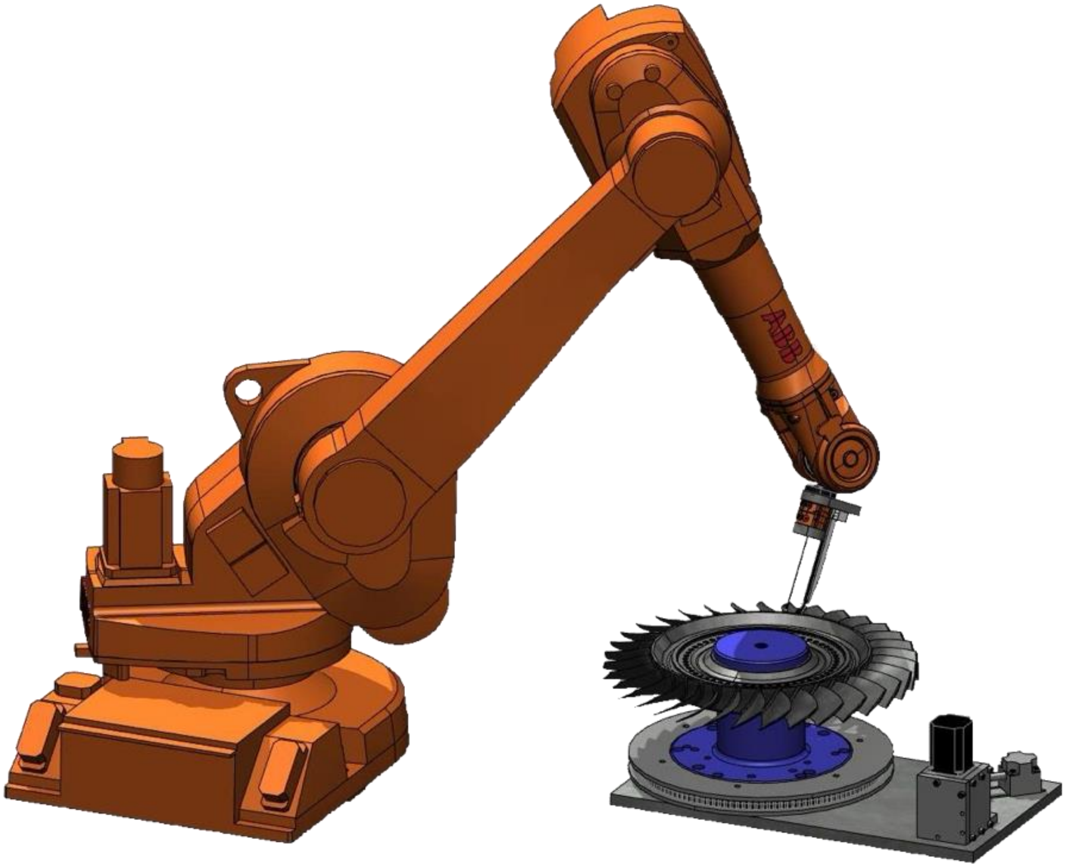
Blisk Inspection System



Nicholas Hollan & Kevin MacIntosh

Advised By: Professor Craig Putnam & Professor Kenneth

Blisk Inspection System

A Major Qualifying Project
Submitted to the Faculty of Worcester Polytechnic Institute
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
in cooperation with
GE Aviation in Hooksett, New Hampshire

Submitted By:
Nicholas Hollan
Kevin MacIntosh

Project Advisors:
Craig Putnam
Kenneth Stafford

Table of Contents

Table of Figures

Blisk Inspection System

Nicholas Hollan, Kevin MacIntosh
Advised By: Professor Craig Putnam, Professor Kenneth Stafford

***Abstract*—The goal of this project is to automate the process of inspecting blisk root fillets for GE Aviation. The Blisk Inspection System utilizes an ABB robotic arm equipped with a custom end effector, actively communicating with a programmable logic controller and custom turntable. This system is primarily controlled through a windows application, serving as a user interface while also managing computer vision and logging inspection results. Quality analysis of bladed disk (blisk) root fillets is currently being performed by hand, requiring excessive labor and consumable costs. This system will serve as a suggested solution to reduce overall cost of production and increase factory output.**

I. INTRODUCTION

The GE Aviation location in Hooksett, New Hampshire primarily manufactures bladed disks, known as blisks, which are used to compress intake air inside of jet engines. Figure 1 depicts an example of a single stage blisk.

Figure 1: LEAP Series Blisk

Sometimes the disk and blades are manufactured separately (Figure 2), but this is not the case with blisks. Instead of combining two separate components, blisks are machined from a single piece of material, in this case titanium. This avoids unnecessary joints, that act as structural weaknesses, while also increasing compression efficiency. These blisks can vary greatly in shape and size, the plant in Hooksett manufactures blisks ranging from 152.4mm to 914.4mm in diameter. Some blisks have multiple layers of blades, known as stages, and the distance between stages can vary 17.8mm to 50.8mm. The curvature and spacing of blades may also vary between different stages or blisks.

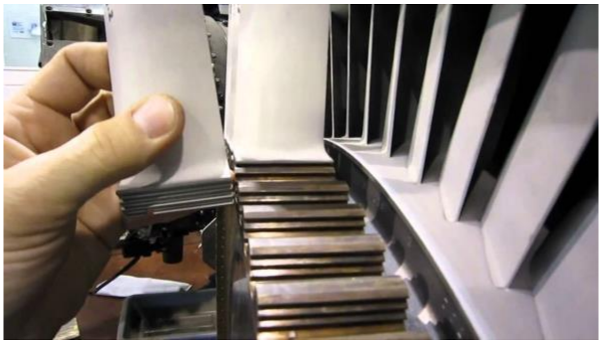


Figure 2: Disk and Blade Assembly

This project focuses on GE Aviation’s LEAP series blisks. The LEAP series refers to a specific set of blisks, all part of GE’s LEAP engine. This series includes three different blisks, two single-stage blisks and one two-stage blisk. The part numbers for the three LEAP blisks are 2468M19P01 (P01), 2468M17P02(P02), and 2468M18G01(G01). The Hookset location is increasing their production of the LEAP series, currently projected to compose 60-80% of their total production volume by 2021.

Blisks are an extremely critical component in jet engines, and imperfections in their fabrication can cause catastrophic failures. GE Aviation goes through a host of inspection processes to confirm that every component of a blisk is up to specification standards. One of the many inspection processes is the inspection of blade root fillets (Figure 3), ensuring that the radii of the fillets are within tolerance. The radii of blade root fillets are critical to the overall strength of each individual blade, making this a vitally important inspection step. The inspection process is currently done by hand, and the Hooksett location wants to automate it.

