the peak ends in order to have the intermediate portion of the signal. These signals will aid in calculating all the respective statistical values and then displaying them in the web based platform.

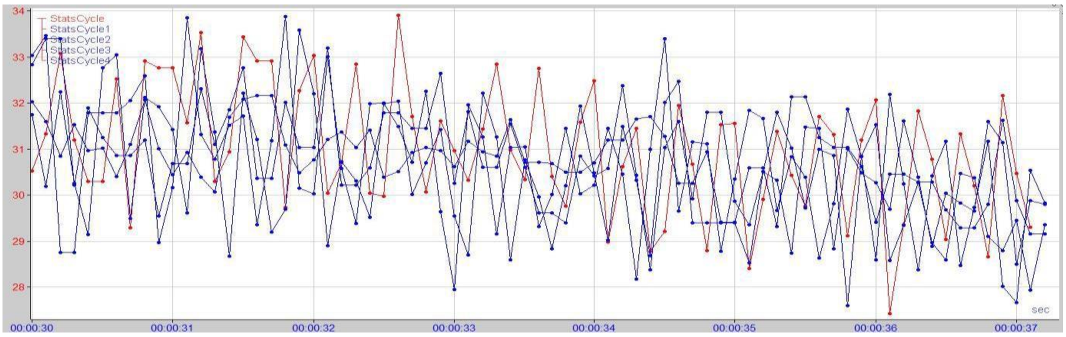
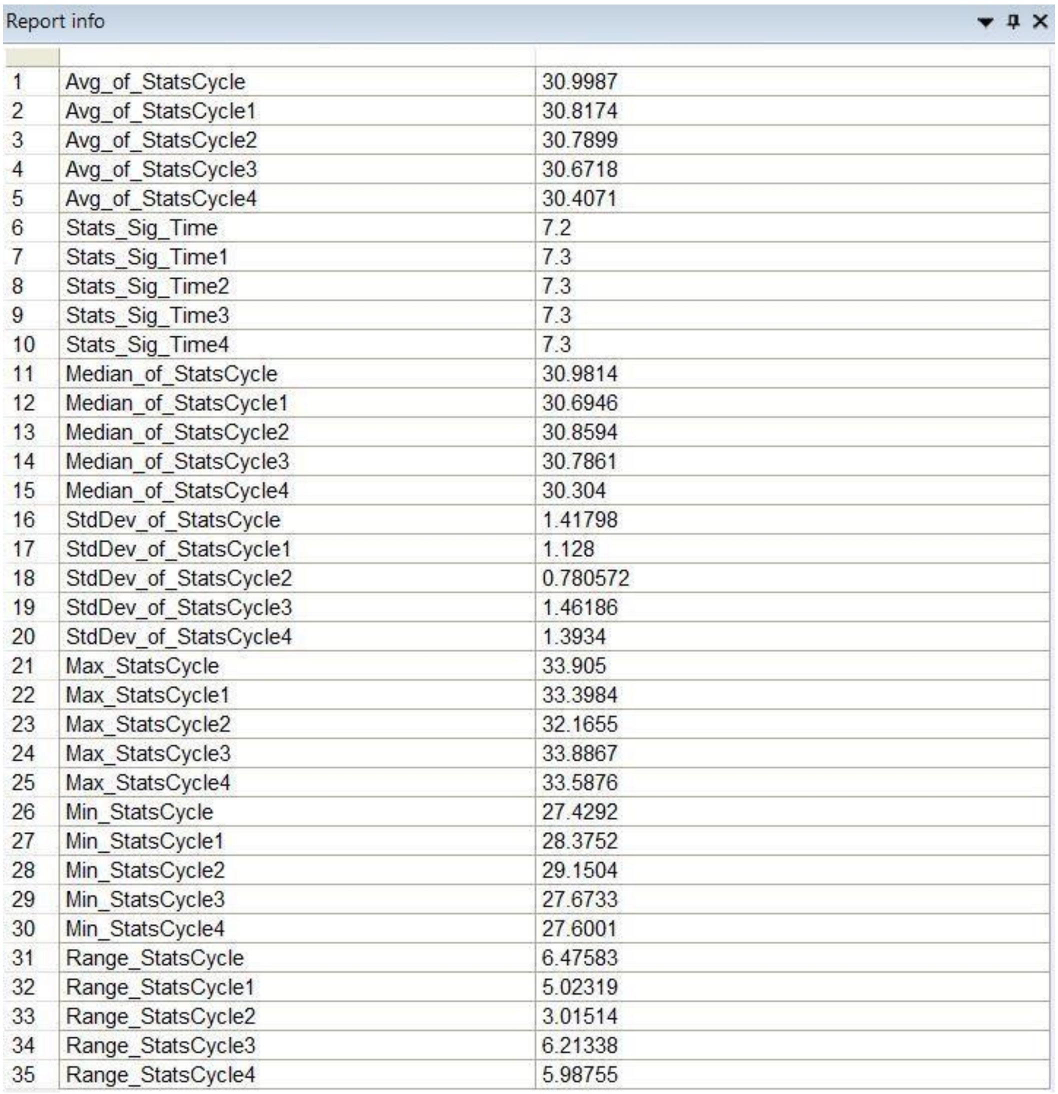


