NIST Technical Note
NIST TN 2270

Uncertainty Analysis of the Dual-Lift Use Case for the Industrial Wireless Testbed

Karl Montgomery
Richard Candell
Mohamed Kashef (Hany)

This publication is available free of charge from:
https://doi.org/10.6028/ NIST.TN.2270

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Technical Note
NIST TN 2270

Uncertainty Analysis of the Dual-Lift Use Case for the Industrial Wireless Testbed

Karl Montgomery
Richard Candell
Mohamed Kashef (Hany)
Smart Connected Systems Division
Communications Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/ NIST.TN.2270

October 2023

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

NIST TN 2270
October 2023

Publication History
Approved by the NIST Editorial Review Board on 2023-09-30

How to Cite this NIST Technical Series Publication
Montgomery K, Candell R, Kashef M (2023) Uncertainty Analysis of the Dual-Lift Use Case for the Industrial Wireless Testbed. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Technical Note (TN) NIST TN 2270. https://doi.org/10.6028/NIST.TN.2270

NIST Author ORCID iDs
Karl Montgomery: 0000-0001-5405-6665
Richard Candell: 0000−0002−6679−8823
Mohamed Kashef: 0000−0002−6619−3509

Contact Information
karl.montgomery@nist.gov

Abstract

This report serves to characterize the uncertainty of measurements for key performance indicators (KPIs) used to measure the physical and network performance of the NIST industrial wireless dual-lift testbed. A review of the methods used to determine uncertainty is presented. We use the method known as the Guide to the Expression of Uncertainty in Measurement (GUM) to calculate the uncertainty values for each measurement. A description of the dual-lift use case and the KPIs are presented. The resulting expanded uncertainty values are calculated for the KPIs, which are the positional Cartesian error between the leader and follower robots, and the latency of the data packets associated with the informational transactions of the leader and follower's control loop. It is concluded that the measurement systems used in the dual-lift use case have relatively small uncertainty values for the detection of latency perturbations in the real-time control application implemented on the testbed.

Keywords

Industrial wireless communications, Measurement uncertainty, Manufacturing

Table of Contents

1. Introduction to Calculating Uncertainty 1
1.1. Definitions and Key Terms 1
1.2. GUM Method for Calculating Uncertainty 1
1.2.1. Evaluating Standard Uncertainty 2
1.2.2. Determining the Combined Standard Uncertainty 2
1.2.3. Determining Expanded Uncertainty 2
2. Description of Dual-Lift Use Case 3
2.1. Use Case Overview 3
2.1.1. Leader-follower implementation 5
2.1.2. ROS nodes, robot description 6
2.1.3. Message flows, leader-follower stream 6
2.2. Description of the Position Tracking Measurement System 6
2.2.1. Physical 3D Tracking Camera System Setup 7
2.2.2. Cartesian error magnitude between leader and follower KPI 7
2.3. Description of the latency measurement system 7
2.3.1. Latency measurement devices 7
2.3.2. RTT KPI, from the leader-follower TCP stream. 9
3. Uncertainty calculation of Cartesian error KPI 9
3.1.1. Description of data collection for determining uncertainty 9
3.1.2. Resulting calculation of combined, and expanded uncertainty 11
4. Uncertainty calculation of RTT KPI between leader and follower 11
4.1. Uncertainty of RTT from the ET2000, a Type B uncertainty 12
4.2. Uncertainty of RTT from the SharkTap, a Type A uncertainty 12
4.3. Results 13
5. Conclusion 13
References 15

List of Tables

Table 1. Definitions of Uncertainty Terms. 1
Table 2. Leader's Maximum Standard Deviations for Points A to D in the Circle 10
Table 3. Follower's Maximum Standard Deviations for Points A to D in the Circle 10
Table 4. Type A input values to calculate RTT uncertainty using the SharkTap 13

List of Figures

Figure 1. Dual-Lift Use Case Components 4
Figure 2. Network Diagram for Dual-Lift Use Case..... 5
Figure 3: ROS Message Flows for Dual-Lift Use Case..... 6
Figure 4: Industrial Wireless Testbed with Overhead 3D Tracking Camera System 7
Figure 5: ET2000 Network Packet Capturing Device 8
Figure 6: SharkTap Gigabit Ethernet TAP Device. 8
Figure 7: Static Points on Circular Path for Measuring Uncertainty 10

1. Introduction to Calculating Uncertainty

This section introduces the method used to determine the uncertainty of the measurements made in the dual-lift use case deployed within the NIST industrial wireless testbed. A description of the testbed components and measurement systems will be covered in Section 2.