The goal of this project is to automate the inspection of blade root fillets on LEAP series blisks for GE Aviation. GE Aviation hopes to reduce the time and cost of inspecting each blisk during Quality Assurance (QA) checks. Since inspecting blade root fillets is a repetitive task, a robotic solution was the chosen approach for this project. A robotic system also reduces labor costs associated with root fillet QA checks, requiring less human intervention. Due to Hooksett's increased production of LEAP components, an automated solution would drastically decrease the overall cost and time required for inspecting blisks. Although the scope of this project is limited to LEAP series blisks, the process could be adapted to inspect a wider range of blisks in the future. It should be mentioned that this team worked concurrently with another team to complete this project. Throughout this paper the other team will be referred to as Team-B.

Figure 3: Root Fillets Highlighted in Green

II. BACKGROUND

A. Manual (Current) Inspection Process

There currently exist two methods approved by GE Aviation for inspecting blisk root fillets. The primary method involves an inspector coating the fillet in a thin layer of developer powder. Once coated, the inspector proceeds by dragging a precision ball gauge along the fillet, leaving a trail in the developer powder. This process will be done twice for every fillet, once with a minimum radius ball gauge and again with a maximum radius ball gauge. After dragging the ball gauge along the fillet, the inspector can determine whether or not the fillet passed inspection based on the number of visible contact points in the developer. Any point of contact will be obvious from its lack of developer powder. If there are two contact trails, then the fillet has a smaller radius than the given ball gauge, and if there is only one contact trail, the fillet has a larger radius than the given ball gauge. This method takes a trained QA inspector fifteen to thirty minutes for each stage of a blisk.

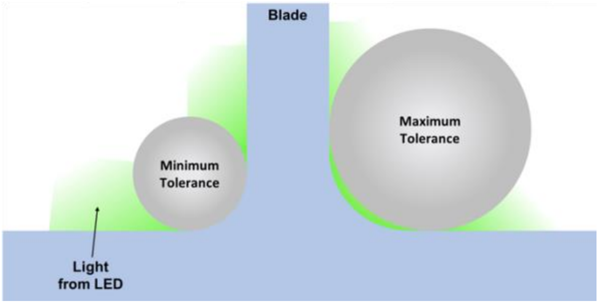


Figure 4: Observed Light Patterns

The second method utilizes the same ball gauges, but instead of developer, the inspector shines a light behind the ball. Then, the inspector can determine whether a fillet passes or fails based on the light patterns produced around the ball. Figure 4 shows an example of the light patterns produced when using this method. Similar to the other inspection method, the points of contact determine whether a fillet passes or fails inspection. If the primary method fails, this method is used to provide definitive results. This method takes longer for operators but is considered by GE to be equally reliable.

GE Aviation currently maintains a three-hour per stage time limit on the full inspection process. For a robotic system to be considered a viable solution, the system must be able to inspect a single stage in less than an hour.

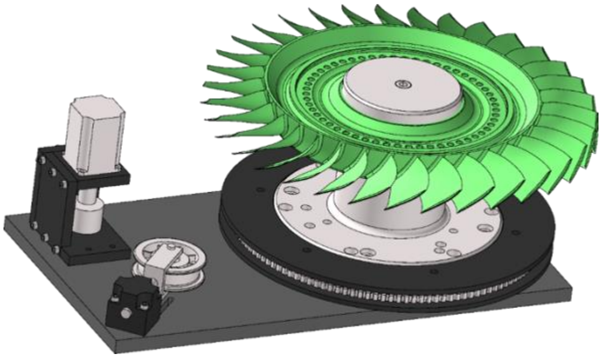


Figure 5: CAD Model of Turntable

B. Previous Project Results

(1) Turntable

The systems turntable is pictured in Figure 5. This turntable utilizes a GE Aviation standard blisk mounting fixture to hold each of the three LEAP series blisks. A 3D printed spacer was made by the previous team to make the blisk fit tightly in the holder, keeping it from moving out of position. The main part of the turntable consists of two acrylic sprockets mounted between two larger aluminum disks. This fixture is then mounted on top of a Lazy Susan bearing, connected to the turntable's base. This fixture is rotated by a tensioned rubber belt, being driven by a Pololu stepper motor. A custom belt tensioner sits in place ensuring that the belt is properly tensioned. The system uses a Pololu A4988 Black motor driver, to drive the motor using a Raspberry Pi. The motor driver is set to have 3,200 steps per revolution. The turntables rotational accuracy was calculated to be ±0.0006in at the hub of the blisk by the previous team.

(2) Controller

The current system is being controlled by a Raspberry Pi. This controls the computer vision processing, python application, the application's user interface, and the communications with the robot and stepper motor driver. The system implements an ABB IRB 1600 industrial robot, with an ABB IRC5 controller.

(3) End-of-Arm Tooling (EOAT)

The top section of the EOAT includes a camera, LED, and ball gauges. Due to several ordering errors, part lead times, and time constraints, the current tool does not use the correct ball gauges. In place of the correct ball gauges the design implements two 3D printed shafts with ball bearings glued to the end. The camera that was used is a GiraffeCam 1.0, a small endoscopic camera, equipped with a 45-degree mirror. This camera was mounted at the tip of the tool, to capture images of the ball gauge. Mounted on the opposite side of the ball gauge is a green surface mount LED. The camera and LED were mounted so that they did not rotate with the ball gauges. This allowed a single LED and camera to be used for the inspection process. A single exposed copper wire was placed at the end of the tool to sense contact with the blisk. This was used to initialize the location of the first blade at the beginning of the inspection process.

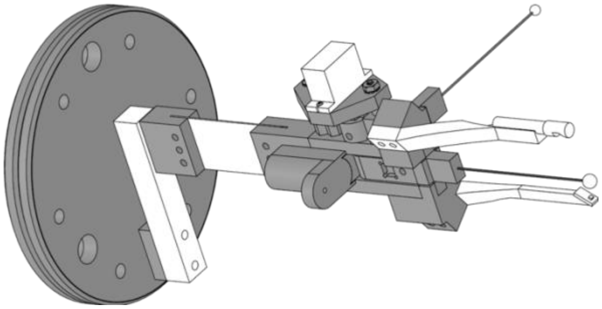


Figure 6: Last Year's EOAT Design

The EOAT's middle section allows the tool to switch between minimum and maximum ball gauges. This is accomplished using a micro-servo that rotates the two ball gauges mounted perpendicular to each other. Originally the design implemented electromagnets to maintain positive retention of the ball gauges, but due to unforeseen issues this was eliminated from the design. To inspect a new blisk, the operator would need to replace the ball gauges by hand.

The bottom section of the EOAT contains mostly compliance related components. The tool is designed to be compliant in two axes and rigid in the other. Compliance in the x-axis was accomplished using a spring steel reed. While compliance in the z-axis was accomplished by two spring loaded pins passing through a machined block. The spring steel reed also reduced the y-axis compliance, making it more rigid. A load cell was mounted at the base of the tool, to give force feedback to the application. The compliances were designed to properly seat the ball gauge in the fillet.