Figure 10 Stats cycle values

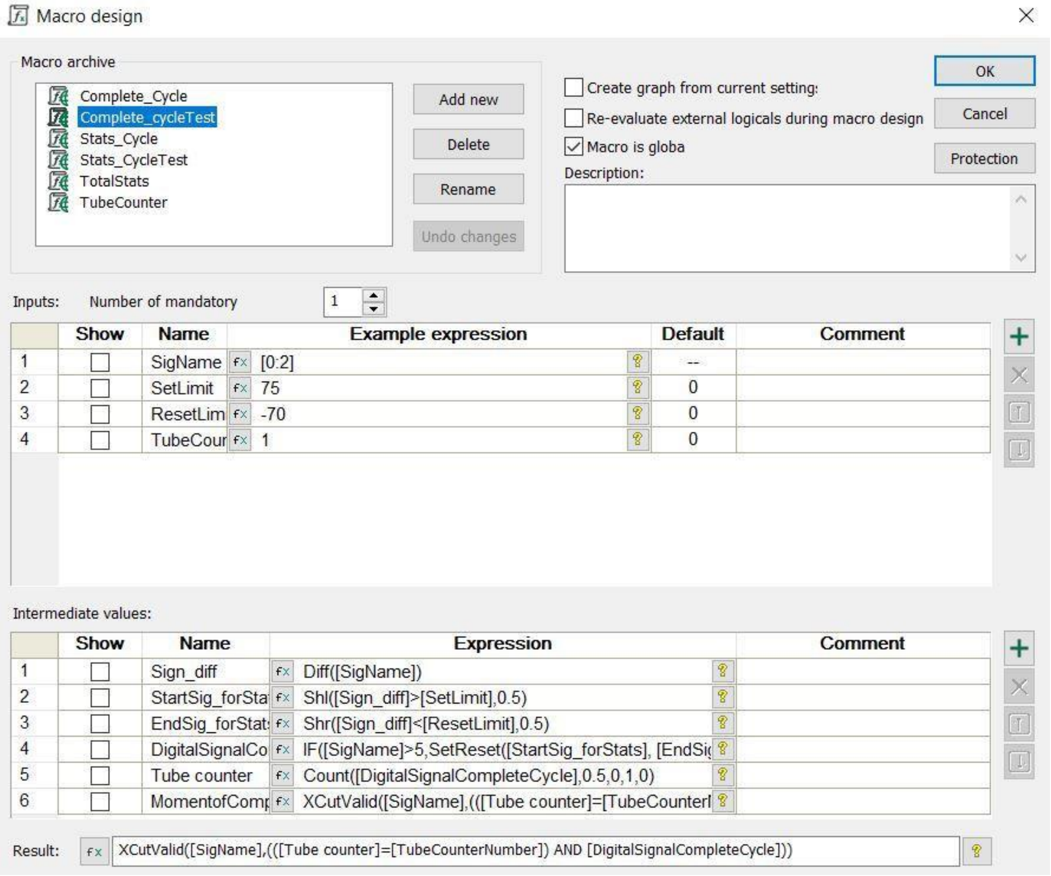
All the values calculated in Figure 11 will then be displayed in the report generator as well for a quick display of statistics. There are 5 values recorded for each value of average, time, median, standard deviation, maximum, minimum, and range. The difference in the values could be noticed and can therefore help in having an optimized way of analysis and ultimately help in harnessing the production growth.