1.1. Definitions and Key Terms

Understanding the key terms regarding uncertainty and its calculation is needed to comprehend the later sections in this report. These terms include standard uncertainty, Type A and B evaluations of uncertainty, combined standard uncertainty, expanded uncertainty, and coverage factor. These definitions directly come from the Guide to the Expression of Uncertainty in Measurement (GUM) [1] and are presented in Table 1 for convenience.

Table 1. Definitions of Uncertainty Terms.

Uncertainty Terms	Definitions
Standard uncertainty	Uncertainty of the result of a measurement expressed as a standard deviation.
Type A evaluation of uncertainty	Method of evaluation of uncertainty by the statistical analysis of series of observations.
Type B evaluation of uncertainty	Method of evaluation of uncertainty by means other than the statistical analysis of series of observations.
combined standard uncertainty	Standard uncertainty of the result of a measurement when that result is obtained from the values of several other quantities, equal to the positive square root of a sum of terms, the terms being the variances or covariances of these other quantities weighted according to how the measurement result varies with changes in these quantities.
expanded uncertainty	The quantity defining an interval about the result of a measurement that may be expected to encompass a large fraction of the distribution of values that could reasonably be attributed to the measurand.
Coverage factor	numerical factor used as a multiplier of the combined standard uncertainty to obtain an expanded uncertainty.

1.2. GUM Method for Calculating Uncertainty

The GUM method consists of evaluating uncertainty based on a variety of factors, including errors, random effects, and corrections. Errors stem from the imperfect system of measurement and can be random. Systematic errors occur when a recognized effect produces an influence on the measurement. These errors can be corrected by using a correction factor. The GUM also

shows how to evaluate standard uncertainty, determining the combined standard uncertainty, and finally determining the resulting expanded uncertainty.

1.2.1. Evaluating Standard Uncertainty

Evaluating the standard uncertainty for a measurement starts by determining the measurement equation, seen in Eq. 1, where X_i are the random input quantities, f is the measurement function, and Y is the measurement output value.

$$Y = f(X_1, X_2, X_3 \ldots\ldots) \tag{1}$$

For type A uncertainties, value(s) of standard uncertainty are to be calculated, which are determined by taking the standard deviation of an input estimate, x_i, of the random input quantity X_i to obtain the standard uncertainty, $u(x_i)$.

For type B uncertainties, the standard uncertainty, $u(x_i)$, is evaluated by scientific judgement based on all the available information on the possible variability of X_i. As stated in the GUM, the available information may include previous measurement data, experience with or general knowledge of the behavior and properties of relevant materials and instruments, manufacturer's specifications, data provided in calibration and other certificates, and lastly, uncertainties assigned to reference data [1].

1.2.2. Determining the Combined Standard Uncertainty

For uncorrelated input quantities, Eq. 2 is used to determine the combined standard uncertainty, $u_c(y)$, from the positive square root of the combined variance, $u_c^2(y)$:

$$u_c^2(y) = \sum_{i=1}^{N} \left(\frac{\partial f}{\partial x_i}\right)^2 u^2(x_i) \tag{2}$$

In Eq. 2, N is the number of input variables, and $\partial f / \partial x_i \ is$ the partial derivative of f taken with respect to each x_i. For correlated input quantities, please refer to the GUM, since the measured input quantities in this report were determined to be uncorrelated.

1.2.3. Determining Expanded Uncertainty

The expanded uncertainty is the product of the coverage factor, k, and the combined standard uncertainty, $u_c(y)$. The value of k is determined by the inverse of the two-tailed Student's T Distribution function. To calculate k, the inputs to this function used are the effective degrees of freedom, v_{eff}, and the confidence level ratio, from 0 to 1. If one was to choose a confidence level of 99%, meaning in 99% of the measurements, the error in the measurement does not exceed the stated uncertainty value. A confidence level ratio of 0.99 equates to 99%. Please refer to the GUM for the equation to calculate v_{eff}, which uses the degrees of freedom from each input variable, corresponding uncertainties as weighting factors, and the combined uncertainty in

its calculation. The effective degrees of freedom value are used to calculate k. Below is the equation used to calculate the resulting expanded uncertainty:

$$U = ku_c(y). \tag{3}$$

2. Description of Dual-Lift Use Case

This section will provide a practical description of the dual-life use case on the NIST industrial wireless testbed. The use case involves two robots that pick up and move a metrology bar through the testbed using a leader-follower control implementation.