(4) Computer Vision

Last year's team created a computer vision algorithm to determine whether or not a fillet was within tolerance. The algorithm analyzed light patterns in a similar fashion to GE's current secondary inspection test. Rather than relying on the eyes of a human inspector, an endoscopic camera was used to capture images of the light patterns. To process the frames taken from the camera feed, last year's team utilized the OpenCV image processing library.

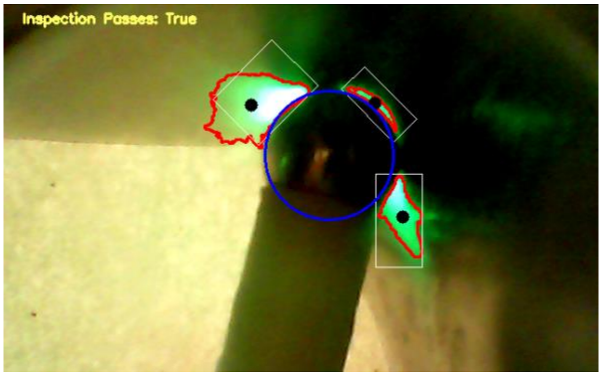


Figure 7: Result from Last Year's C.V. Algorithm

The algorithm found the ball gauge and found light regions by masking the original frame for the green light emitted by the LED. The number of light regions found determined whether or not a fillet was within tolerance.

III. METHODOLOGY

Starting with the cumulative work of previous teams, the first task of this project was to evaluate which elements could remain the same and which needed to be modified or replaced. To fulfil the goal of this project and provide GE Aviation with a robust industrial solution to the problem of automating blisk fillet inspection, several parts of the previous system needed to be modified. The particular elements that needed redesign were the custom end effector, and the underlying control system. The control system was the first element to be redesigned. Followed by a redesign of the blisk inspection application and computer vision algorithm. A camera mount was being designed to comply with a new end of arm tool designed by Team-B. To connect the individual parts of the EOAT an interface plate was designed. The complete end effector assembly was prototyped and tested to improve performance over several iterations. Once the end effector was in its final stages, RobotStudio was used to simulate the robot pathing along the root fillets of a single blade on the P02 single-stage LEAP series blisk. Once the pathing was perfected the robot was programmed to replicate the task. Finally, the programmable logic controller (PLC) was configured to rotate the turntable from one blade to the next after receiving a signal from the IRC5 controller.

A. *System Controller*

The previous system implemented a Raspberry Pi to control the full system, this method was determined to be insufficient for the industrial application at hand. As a result, it was decided that utilizing a PLC was a better solution to the control problem. An Automation Direct Do-more T1HE programmable logic controller, with a variety of auxiliary modules, was chosen as the PLC for this system. The T1HE was chosen because it was readily available to the team and had all the capabilities needed for this application. The ABB IRB 1600 was the best robotic manipulator option, with six degrees of freedom, it is perfectly capable of making the complex movements needed to maneuver a tool in between the blades of a blisk. The ABB IRB 1600 also comes with the RobotStudio software suite, which greatly simplifies complex trajectory planning and many other complicated tasks. The RobotStudio suit also contains a virtual robot controller (mimicking the physical IRC5 controller) allowing the team to test the robot in a simulation mode before running any routines on the physical robot. Another important component of the system is the turntable which is used to rotate the blisk as the robot paths the fillets. This reduced the number of paths that needed to be created for the robot. The final component of the system is a Windows application utilizing computer vision to process frames from the camera feed and determine if fillets are up to specifications.

B. Programmable Logic Controller (PLC)

The PLC will act as a communication hub between each component of the system. Implementing a PLC makes the system more robust, bringing it closer to an industrial standard, and making it easier to implement in a factory setting. The PLC utilizes ladder logic programming, this works by running a list of conditional arguments in sequence, continuously. This allows the PLC to receive digital status updates from its connected components. With this knowledge it can systematically progress through all the steps of the inspection process. The implementation of a PLC also makes the entire system more modular, allowing subsystems to be modified or replaced with relative ease. If the new subsystem can provide the same status updates, the PLC can continue to do its job just the same. Another major benefit to using a communication hub, was that custom drivers didn't need developed to handle cross-component communications.

Figure 8: Automation Director Do-more T1HE PLC

As stated above, the PLC used for this application was an Automation Direct Do-more T1HE; Figure 8 (above) shows a picture of this PLC. The T1HE was connected to a T1H-EBC100 module via Ethernet, allowing the T1HE to remotely control any module connected to the EBC100. This relationship was utilized to place a remote rack of modules closer to the turntable and inspection area, since the T1HE is located well outside of the inspection workspace. The T1HE and EBC100 are connected via CAT5 through a network switch, allowing them to communicate over a private network. When setting up the network originally, the T1HE was having problems when attempting to communicate with the EBC100. The problem was that the T1HE and EBC100 had been configured on different subnets. To resolve this issue the network was set up without a connection to a DHCP server, allowing each device on the network to have a static IP address. This allowed the EBC100 to be reconfigured with a new IP in the proper subnet, resolving the communication problems between the T1HE and EBC100. The IRC5 was also configured on the same subnet. This created a private network allowing the ABB controller and PLC to communicate with one another. Located on the remote rack was a CTRIO module along with a 16-channel digital input module, and 16-channel digital output module all connected to the EBC100. These modules fulfilled all the PLC communication requirements for this system. Although the IRC5 could communicate with the PLC over the network, the information that needed relayed between the two was very simple and could be handled using digital I/O communications instead.

C. Physical Components

The previous system implemented a Raspberry Pi to control the full system, this method was determined to be insufficient for the industrial application at hand. As a result, it was decided that utilizing a PLC was a more robust solution to the control problem. An Automation Direct Do-more

T1HE programmable logic controller, with a variety of auxiliary modules, was chosen as the PLC for this system. This PLC was chosen because it was readily available to the team and had all of the capabilities needed for this application. The ABB IRB1600 was the best robotic manipulator option, with six degrees of freedom, it is perfectly capable of making the complex movements needed to maneuver a tool in between the blades of a blisk. The ABB IRB1600 also comes with the RobotStudio software suite, which greatly simplifies complex trajectory planning and many other complicated tasks. The RobotStudio suit also contains a virtual robot controller (mimicking the physical IRC5 controller) allowing the team to test the robot in a simulation mode before running any routines on the physical robot. Another aspect of the system is the turntable which was used to rotate the blisk as the robot pathed the fillets. This reduced the number of paths that needed to be created for the robot. The final component of the system is a windows application utilizing computer vision to process frames from the camera feed and determine whether or not fillets are up to specifications.