1	Avg_of_StatsCycle	30.9987
2	Avg_of_StatsCycle1	30.8174
3	Avg_of_StatsCycle2	30.7899
4	Avg_of_StatsCycle3	30.6718
5	Avg_of_StatsCycle4	30.4071
6	Stats_Sig_Time	7.2
7	Stats_Sig_Time1	7.3
8	Stats_Sig_Time2	7.3
9	Stats_Sig_Time3	7.3
10	Stats_Sig_Time4	7.3
11	Median_of_StatsCycle	30.9814
12	Median_of_StatsCycle1	30.6946
13	Median_of_StatsCycle2	30.8594
14	Median_of_StatsCycle3	30.7861
15	Median_of_StatsCycle4	30.304
16	StdDev_of_StatsCycle	1.41798
17	StdDev_of_StatsCycle1	1.128
18	StdDev_of_StatsCycle2	0.780572
19	StdDev_of_StatsCycle3	1.46186
20	StdDev_of_StatsCycle4	1.3934
21	Max_StatsCycle	33.905
22	Max_StatsCycle1	33.3984
23	Max_StatsCycle2	32.1655
24	Max_StatsCycle3	33.8867
25	Max_StatsCycle4	33.5876
26	Min_StatsCycle	27.4292
27	Min_StatsCycle1	28.3752
28	Min_StatsCycle2	29.1504
29	Min_StatsCycle3	27.6733
30	Min_StatsCycle4	27.6001
31	Range_StatsCycle	6.47583
32	Range_StatsCycle1	5.02319
33	Range_StatsCycle2	3.01514
34	Range_StatsCycle3	6.21338
35	Range_StatsCycle4	5.98755

Figure 11 Report generator Stats values

Moreover, in Figure 12 Macros will be generated where it will have all the values encapsulated and will be used in an optimized way where the user does not have to perform all the calculations repeatedly. The macros can be exported to be used by other users, additionally when the user mark it as globa then it could be used by the same user for other analysis files as well. It consists of a set of inputs, intermediate values, and a result part where the foremost function is added to perform the calculation.



Macro archive

- Complete_Cycle
- Complete_cycleTest
- Stats_Cycle
- Stats_CycleTest
- TotalStats
- TubeCounter

□ Create graph from current setting:
□ Re-evaluate external logicals during macro design
☑ Macro is globa

Description:

Inputs: Number of mandatory 1

	Show	Name	Example expression	Default	Comment
1	□	SigName	[0:2]	--	
2	□	SetLimit	75	0	
3	□	ResetLim	-70	0	
4	□	TubeCour	1	0	

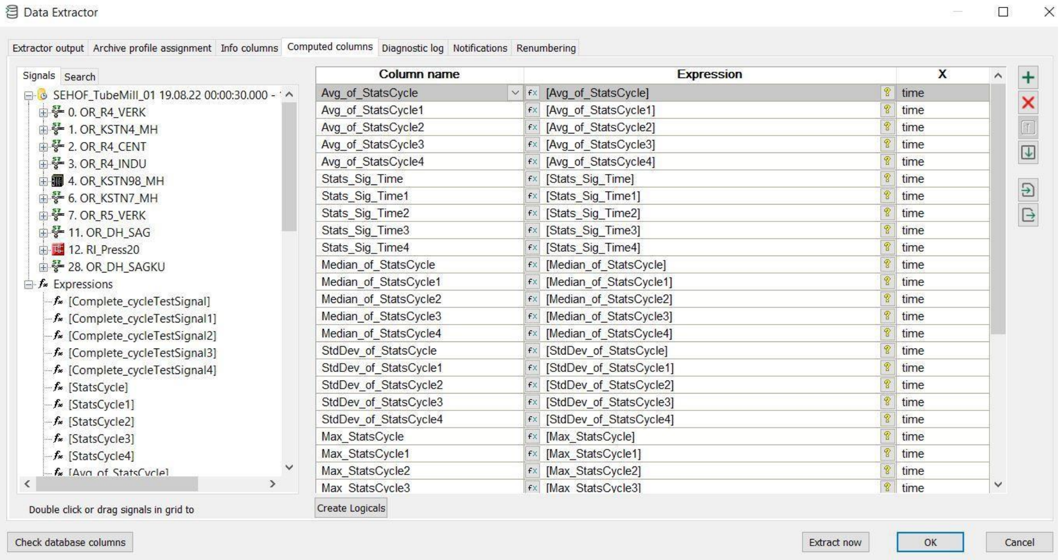
Intermediate values:

	Show	Name	Expression	Comment
1	□	Sign_diff	Diff([SigName])	
2	□	StartSig_forSta	Shl([Sign_diff]>[SetLimit],0.5)	
3	□	EndSig_forStat:	Shr([Sign_diff]<[ResetLimit],0.5)	
4	□	DigitalSignalCo	IF([SigName]>5,SetReset([StartSig_forStats], [EndSi	
5	□	Tube counter	Count([DigitalSignalCompleteCycle],0.5,0,1,0)	
6	□	MomentofComp	XCutValid([SigName],(([Tube counter]=[TubeCounterI	

Result: XCutValid([SigName],(([Tube counter]=[TubeCounterNumber]) AND [DigitalSignalCompleteCycle]))

FIGURE 12 MACROS

In Figure 13 the Data extractor will then assist in uploading data into the database. The user simply has to select the respective signals under the tab of computed columns and then extract the respective information by developing the connection with the database, the developed connection with the database will then be used with SQL Server Management Studio (SSMS) to read the data and then will help in exporting it into the ibaDAVIS for the display.



Data Extractor

Extractor output | Archive profile assignment | Info columns | Computed columns | Diagnostic log | Notifications | Renumbering

Signals | Search

- SEHOF_TubeMill_01 19.08.22 00:00:30.000 -
  - 0. OR_R4_VERK
  - 1. OR_KSTN4_MH
  - 2. OR_R4_CENT
  - 3. OR_R4_INDU
  - 4. OR_KSTN98_MH
  - 6. OR_KSTN7_MH
  - 7. OR_R5_VERK
  - 11. OR_DH_SAG
  - 12. RI_Press20
  - 28. OR_DH_SAGKU
- Expressions
  - [Complete_cycleTestSignal]
  - [Complete_cycleTestSignal1]
  - [Complete_cycleTestSignal2]
  - [Complete_cycleTestSignal3]
  - [Complete_cycleTestSignal4]
  - [StatsCycle]
  - [StatsCycle1]
  - [StatsCycle2]
  - [StatsCycle3]
  - [StatsCycle4]
  - [Avg_of_StatsCycle]

Column name	Expression	X
Avg_of_StatsCycle	[Avg_of_StatsCycle]	time
Avg_of_StatsCycle1	[Avg_of_StatsCycle1]	time
Avg_of_StatsCycle2	[Avg_of_StatsCycle2]	time
Avg_of_StatsCycle3	[Avg_of_StatsCycle3]	time
Avg_of_StatsCycle4	[Avg_of_StatsCycle4]	time
Stats_Sig_Time	[Stats_Sig_Time]	time
Stats_Sig_Time1	[Stats_Sig_Time1]	time
Stats_Sig_Time2	[Stats_Sig_Time2]	time
Stats_Sig_Time3	[Stats_Sig_Time3]	time
Stats_Sig_Time4	[Stats_Sig_Time4]	time
Median_of_StatsCycle	[Median_of_StatsCycle]	time
Median_of_StatsCycle1	[Median_of_StatsCycle1]	time
Median_of_StatsCycle2	[Median_of_StatsCycle2]	time
Median_of_StatsCycle3	[Median_of_StatsCycle3]	time
Median_of_StatsCycle4	[Median_of_StatsCycle4]	time
StdDev_of_StatsCycle	[StdDev_of_StatsCycle]	time
StdDev_of_StatsCycle1	[StdDev_of_StatsCycle1]	time
StdDev_of_StatsCycle2	[StdDev_of_StatsCycle2]	time
StdDev_of_StatsCycle3	[StdDev_of_StatsCycle3]	time
StdDev_of_StatsCycle4	[StdDev_of_StatsCycle4]	time
Max_StatsCycle	[Max_StatsCycle]	time
Max_StatsCycle1	[Max_StatsCycle1]	time
Max_StatsCycle2	[Max_StatsCycle2]	time
Max_StatsCycle3	[Max_StatsCycle3]	time