2.1. Use Case Overview

The use case is implemented using the robot operating system (ROS) nodes [2] to control the leader and follower robots. There is also a supervisor ROS node that coordinates the starting movements to pick up and preload the springs on the extendable metrology bar, which also makes a ball and socket connection to the robot's electromagnetic grippers. The bar is expandable with springs on the sphere ends to allow for large deviations in the leader and follower positions. Therefore, the robots can move within the circular plane without dropping the bar under the worst-case offset scenario. The experiments consist of the leader moving in a pre-determined path, typically a circle, and the follower using a velocity controller with an input stream of position information that is sent by the leader robot using a ROS topic. This leader-follower information stream is directly related to the performance of the follower. Figure 1 shows the testbed components, along with the next unit of computing (NUC) devices that act as a software access point (soft-AP), Ethernet-Wi-Fi adapters, and a wireless station for the interfering traffic sink. There are also infrared (IR) reflecting markers on each robot's end effector, which are used to track their position using a 3D camera tracking system, discussed in Section 2.2.

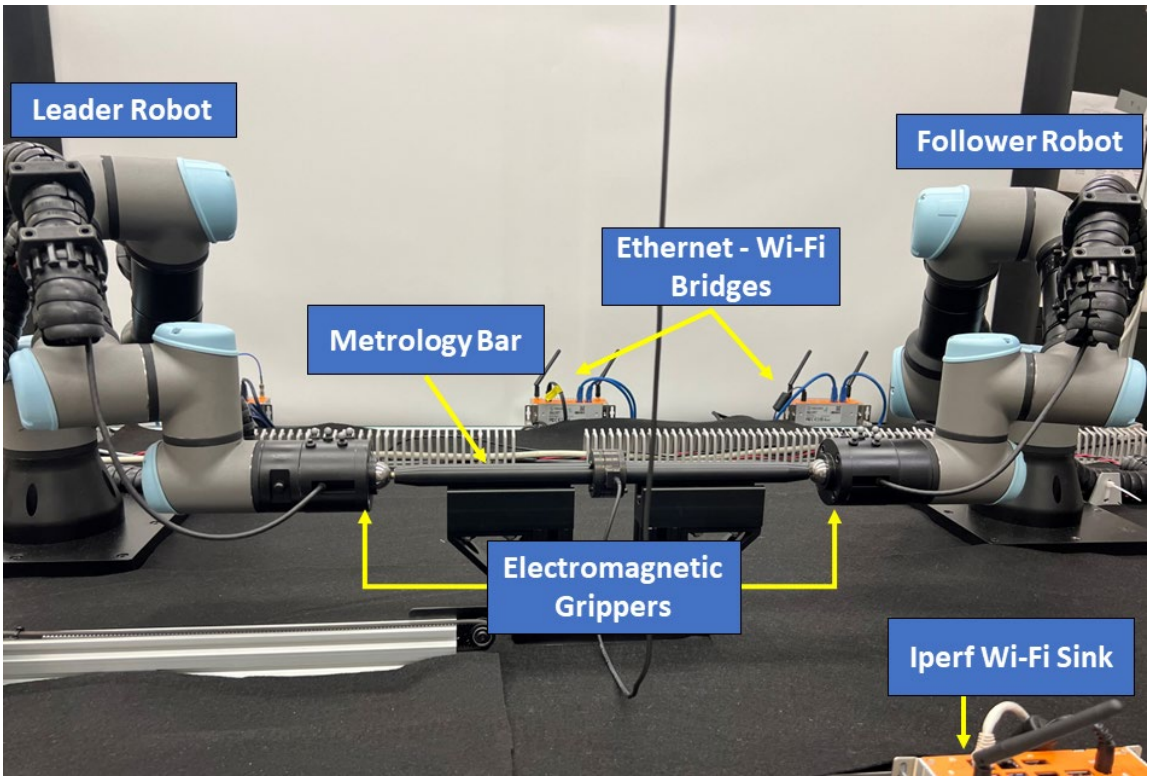

Figure 1. Dual-Lift Use Case Components

Figure 2 is the network diagram of the operational network, which includes the Robots, ROS nodes, wireless NUC bridges and access point, as well as the force-torque sensor located in the center of the metrology bar. The network diagram does not include the tap points for the latency measurement. The tap point to calculate the round-trip time (RTT) metric is located in the Ethernet connection between the leader ROS computer and wireless bridge.