D. *Turntable*

The pre-existing turntable proved through testing that it could successfully rotate a blisk from one blade to the next. Even though the turntable was operational it was necessary to measure the error offset in root fillet locations as the blisk was rotated. This was done by mounting a digital dial indicator on the base of the turntable and measuring the Z- and X-axes locations at root fillets while rotating the blisk. After making several rotations the turntable's error was calculated to be ±3mm in the X, Y, and Z axes. The EOAT was designed to account for this, which will be explained further in the next section. This error is within the necessary specifications for the application, and therefore there was no need to redesign the turntable.

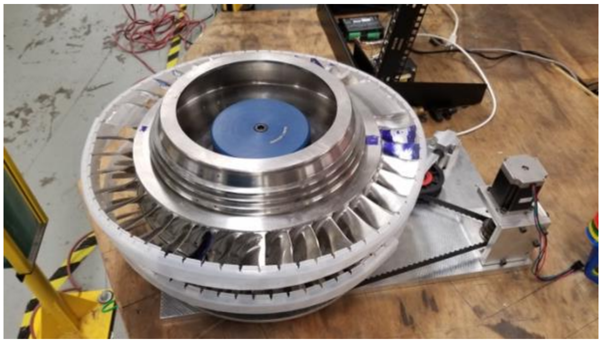


Figure 9: Example of Blisk Mounted on Turntable

The next step was to integrate the pre-existing turntable into the new system. To use the turntable in the new system a new stepper motor driver was chosen to replace the existing Pololu driver. The specific driver was chosen due to its compatibility with the PLC and the existing stepper motor. The selection and integration of this new driver was handled by Team-B.

E. *End-of-Arm Tooling (EOAT)*

After evaluating the pre-existing EOAT it was determined to be inadequate for this application. The most obvious problem that this tool suffered from was excessive play in its mechanical design. The main contributor to this problem was the compliance designed into the tooling to account for the previous team's tool-to-blisk approach angle and any turntable error offsets they may have encountered. More play was introduced into the system by the 3D printed part tolerances, and the servo mechanism that was used to change between ball gauges. Furthermore, the spring steel reed at the base of the tool didn't have the strength to hold the weight of the ball gauge mechanism steady, allowing the tip of the tool to move independently of the base as the robot moved. When trying to maneuver the tool in a very tight space, such as between the blades of a blisk, unintended movement of the ball gauge could cause the system to fail. Another major problem with the previous design was that the ball gauge would skip along the fillet while pathing, causing the ball gauge to be unseated at random times during the inspection process. The pre-existing EOAT was designed to path perpendicular to the blisk as it ran along the fillet, this design decision lead to the tool skipping problem.

It was determined that a complete redesign of the EOAT was necessary to resolve these issues. A successfully redesigned EOAT would have to adequately illuminate the ball, switch between different sized ball gauges in less than one minute, be able to path along a fillet without skipping, be perpendicularly compliant within 3mm of the root fillet, maintain positive retention of the ball gauge and an EOAT tip error of less than 0.1mm. Using these specifications as a guideline a new tool was designed by Team-B.[1]

Figure 10: Team B's Final EOAT Design

The newly designed EOAT lacked the necessary considerations for integrating a camera mount and needed to be redesigned to accommodate for this issue. Since the EOAT implemented an ATI tool changer as a solution for switching between different ball gauge sizes during inspection, it was necessary to connect the camera mount to the EOAT on the male side (or arm side) of the tool changer. This configuration uses a single camera for the inspection process of multiple ball gauge sizes. The current layout forces the camera mount to reach, unsupported, the entire length of the tool changer plus the length of the ball gauge shaft and ball gauge holder assembly.

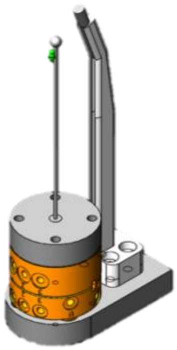


Figure 11: Model of Final EOAT

There was a considerable amount of extra material included in this design of the EOAT that could be removed to decrease the necessary length of a camera mount assembly. The excess material was removed, allowing for a shorter camera mount assembly, and increased camera stability. Another issue with the EOAT design is a crude press fit for the ball gauge shaft into the ball gauge holder. This issue was resolved by adding a set screw to properly secure the ball gauge shaft to the holder.

F. Compliance

To resolve the tool compliance and fillet skipping problems, a new tool-to-blisk approach angle was decided upon. Rather than use the previous teams perpendicular approach, Team-B determined that 20 degrees was an optimal approach angle for the new EOAT. This allowed the x and y axis compliances to come from the elastic deflection range of the ball gauge shaft. As stated in the Turntable section, the necessary compliance for the x and y axes needed to be roughly 3mm to account for the turntable tolerances. The previous team determined that a force greater than 1/8lbs or 0.556N could damage the surface of the blisk. To provide the compliance necessary without exceeding a force of 0.556N the ball gauge shaft diameter would need to be calculated (as seen in the calculations below).

$$\sigma_B = \frac{F_a * L^3}{3 * E * I}$$

$$\sigma_B = Maximum\ Deflection = 3mm$$
$$= 0.003m$$

$$F_a = Force\ at\ EOAT\ Tip = 0.125lbs$$
$$= 0.556N$$

$$L = Length\ of\ Beam = 152.4mm$$
$$= 0.152m$$

$$I = Area\ Moment\ of\ Inertia\ = \frac{\pi * r^4}{4}$$
$$= \frac{\pi * d^4}{64}$$
$$= 3.98e - 12\ m^4$$

$$E = Young's\ Modulus\ (Steel)$$
$$= 3e10^7\ lbs/in^2$$
$$= 2.07e11\ N/m^2$$

$$d = \sqrt[4]{\frac{64 * F_a * L^3}{3 * E * \sigma_B * \pi}} = 0.002m = 2mm$$

G. *Camera Mount*

Originally the team considered having a camera apparatus in a fixed location either above or below the blisk, rather than on the EOAT itself. This seemed like a favorable approach for a number of reasons; the picture quality would be much higher, the tool would be less bulky, and pathing would be simpler. But due to the sharp curvature of the blisks blades, it was not possible to see every part of a fillet from a single fixed location, regardless of the viewing angle. The most realistic implementation of this solution would need multiple cameras (2+) viewing a tool as it pathed the fillet. Since this solution required more cameras, it was not implemented. Instead, a tool side camera mount was the chosen approach.

Figure 12: GiraffeCam 1.0 Endoscopic Camera

The first step to designing the camera mount was to determine how far the camera needed to be from the ball. The camera being used, a GiraffeCam 1.0 (Figure 12), was not designed for close proximity applications, so the previous team had done some testing to find the camera's workable focal ranges. Combining their results with more testing, a distance of 19.05mm was selected. This distance provided the clearest images, as close to the ball as possible. With the distance decided upon, it was time to choose the angle at which the camera would view the ball. Since the EOAT's approach angle was 20 degrees and the LED would be placed within the fillet as the tool pathed, the camera would be mounted at a 20-degree angle from the tip of the ball. Now, the camera mount must be able to reach from the bottom of the tool changer, so that a single camera can be used throughout inspection. The first prototype of the camera mount was designed to sit on a plate placed in-between the tool changer and the ABB 1600, reaching the whole length of Team-B's original tool.