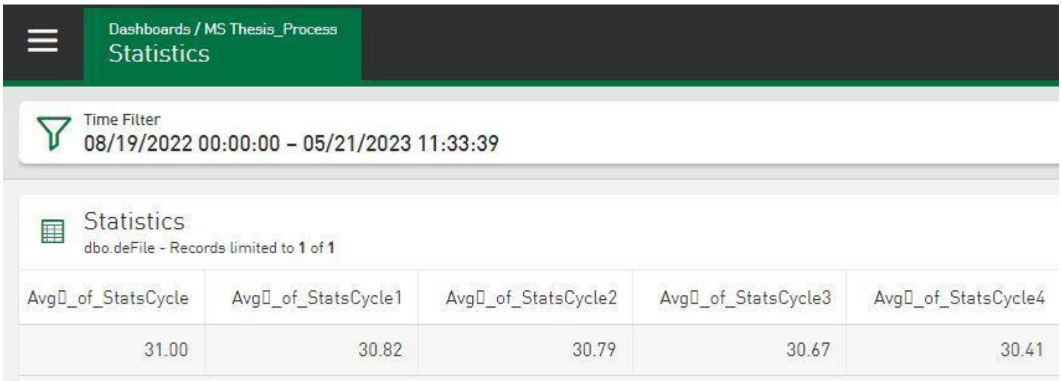
Double click or drag signals in grid to

Create Logicals

Check database columns | Extract now | OK | Cancel

FIGURE 13 DATA EXTRACTOR

In Figure 14, and Figure 15 all the statistical values calculated have been displayed in the ibaDavis in 2 different tile types. The figure 14 displays the values in the form of tables of all statistical values and in the same the figure 15 displays all values in the form of grid where every single value is displayed exclusively.



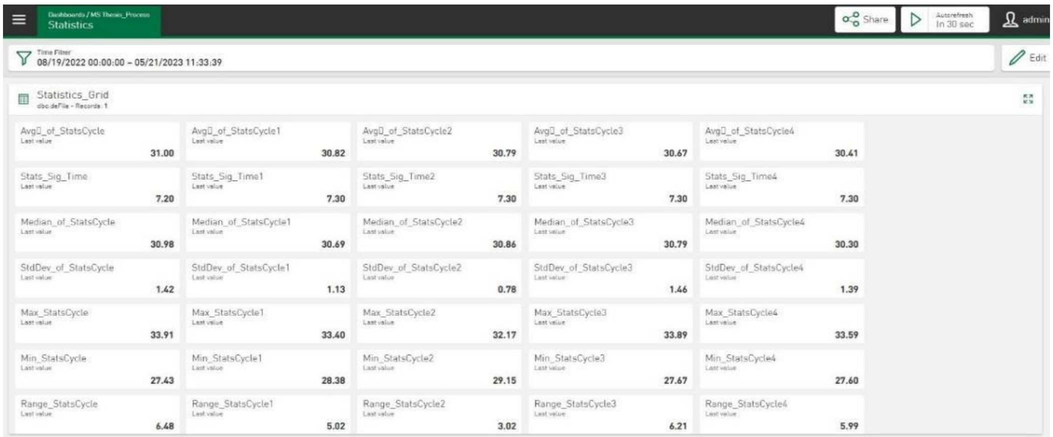
Dashboards / MS Thesis_Process
Statistics

Time Filter
08/19/2022 00:00:00 – 05/21/2023 11:33:39

Statistics
dbo.deFile - Records limited to **1** of **1**

Avg□_of_StatsCycle	Avg□_of_StatsCycle1	Avg□_of_StatsCycle2	Avg□_of_StatsCycle3	Avg□_of_StatsCycle4
31.00	30.82	30.79	30.67	30.41