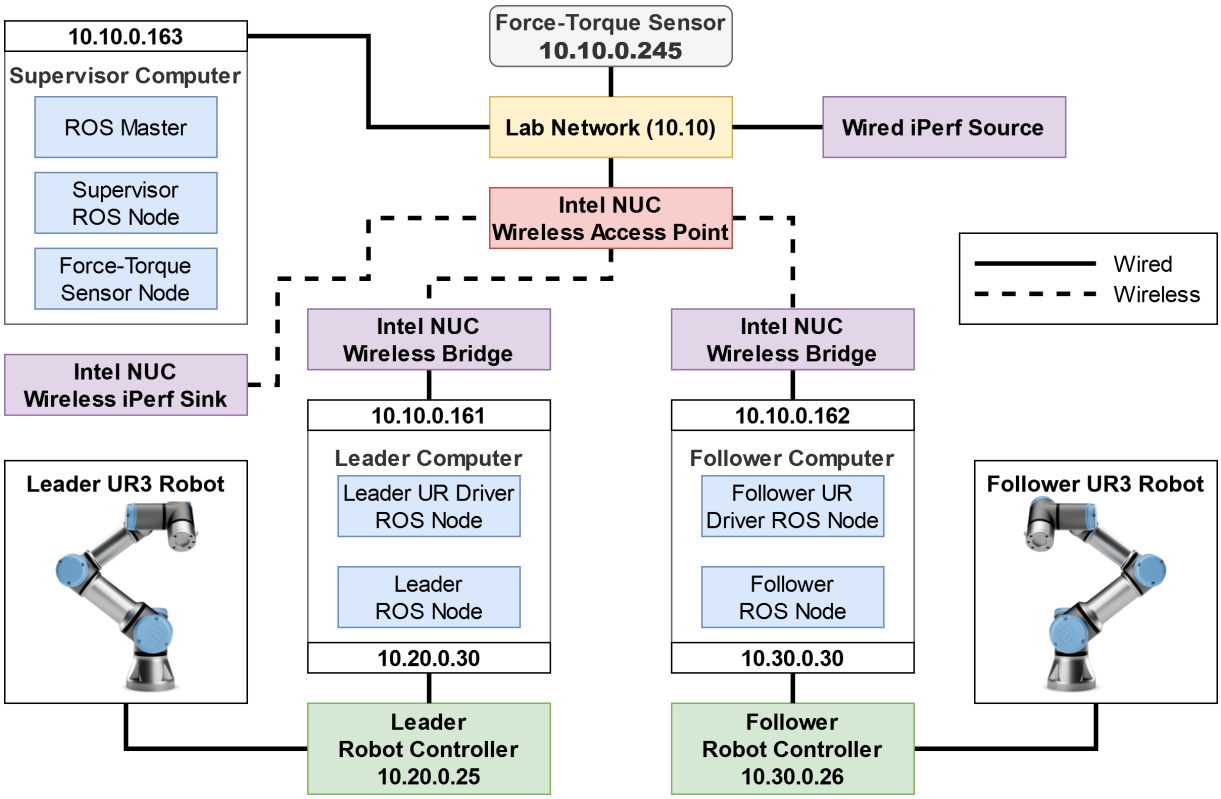

Figure 2. Network Diagram for Dual-Lift Use Case

2.1.1. Leader-follower implementation

The measurements of the use case occur during a circular path that both the leader and follower take. Other movement patterns could be performed, however the circular shape used in this experiment provides an error between the robots that is theoretically constant, due to the steady state velocity of the follower on the circular path. This constant error between the leader and follower occurs with ideal communication, thus when the communication link between the leader and follower is degraded, there is a noticeable physical impact on the follower's error. The error mentioned is the distance from where the follower should be, based on the leaders' current position, to where the follower is on the leaders pre-planned path. There is an offset during the movement of the robots, as the velocity controller used requires a difference in the current and target position for the follower to calculate the velocity it should take. There is no prediction control implemented on the follower to keep the motion of the follower as simple as possible for analysis purposes. When there are latency spikes in the communication link between the leader and follower, the follower can be seen to noticeably jerk and jitter. This testbed represents a real-time control use case, typical in many types of demanding industrial applications. This use case provides measurement capability to assess the latency and physical impacts, based on how well the communication system performs.