Figure 13: First Prototype of Camera Mount

There were several issues with this prototype; the wire sheath would have worked better as a press fit, the camera couldn't be removed once the mount was attached to the plate, and the width of the curved structure was just a little large to fit between the P02's blades. Fixes for these problems all came along with the removal of the EOAT's excess material. The camera mount went through a few more iterations finally ending as seen in Figure 14.

Figure 14: Model of Final Camera Mount

While this camera mount was being designed, an interface plate was being designed simultaneously. The pin and mounting holes located at the bottom of the piece are meant to interface with the plate that will be discussed in greater detail in the next section. Several prototypes of this camera mount were 3D printed on varying machines. The final part was printed on a Dimension SST 1200es machine with a 0.01-inch layer thickness and a 0.006 in error range.

H. *Computer Vision*

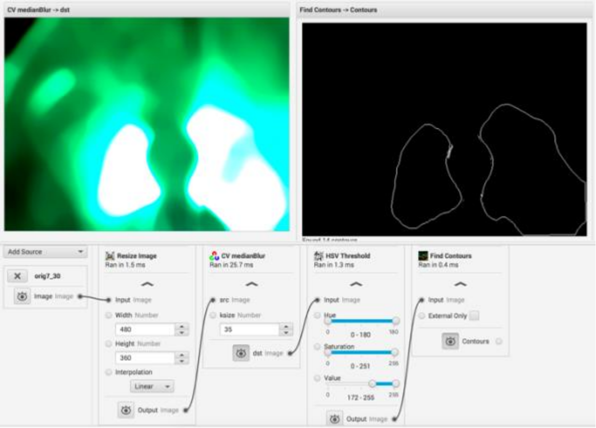
Computer vision is an integral component of the blisk inspection system as it is responsible for determining whether each fillet is actually within tolerance. As previously described in the Camera Mount section, the LED is placed on the opposite side of the ball gauge from the camera mount. By placing the LED on the opposite side, the ball gauge obstructs the unwanted light from the LED providing the camera with a clear view of the light patterns created by the point(s) of contact between the ball gauge and the fillet. The video captured by the camera must be examined to determine whether or not the visible light pattern indicates a fillet that is within tolerance.

Last year's team researched various image processing libraries and decided that the open-source OpenCV platform had all the tools required for this project's image processing needs. OpenCV is written in C++ and therefore OpenCV's primary interface is C++. However, bindings and wrappers exist in other languages allowing them to utilize the OpenCV image processing library. The library is also cross-platform, so techniques from last year's computer vision algorithm, programmed in Python, can be used despite the decision to switch from a Raspberry Pi to a Windows laptop.

The decision made by last year's team to program using Python 2.7 was a surprising choice, considering Python 3.0 was released in 2009 with substantial additions and significant changes to the syntax of the language. Industrial applications are already known for using outdated programming languages because many of these applications remain unchanged over long periods of time. Considering the possibility that the blisk inspection system could have a long shelf-life in industry, it was decided that the most recent stable version of Python should be used (Python 3.6.4). The decision to make this switch to Python 3 meant that the techniques, rather than the exact code, used by last year's team could be utilized in the new implementation in Python 3 due to the major differences in syntax.

I. *Computer Vision Algorithm*

The key techniques used by last year's team for analyzing and deciding whether a fillet is within tolerance were implemented again in the new algorithm as both image masking and the Hough Circles Transformation were determined to be critical in the detection of contact points. Additional OpenCV features were utilized to improve the efficiency and the accuracy of the inspection.



Resize Image: Ran in 1.5 ms
Width: 480
Height: 360
Interpolation: Linear

CV medianBlur: Ran in 25.7 ms
ksize: 35

HSV Threshold: Ran in 1.3 ms
Hue: 0 - 180
Saturation: 0 - 251
Value: 172 - 255

Find Contours: Ran in 0.4 ms
External Only: ☐

Figure 15: GRIP Engine Isolating Light Regions

Similar to last year’s team, the GRIP (Graphically Represented Image Processing) engine helped in the development of the computer vision algorithm. However, unlike last year’s team, the generation of Python code from GRIP was unusable because the GRIP engine only generates Python 2 code. Last year's team used GRIP to mask each image and detect contours. To mask an image for the green LED light, the image was first converted from the default RGB color model to an HSV color model using the OpenCV tool, HSVThreshold. The GRIP engine has sliders that allow the user to define custom ranges for hue, saturation, and value. A mask of the original image is the result because only the colors that fall in the defined range will be visible. The sliders for defining the HSV threshold were very useful in the new algorithm because they can be configured quickly to find accurate ranges for the hue, saturation, and value that will be implemented in Python 3.

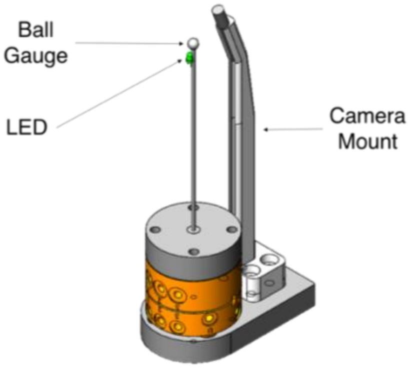


Figure 16:Final EOAT w/ Labeled Vision Components

Last year’s team utilized the Hough Circles Transformation from the OpenCV library to find the ball gauge in the camera frame. The Hough Circles function in the OpenCV library allows for a range, a minimum radius and a maximum radius, to be entered and only the circles with a radius within that range will be identified. Each LEAP series blisk has its own unique collection of ball gauges, each of which has its own unique radius. By determining the radius of each ball gauge in the camera frame, configurations can be set for the different ball gauges for each blisk. The algorithm created by last year’s team used a small range for the radius. A narrow range was possible due to the design of the end of arm tooling. Both the position of the camera and the position of the ball gauge were fixed, allowing the Hough Circles feature to easily identify the ball gauge when given the appropriate range of the radii.

The newly designed blisk inspection system drags the ball gauge along each fillet, which results in a varying amount of bend in the shaft. The variable bending of the shaft causes the relative distance between the fixed camera and the ball gauge to change. As a result, a wider range needed to be defined inside of the Hough Circles function. In relation to the fixed view of the camera, the bending of the shaft also causes the location of the ball gauge to change. This means that the ball gauge could be located anywhere inside of the camera frame and possibly even outside of the frame at the extremes where plastic deformation of the shaft takes place. The new algorithm needed to detect the ball gauge no matter where the ball gauge is located inside of the frame. The inspector is also notified if the fillet is failing inspection due to the ball bearing being outside of the frame or due to the ball bearing being too close to the edge of the frame, preventing the algorithm from examining the visible light patterns around the ball bearing.