FIGURE 14 TABLES_IBADAVIS



Dashboards / MS Thesis_Process
Statistics

Share | Autorefresh In 30 sec | admin

Time Filter
08/19/2022 00:00:00 – 05/21/2023 11:33:39

Edit

Statistics_Grid
dbo.deFile - Records: 1

Avg□_of_StatsCycle Last value 31.00	Avg□_of_StatsCycle1 Last value 30.82	Avg□_of_StatsCycle2 Last value 30.79	Avg□_of_StatsCycle3 Last value 30.67	Avg□_of_StatsCycle4 Last value 30.41
Stats_Sig_Time Last value 7.20	Stats_Sig_Time1 Last value 7.30	Stats_Sig_Time2 Last value 7.30	Stats_Sig_Time3 Last value 7.30	Stats_Sig_Time4 Last value 7.30
Median_of_StatsCycle Last value 30.98	Median_of_StatsCycle1 Last value 30.69	Median_of_StatsCycle2 Last value 30.86	Median_of_StatsCycle3 Last value 30.79	Median_of_StatsCycle4 Last value 30.30
StdDev_of_StatsCycle Last value 1.42	StdDev_of_StatsCycle1 Last value 1.13	StdDev_of_StatsCycle2 Last value 0.78	StdDev_of_StatsCycle3 Last value 1.46	StdDev_of_StatsCycle4 Last value 1.39
Max_StatsCycle Last value 33.91	Max_StatsCycle1 Last value 33.40	Max_StatsCycle2 Last value 32.17	Max_StatsCycle3 Last value 33.89	Max_StatsCycle4 Last value 33.59
Min_StatsCycle Last value 27.43	Min_StatsCycle1 Last value 28.38	Min_StatsCycle2 Last value 29.15	Min_StatsCycle3 Last value 27.67	Min_StatsCycle4 Last value 27.60
Range_StatsCycle Last value 6.48	Range_StatsCycle1 Last value 5.02	Range_StatsCycle2 Last value 3.02	Range_StatsCycle3 Last value 6.21	Range_StatsCycle4 Last value 5.99

FIGURE 15 GRID_IBADAVIS

3.2.3 Conclusion and future improvements

To conclude, the project is fundamentally the interoperability of systems where different systems are integrated and the analyzed data is then fused together for time efficient analysis. The tasks have been performed under the banner of iba-systems. The data is acquired first from ibaHD-Server, then analyzed in ibaAnalyzer with the help of different expressions, after that the data is then exported into an external database with the help of ibaAnalyzer data extractor. The data is then configured and exported into ibaDavis with the help SQL Server Management Studio (SSMS) and in the end displayed in ibaDavis.

The future prospects look like having vectors that can combine the signals in a single scale and the text signal could help in adding the relevant information of the signal. Moreover, ibaAnalyzerdB could be used to export data directly from ibaHD-Server into ibaDavis, external databases and from external databases into HD-server. ibaDatcoordinatordB could also do the job of ibaAnalyzerdB and along with that it could automate the jobs as well which could be a possible improvement in this project for the future.

3.3 Case study 3 - Vision system to automatically generate time measurement

3.3.1 Background

As part of the manufacturing process of super duplex stainless steel (SDSS) at the Forging shop of Alleima AB in Sandviken, materials are heated in a furnace and then quenched in a quench tank. There is a temperature sensor in the outlet of the cooler. Since the cooling system is dynamic (has inertia), the temperature sensor registers a temperature rise in the water tank with a delay. Considering the rapid change in mechanical properties of hot materials, it is important to determine when the material leaves the furnace and when cooling begins in a more precise way. This project proposed a vision system-based automatic timestamp measurement technique to facilitate the need. The proposed mechanism first detects the hot metal bars and then uses an event-based technique to determine when the detected object drops into the water tank for quenching. Finally, the obtained timestamp has been compared with the temperature sensor measurement to determine the delay. This project uses an iba-system to perform the entire procedure. The video sequence is collected using ibaCapture and the image processing took place in ibaVision. The measured timestamp is transferred to ibaPDA for further analysis.

3.3.2 Process and results

The production of SDSS in the forging shop follows several steps. Heat treatment is one of them. During the heat treatment, the steel is heated to a high temperature, held at an elevated temperature for some period, and then quenched in the water tanks. The whole process is captured by the camera used in this project. The camera view in Figure 16 shows there are two water tanks located in the forging shop. This project is only interested in when the cooling begins, therefore two water tanks have been declared as the Region of Interest (ROI).

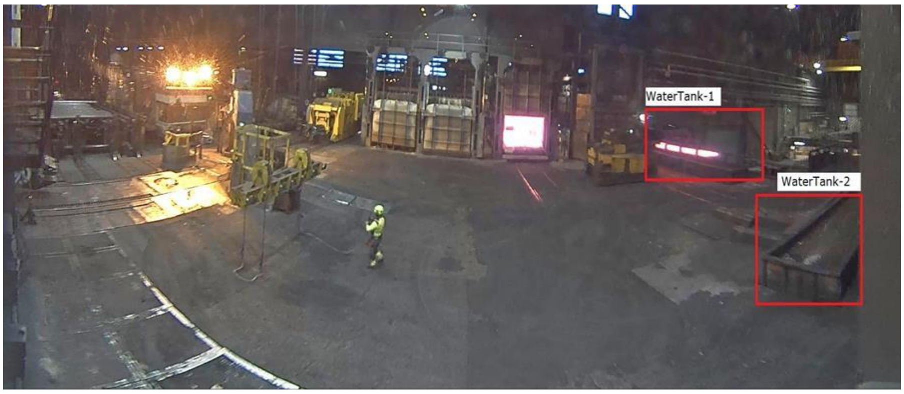


FIGURE 16 REGION OF INTEREST.