2.1.2. ROS nodes, robot description

There are several ROS nodes for the use case which include the leader, follower, supervisor, and object detection nodes. Each of the leader and follower ROS nodes connects one of their Ethernet ports directly to each respective robot controller, which runs an external control program to allow for each ROS node to directly control the corresponding robot. The supervisor node is used to start and stop the use case and manages the connection and the initial phase of picking up the metrology bar. Another Ethernet port on the leader and follower ROS nodes is used for communication to the network through exchanging ROS messages.

2.1.3. Message flows, leader-follower stream

There is a ROS topic named "desired_tcp_pose" that the leader node publishes at 125Hz, to which the follower node subscribes. This topic can be seen in Figure 3. The topic uses the Transmission Control Protocol/Internet Protocol (TCP/IP), which requires an acknowledgement (ACK) to return to the leader node before the next data message with the position information is sent. Therefore, if the latency between the data packet and ACK is greater than 8ms, which is the update period, then the effective update rate is reduced. This rate reduction poses issues for the real-time control loop. We have implemented software-based wireless time-sensitive networking (TSN) to provide more deterministic latency on the wireless link between the leader and follower nodes, while the rest of the services and actions are placed in the best effort queues. TSN will not be further discussed in this report. The work in [3] explains the wireless TSN implementation and experiments.

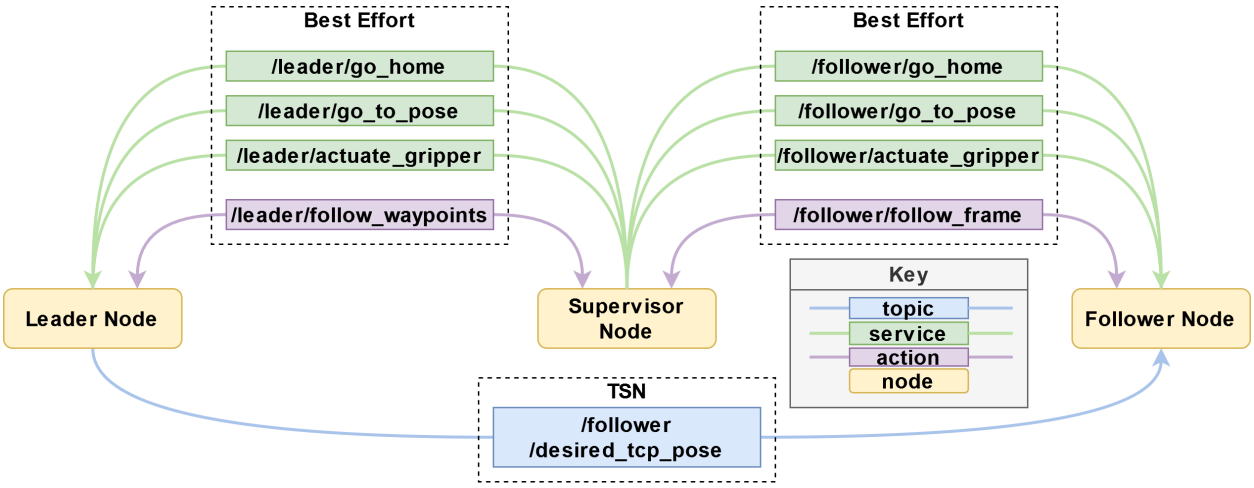

Figure 3: ROS Message Flows for Dual-Lift Use Case

2.2. Description of the Position Tracking Measurement System

The industrial wireless testbed utilizes four OptiTrack Prime 13W [4] cameras to track the positions of the robots. The cameras are overhead the testbed, such that the markers on each robot's end-effector are always in view during experiments to ensure accurate tracking. The cameras record each robot's position, then the end-effector positions are used to calculate the error between the leader and follower post-capture.

2.2.1. Physical 3D Tracking Camera System Setup

The cameras can be seen in Figure 4 below, which shows them overhead the testbed.