The new algorithm uses the same idea that last year’s team used to distinguish the different regions of light. OpenCV’s findContours function outlines each region of light found from the HSV threshold mask of the original frame. By counting the different regions of light in the frame, the inspection status of the fillet at the location of where the frame was captured can be updated. When inspecting using the small ball gauge, there must be exactly two distinct regions without even taking into account the location of each region. If a different number of regions is found, the fillet fails inspection at the location because the fillet must be smaller than the desired minimum size. The fillet meets the minimum size requirement if the center of each distinct contour region is located outside of the circle detected for the ball gauge.

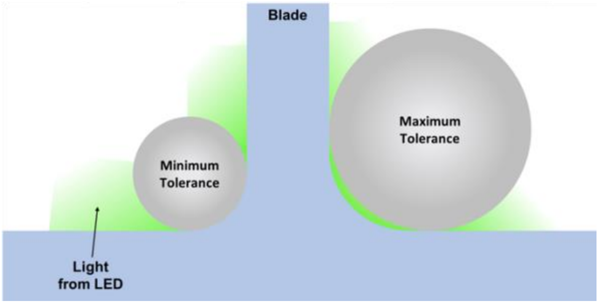


Figure 17: Observed Light Patterns

When inspecting using the large ball gauge, there must be exactly three distinct regions of light. Any other number of regions means the fillet is not within the desired tolerance. If the center of each of the three contour regions is outside of the circle detected for the ball gauge, the fillet meets the maximum size required at that location.

J. Inspection Application

The team determined that an application running on a Raspberry Pi was insufficient for the task at hand. As a result, the team decided that the development of an application running on a Windows machine would be able to effectively run the inspection and be a more professional approach. There are many different ways to create a Windows application, but the team decided that a Windows Forms application could be created easily using Visual Studio. By deciding to implement the blisk inspection application as Windows Forms application, the team was able to take advantage of the easy drag and drop feature for designing the user interface. While the user interface is an important aspect of any application, the team decided that more time needed to be spent on implementing the app's core functionality.

The Windows Forms application is programmed in Visual C# and the team decided that the application should be the primary component interacting with the user. Originally, the team wanted the entire inspection process to be automated; however, the team soon realized that the inspector would have to be involved for mounting the blisk on the turntable. It was also later determined that the system would require the inspector to flip the blisk when inspecting the two stage blisk. The inspection system can be fully autonomous in the near future, but this goal was out of the scope of the project.

There are many steps during the inspection of blisk root fillets. The team decided that a simple home screen (*Figure 19*) for the application would provide the user with an easy start screen and menu. There are two important buttons on the home page. The first button, "Inspect Blisk", will start the inspection setup process if clicked by the user. The second button, "View Inspection History", will open up Windows File Explorer to the directory containing the results of past inspections. At the end of every inspection, all frame results are saved into a spreadsheet that is then placed in the results file directory. Clicking the "Inspect Blisk" button will display a new screen (*Figure 20*) allowing the user to select which LEAP series blisk they are inspecting. To prevent confusion in blisk selection, a model and specifications for the selected blisk type are shown. A series of displays will then guide the user through safely mounting the blisk onto the turntable (*Figure 21*) and through indexing blade zero of the blisk. When the inspector completes the setup process, they will be taken to the inspection user interface (*Figure 22*). From here, the inspector can start an inspection by clicking on the "►" button or the inspector can configure the inspection settings by clicking on the settings button (gear icon).

The base of the application is a Form. A Form is a tool unique to a Windows Forms application and it acts as not only the main controller for the entire application, but also as a container for all the UI components. A Control object was made for each screen the team decided should be displayed to the user at different times during the inspection process. Actions made by the user on any of the Control object screens are handled on the UI thread by the main Form.

After settling for a semi-autonomous system, the team implemented a way to safely guide the inspector through these processes when they arise. After selecting the specific blisk, directions appear guiding the user through the mounting process. The team decided that the directions should be easily understood by someone with technical skills but with no knowledge of the specific system. While the inspection is running, the inspector is able to follow along as each fillet is inspected and marked as within tolerance or out of tolerance. If the inspector is inspecting the two stage blisk, another screen will appear prompting the user with directions on how to flip the blisk and securely remount the blisk.

The computer vision algorithm is started by the application after inspection setup is completed. Details about the selected blisk will be sent to the Python script containing the computer vision algorithm. Threading was a necessary component in the implementation of the application because the team determined that is necessary to keep the application running efficiently while simultaneously processing each frame captured by the camera.

The Windows application is responsible for passing frames from the camera feed to the team's computer vision algorithm. Unlike last year's computer vision algorithm, the team's algorithm did not limit the frames per second (FPS) that could be used. The new algorithm in combination with the change from a Raspberry Pi to a Windows machine allowed the inspection to use a higher frame rate. The camera's frame rate of 30 FPS was able to be used. The minimum spatial resolution specified by GE Aviation is 25 captures per inch (CPI), so a frame rate of 30 FPS would easily surpass the minimum requirement. If any frames still needed to be processed after the pathing of the fillet finished, they could still be tested by the team's algorithm while the system was switching to the next fillet.

K. ABB Robot Pathing

Once the complete EOAT was designed and the ideal tool approach angle was determined it was time to program the ABB to path along fillets. First, all of the system's physical components were modeled in RobotStudio. Then, tool and work station coordinate systems were defined. Finally, utilizing RobotStudio's auto path feature, a new target pose was placed every millimeter along the fillets curve. The tool orientation at the targets needed modified to avoid collisions, then the code could be uploaded to the physical robot.

To initiate pathing, it is necessary for the inspector to properly align the tool with the first blade. Once aligned, an alligator clip is attached to the blade, allowing the robot to sense initial contact. After initialization, the robot autonomously paths the fillets, sending signals to the PLC to change blades at the end of its path. After making one full pass of the blisk, the robot paths over to the tool holder, grabbing the next size ball gauge.

IV. RESULTS

A. End of Arm Tooling

Due to lead times, the proper ball gauges were not received in time to test the entire tool. Smaller ball gauges with the same length shaft were used as a stand-in, during testing. The EOAT successfully fit between the blades of the LEAP series blisks, allowing for a 20-degree approach angle. The new tool design left no measurable play within the system and could switch between different sized ball gauges. The camera mount design successfully kept the camera steady while the robot moved, providing good images for the computer vision algorithm to process. The interface plate successfully connected all the parts of the EOAT, properly aligning the camera mount and the ball gauge. The interface plate was also able to successfully align the EOAT on the ABB robot.