Next an object detection technique has been applied to identify the hot material inside ROI. The object detection technique follows a classical image processing approach. A color-based thresholding has been applied to detect hot metal. The collected video frame is in Red, Green, Blue (RGB) color format. A gaussian filter is used to smoothen the image and then it is converted to Hue, Saturation, Value (HSV) color format. Using the HSV color map the range of each component has been selected to detect the hot metal bar. Once the bar is detected some morphological operations has been applied to retain the actual shape of the metal bar.

Figure 17 (A) is the RGB image, (B) blurred image, (C) shows the image in HSV color form.

Figure 18 (A) represents the thresholded image after applying a mask using HSV

color specification for hot metal, figure (B) & (C) show the result of morphological operation.

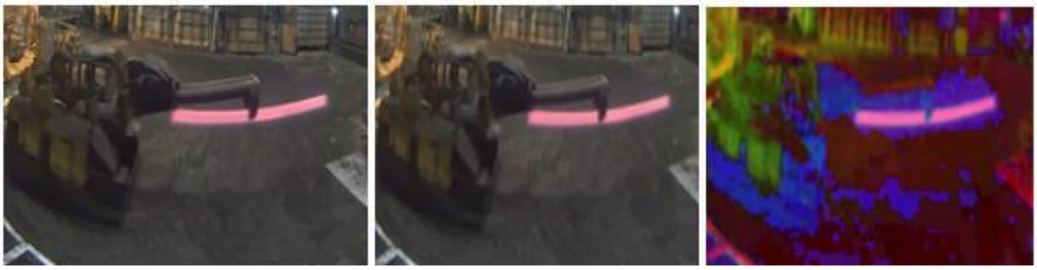


FIGURE 17 (A) BGR IMAGE (B) BLURRED IMAGE (C) HSV IMAGE

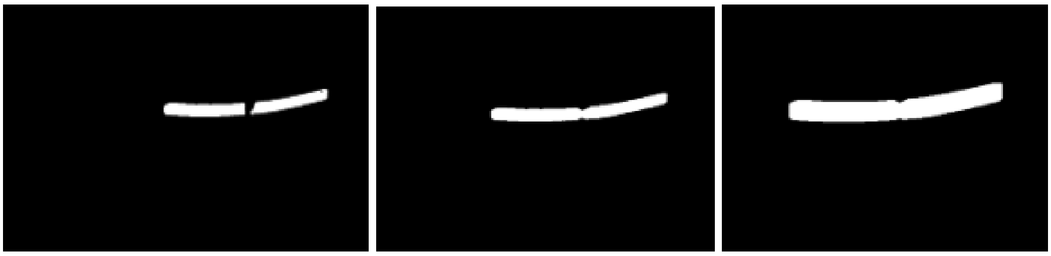


FIGURE 18 (A) THRESHOLDED IMAGE (B) MORPHOLOGICAL CLOSING (C) DILATION

Once the object is detected the next task is to track the object and find when the cooling begins. The algorithm detects the hot metal bar properly when it enters ROIs and draws a contour line around the bar. To detect the time stamp, a flag has been set to FALSE at the beginning of the algorithm. When the algorithm detects the contour line inside the ROI, that is, when the length of the contour is greater than 0, the flag will turn to TRUE. While the flag remains TRUE and the contour length becomes 0 again, it will display the timestamp and reset the condition to its previous state.

In the next step, the prepared Python scripts with necessary libraries have been uploaded to the ibaVision. The input and output variables have been declared in the ibaVision input and output modules respectively. The required timestamp measurement is configured as a digital signal in the ibaVision output module. Finally, the digital signal is sent to ibaPDA and combining the time measurement of the vision system with temperature sensor the delay has been calculated as seen in Table 2 time measurement comparison.