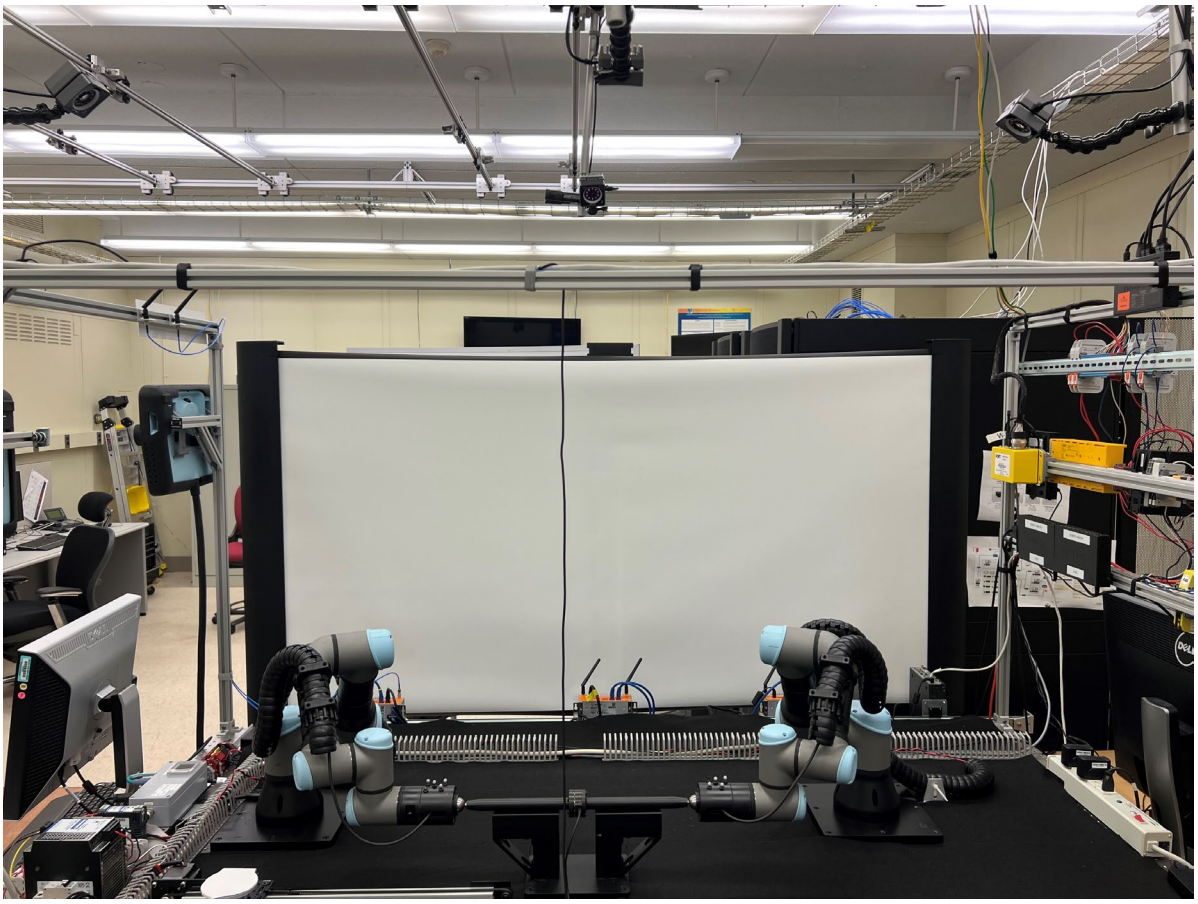

Figure 4: Industrial Wireless Testbed with Overhead 3D Tracking Camera System

2.2.2. Cartesian error magnitude between leader and follower KPI

For the experiments and uncertainty analysis, the error between the leader and follower is the key performance indicator (KPI) of the physical performance of the robots. This metric is directly related to the network performance, allowing for a correlation between the physical and network performance of the use case. The cartesian error magnitude KPI is the physical distance between the follower position and the required position, which is based on the leaders' current position on its path.

2.3. Description of the latency measurement system

2.3.1. Latency measurement devices

The ET2000 device from Beckhoff Automation [5] is an in-line industrial Ethernet multi-channel network tap device, that allows for accurate packet capture with low delays. It allows for real-

time Ethernet capture of four independent channels at a speed of 100 Mbit/s. The spec sheet claims a capture delay of less than 1 us from the transmitted packets, a timestamp resolution of 1 us, and an accuracy of 40 ns. The uplink port is sent to a capture machine that is time synchronized to a grand leader clock on the network, which is used for global time synchronization across nodes in the testbed. Below is a picture of the device used, in Figure 5.