B. *Computer Vision*

The team's computer vision algorithm was successfully able to determine whether or not the fillet was within tolerance for a given frame. Since the team's EOAT does not utilize any force sensors, precise pathing had to be accomplished to make sure the ball gauge fell into the fillet. Unfortunately, the team's algorithm was unable to distinguish between the ball gauge being seated in a fillet and the ball gauge having one contact point with a blade. However, if contact was lost between the ball gauge and the blisk, an error message is displayed to the inspector.

C. *Windows Application*

The Windows Forms application provided the inspector with a simple and information interface. When the inspector was required to mount/unmount a blisk or to index the starting blade for the blisk, step-by-step directions were displayed on the screen to guide them through the process. As the inspection was running, the inspector was able to view inspection results for each fillet in real-time. Frames from past inspections were also easily accessible to the inspector at any time through the home page.

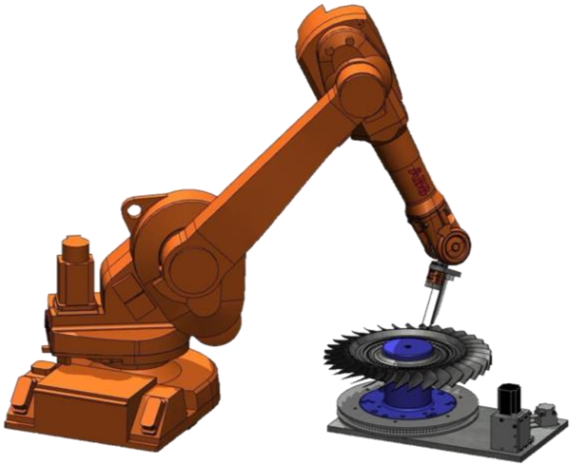


Figure 18: Entire Blisk Inspection System

V. CONCLUSION

In summary, each individual component that was designed as part of this solution operates as intended in its current capacity. The team is also confident that with further work this system can reach a more reliable state. In addition, the team believes that the Blisk Inspection System is a feasible solution for inspecting General Electric Aviation's LEAP series blisks.

A. *Social Implications*

Autonomous systems can be introduced into an industrial workplace to replace monotonous tasks. Oftentimes, automated systems create a great deal of concern. Autonomous systems are forcing people into unemployment because there is a lesser need for human labor. Despite this common concern, the Blisk Inspection System should be looked at differently. This semi-autonomous system is performing a very repetitive inspection process that relies on a high degree of accuracy. With a robotic system, we will not see the decrease in accuracy that is often seen in humans when it comes to performing very tedious and time-consuming tasks. Due to the possible catastrophic consequences of even the tiniest mistake in the inspection of a blisk, the need for a consistent and accurate inspection method is clear.

The common industrial workspace comes with its fair share of safety concerns. The implementation of autonomous systems does not eliminate these hazards. The Blisk Inspection System is a semi-autonomous system and a human is required at several points during the inspection of a blisk. A shared workspace between robotic systems and humans only increases the number of safety concerns found in the common industrial workplace. The Blisk Inspection System accounts for these concerns when guiding the human inspector through the human-required processes. For example, the inspector is directed to put on their work gloves before handling a blisk during the blisk mounting process. A fully autonomous inspection system would significantly reduce the safety concerns found in both the semi-autonomous system and the manual inspection process.

B. *Future Work*

There are many ways in which this project can be improved upon and expanded by future work. This section focuses on these areas.

(1) Turntable

The current turntable design could be improved by implementing a positional feedback system. Currently, the stepper motor is driven without feedback, and although this showed no issues in testing, the addition of positional feedback would make the system more robust. Another way in which the turntable could be improved is by changing the orientation in which the blisks are held. The current turntable holds blisks horizontally, leaving very little space between the bottom of a blisk and the work surface. Holding blisks in a vertical orientation leaves more space for pathing on either side of a blisk. This problem was discussed further in Team-B’s paper. The system as it stands, requires that a two stage blisk be flipped over to inspect the second stage, and a vertical turntable implementation would eliminate the need for this.

(2) EOAT

The primary problem with this EOAT design is its lack of force feedback. Without force feedback the system relies solely on accuracy to path a fillet, which is uncharacteristic of a robotic solution. Being that this was outside the scope of this project, there was not enough time to design force feedback into the EOAT. This change would be necessary before implementing this EOAT in an industrial setting.

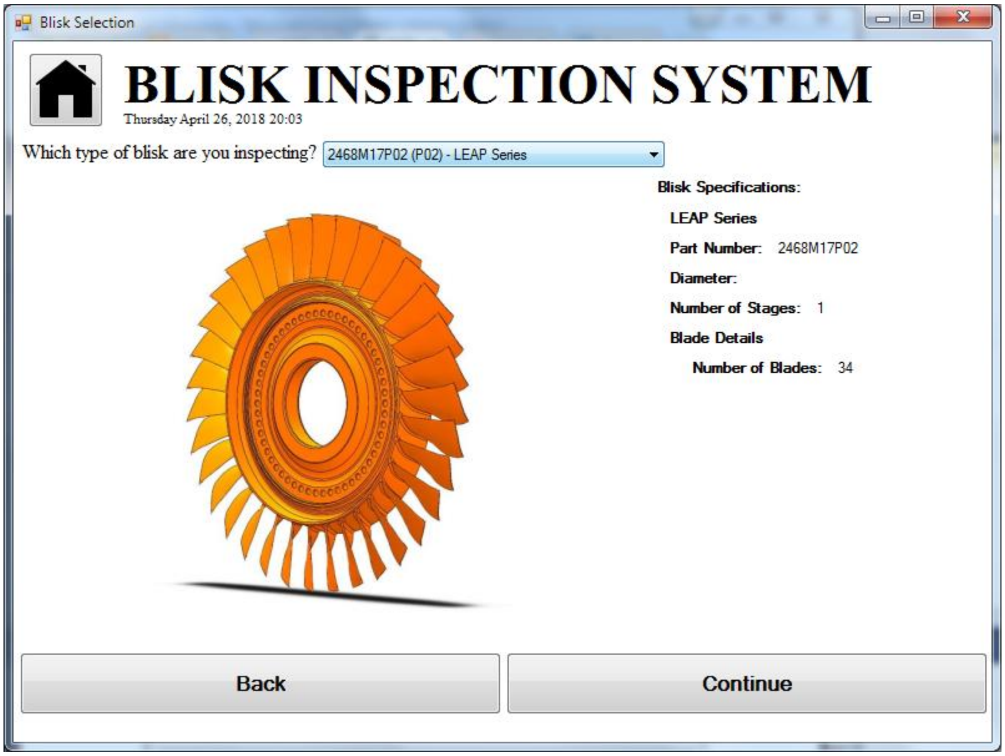
VI. ACKNOWLEDGEMENTS

The team would like to thank both Professor Craig Putnam and Professor Kenneth Stafford for their constant support over the last nine months. The team would also like to extend a big thank you to our primary contact at GE Aviation, John Graham, along with the rest of the GE Aviation plant in Hooksett, NH for their support throughout the course of the project.