Vessel	Vision System Time Measurement	Temperature Sensor Time Measurement	Delay (s)
Water Tank 1	20:14:44	20:15:00	16
Water Tank 1	2:24:20	2:24:36	16
Water Tank 1	12:06:46	12:07:05	19
Water Tank 2	4:30:00	4:30:08	8
Water Tank 2	4:30:01	4:30:09	8
Water Tank 2	7:05:15	7:05:21	6

TABLE 2 TIME MEASUREMENT COMPARISON.

To conclude, it has been observed that the temperature sensor at water tank 2 registers less delay compared to the temperature sensor at water tank 1. The average delay encountered in water tank 2 is approximately 7s, whereas it is around 17s for water tank 1.

3.3.3 Conclusion and future improvements

The proposed machine vision algorithm assures a promising result to realise the specific need of this project. It shows a possible way to determine timestamp using a vision system automatically. The object detection method used in this project follows a conventional image processing technique. A color-based object thresholding along with some morphological operation leads towards a successful detection. The use of the iba-system along with machine vision opens another dimension for this project. The new plug-in concept of IbaVision with Python 3.8 ensures a desired outcome using a vision system. In addition, the interface connection of ibaVision with other iba-modules works in an efficient way. A future improvement to the proposed system could be a dimension-based detection technique along with color scheme method for a more accurate detection. While the current project has focused on the traditional machine vision approach, a deep learning or machine learning approach could be another future research interest.

4 Experiences using iba - Result

Here the benefits and disadvantages of iba-systems are discussed, according to student experience.

4.1 Positive aspects of using iba-systems

Signal processing is easily done in ibaAnalyzer. The ability to have a visual overview of signals and modifications applied to said signals gives an intuitive understanding of the operations compared to softwares such as Matlab and others.

Iba-systems are a great source for data fusion, data integration and interoperability of systems. It aids in performing several tasks using different modules dedicated to individual specific tasks. It can perform tasks ranging from data acquisition, data recording and storage, data analysis, integration to external databases, automating the tasks, and in the end displaying them in a web-based platform.

In terms of image processing ibaVision assures promising outcomes. The plug-in concept of ibaVision with Python 3.8 gives the opportunity to use a wide range of available libraries of Python and build advanced applications. The interface connection between different iba-modules are found to be straightforward and seamless.

4.2 Negative aspects of using iba-system

The lack of community is an issue. Minor mistakes can take a long time to fix when one is working with iba. Conventionally one can search for specific problems to gain information on how to solve the issue at hand. This is not the case with iba.

There is a huge potential in the iba-systems. However, an overview of all modules is needed as it is hard to grasp which modules may be used for a particular project. There are some distributions available that can simplify the package management and deployment. It can expectantly

The modular concept of iba-system is helpful to facilitate specific user needs. However, it also creates the dependencies on different iba-modules. For example, ibaVision cannot run standalone. It requires other iba-modules to receive and send the processed data. In this case ibaVision only creates an interface connection within the system.

5 Discussion

The joint project setup has been a fun and rewarding arrangement in which students can support one another. This is especially important considering the lack of community, from which one often gets guidance from during conventional projects i.e., programming. It has also proven to be efficient as students share experiences instead of performing redundant work.

Having two supervisors you are frequently able to have your supervision needs met. This setup has also proven to be a beneficial experience, as the supervisors bring different areas of expertise to the table. The university professor concentrates on the academic aspect of the thesis, while the company supervisor possesses expertise in the process you are studying.

Companies using iba should contribute to a community regarding iba-systems. This should not only be seen as helping future students in iba related projects, but rather as a way of training and helping employees. Creation of a joint homepage, to which one can upload scripts, codes and other examples, would increase the productivity of employees as well as minimizing the time spent on training.

Furthermore, the possibility to ask questions in public (open for selected companies) benefits all members of this collaboration. Abovementioned community would be the catalyst that would accelerate the goal of sharing experience between members of Jernkontoret.

6 References

[1] "diva-portal," 02 06 2023. [Online]. Available: https://www.diva-portal.org/smash/search.jsf?dswid=-7630. [Accessed 02 06 2023].

The Swedish Iron and Steel Producers’ Association

Since its foundation in 1747, Jernkontoret has been owned jointly by the Swedish iron and steel companies. Jernkontoret represents Sweden’s iron and steel industry on issues that relate to trade policy, research and education, standardisation, energy, the environment and sustainability as well as transportation issues. Jernkontoret also manages the joint Nordic research on steel. In addition, Jernkontoret draws up statistical information relating to the industry and carries out research into the history of mining and metallurgy.

Jernkontoret