Figure 5: ET2000 Network Packet Capturing Device

The Gigabit SharkTap device [6] is an in-line channel probe for a single Ethernet link, which copies the packets and sends them to the output line. There is no manufacturer data regarding the delay and jitter affecting network packets traversing the device; therefore, the uncertainty calculations are based on measured latency of captured packets. These devices have been used in past work, such as [3] and [7], however the ET2000 device is planned to be used in future works, due to significantly less uncertainty introduced into the latency measurements. A picture of the device is shown in Figure 6.

Figure 6: SharkTap Gigabit Ethernet TAP Device.

2.3.2. RTT KPI, from the leader-follower TCP stream.

The RTT KPI used for the analysis of the network performance is determined from the round-trip time of the network captures of the leader-follower TCP/IP stream. The RTT metric is calculated from the time it takes from the TCP data message sent by the leader, to when it receives the ACK back from the follower. This KPI demonstrates how well the network is performing for the real-time control application. Ideally the RTT should be less than 8 ms to ensure the 125 Hz update rate, as higher delays reduce the effective update rate. Since it is a TCP stream, the next data message must wait until the ACK is received or a timeout occurs.

3. Uncertainty calculation of Cartesian error KPI

To calculate the uncertainty of the leader-follower error KPI metric, a test was constructed to evaluate the OptiTrack camera's by using certainty positions taken in the use case. To accurately measure just the uncertainties introduced by the measurement system alone, the data collected was taken when the robots are static. We assume that the uncertainty of a static body is similar to a slow-moving body, as the speed of the robots is relatively slow compared to the update rate of the cameras, moving less than a millimeter in the 240 Hz capture rate of the tracking system. Since we are performing this test using statistical methods, the evaluated uncertainties are considered Type A.

3.1.1. Description of data collection for determining uncertainty

To perform the uncertainty test, the bar is picked up by the robots, and that position is averaged over 10 seconds and saved to calculate the offset between the leader and follower robots. Then, the robots move to the top of the circle, wait 10 seconds, then both robots move to several other points, as shown in Figure 7. The robots hold each position for 10 seconds. The red dot, labeled D, represents the point with the highest most expanded uncertainty, which is based on the maximum standard deviation values in the X, Y, and Z axes. For the data collection, twelve total points were collected at 10 second intervals for each point, such that the "circle" was repeated 3 times.

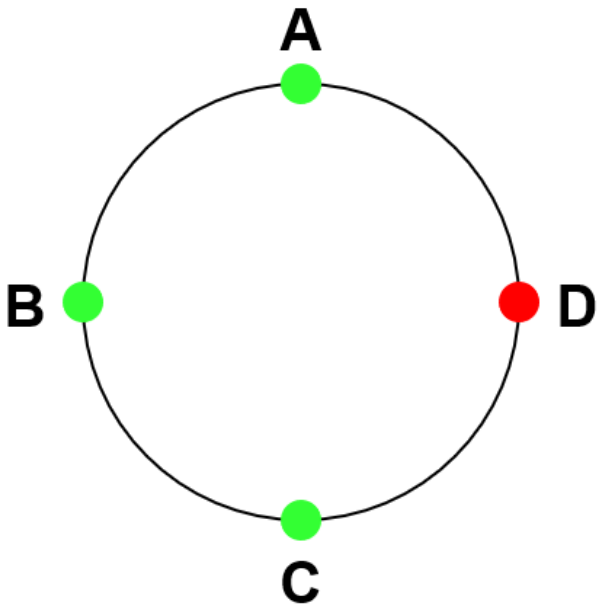

Figure 7: Static Points on Circular Path for Measuring Uncertainty

Below are tables of the maximum standard deviations calculated from all 12 points collected from the uncertainty experiment. It is observed that the Z axis data have more standard deviation than the X and Y axes. This is postulated to be the result of the cameras being approximately the same height in the Z axis, and the markers do not vary much in this axis, leading to more jitter compared to the X and Y axes. To obtain the worst-case uncertainty, the maximum values used in the tables below are the highest values out of the 3 measurements taken on each point of the circle for the experiment. The data selected for each point was sampled by using one in every ten points in the data set, to eliminate any correlation values between successive points. We used the highest standard deviation for each axes from the 12 points taken, to not underestimate the uncertainty. There were 191 points used to calculate each standard deviation, after cutting the first and last second at each point to eliminate residual motion from moving between the points.