APPENDIX A

WINDOWS FORMS APPLICATION

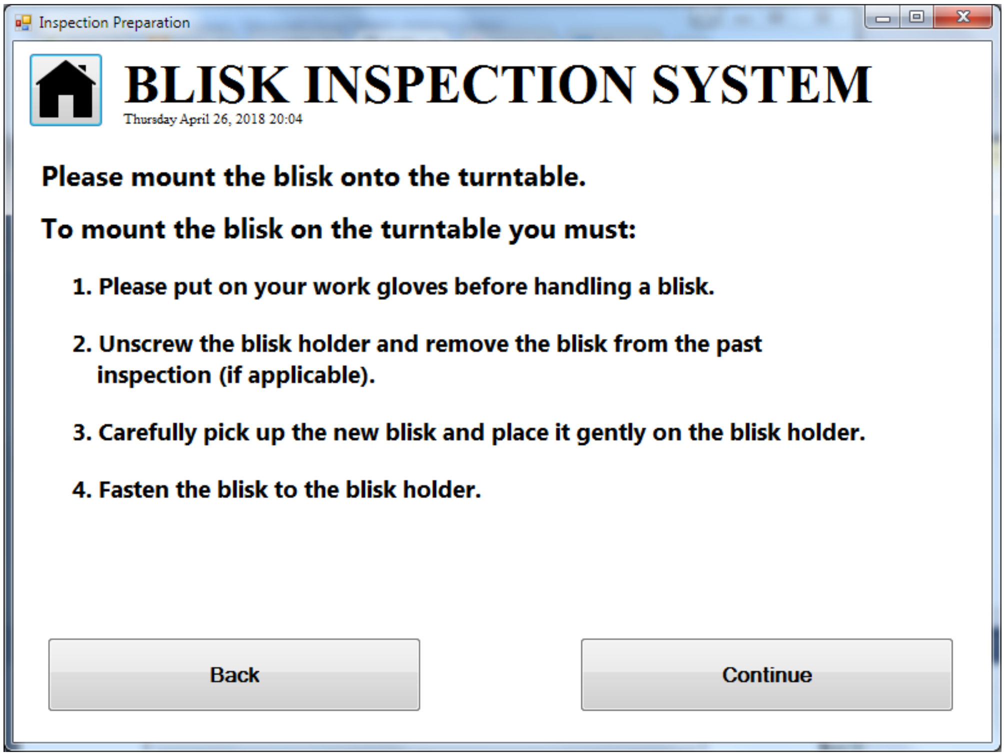
Figure 19: Homepage of Inspection Application



Which type of blisk are you inspecting? 2468M17P02 (P02) - LEAP Series

Blisk Specifications:
LEAP Series
Part Number: 2468M17P02
Diameter:
Number of Stages: 1
Blade Details
Number of Blades: 34

Figure 20: Blisk Selection Screen of Inspection Application



BLISK INSPECTION SYSTEM

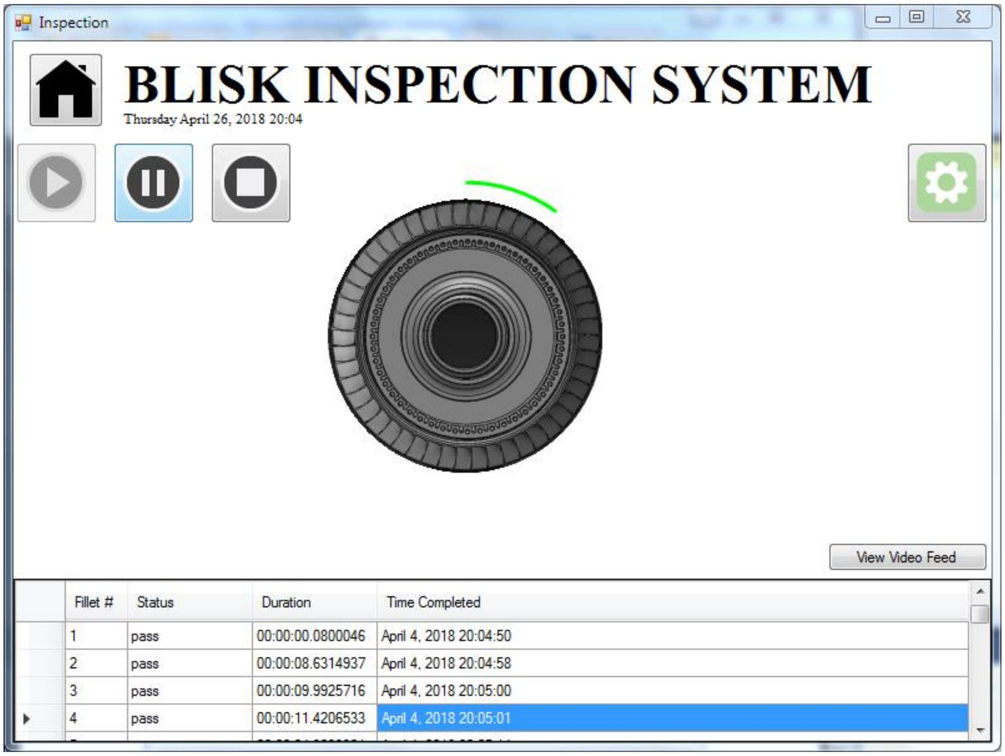
Thursday April 26, 2018 20:04

Please mount the blisk onto the turntable.

To mount the blisk on the turntable you must:

1. **Please put on your work gloves before handling a blisk.**
2. **Unscrew the blisk holder and remove the blisk from the past inspection (if applicable).**
3. **Carefully pick up the new blisk and place it gently on the blisk holder.**
4. **Fasten the blisk to the blisk holder.**

Figure 21: Screen Guiding Inspector Through Blisk Mounting Process



BLISK INSPECTION SYSTEM

Thursday April 26, 2018 20:04

Fillet #	Status	Duration	Time Completed
1	pass	00:00:00.0800046	April 4, 2018 20:04:50
2	pass	00:00:08.6314937	April 4, 2018 20:04:58
3	pass	00:00:09.9925716	April 4, 2018 20:05:00
4	pass	00:00:11.4206533	April 4, 2018 20:05:01

Figure 22: Running Inspection w/ Inspection Feedback

REFERENCES

[1] Hopkins, M., 2018, “Blisk Inspection System,” Technical Report, Worcester Polytechnic Institute, Worcester, MA.

[2] Happell, B., Orton, K., Ouellette, K., Sinkler, C., 2017, “Blisk Inspection System,” Technical Report, Worcester Polytechnic Institute, Worcester, MA.