Table 2. Leader's Maximum Standard Deviations for Points A to D in the Circle

Max Std Dev in each Axes (mm)	A	B	C	D
X	0.0079	0.0076	0.0081	0.0063
Y	0.0066	0.0057	0.0053	0.0057
Z	0.0142	0.00148	0.0143	0.0111

Table 3. Follower's Maximum Standard Deviations for Points A to D in the Circle

Max Std Dev in each Axes (mm)	A	B	C	D
X	0.0063	0.0069	0.0064	0.0083
Y	0.0072	0.0065	0.0053	0.0075
Z	0.0152	0.00171	0.0146	0.0182

We can see from Table 3, that point D for the follower had the most standard deviation, leading to the highest uncertainty, so we use this point as the worst-case for the measurement system's uncertainty calculation in the following section.

3.1.2. Resulting calculation of combined, and expanded uncertainty

To calculate the uncertainty, the measurement equation must be formulated. The error between the leader and follower takes into account the bar length to determine the cartesian error magnitude KPI. Below is the measurement equation for the leader and follower, where X_L, Y_L, and Z_L are the Cartesian positions of the leader, X_F, Y_F, and Z_F are the Cartesian positions of the follower, and X_o, Y_o, and Z_o are the offset values accounting for the bar. The offset values are calculated from the difference between the leader and follower positions in the X, Y, and Z axes when the bar is initially picked up before the leader-follower sequence starts. The measurement equation is seen below in Eq. 4.

$$Error = \sqrt{(X_L - X_F + X_O)^2 + (Y_L - Y_F + Y_O)^2 + (Z_L - Z_F + Z_O)^2} \quad (4)$$

Using Eq. 4 and the equations outlined in Section 1.2 with the six maximum standard deviations in Table 2 and Table 3 for point D, the combined standard uncertainty was calculated to be 0.01018 mm. Then, after calculating the effective degrees of freedom, v_{eff}, to be 751 and using the confidence interval of 99%, k was calculated to be 2.62. Finally, utilizing Eq. 3 yields the resulting expanded uncertainty, U, which is 0.027 mm. There is also a calculated output value bias of 2.56 mm from the measurement equation, due to the position where the offset of the bar is taken not matching the Cartesian error magnitude at Point D. Therefore, the resulting uncertainty, U, is represented as **2.56 ± 0.027 mm** at a 99% confidence interval. This result demonstrates low uncertainty for the 3D tracking system under test.

4. Uncertainty calculation of RTT KPI between leader and follower

The round-trip time calculation is performed by taking the time difference from the data packet from the leader, to when the acknowledgement packet from the follower is received back. This is measured using the ET2000 device or a SharkTap device, which is discussed in 2.2. The ET2000 has its collection port routed to a collection machine, and the packets are timestamped locally on the ET2000. Another method of calculating RTT involves the SharkTaps on the testbed, which are also routed to the collection machine. To measure latency accurately with the SharkTaps, multiple ports are required to be synchronized to a grand leader clock, which uses the IEEE 1588 precision time protocol (PTP) [8] over Ethernet to synchronize measurement clocks used for timestamping. The log file of the synchronization offset accuracy was taken, for both the clock on the Ethernet port and the physical clock on the machine. This data was used in calculating the uncertainty in the delay, along with the uncertainty introduced by the ET2000, specified in the next section.

4.1. Uncertainty of RTT from the ET2000, a Type B uncertainty

According to the manufacturer, the ET2000 has accurate packet capture, at 40 ns of "accuracy" and < 1us delay. It is not specified if this accuracy level is at a 99% confidence interval, but for this report, we assume the standard 95% confidence interval in this reporting to be more conservative. Therefore, by reversing the uncertainty calculation process, we can determine that the standard deviation of the data would have been approximately 20 us. This value is classified as Type B, as it was not determined by statistical means from observations in our work. Using the delay and accuracy reported from the Type B value, and the mean and standard deviations of the clock sync offsets from the collection machine, we can calculate the expanded uncertainty of the round-trip time. The measurement equation used is found in Eq. 7, with T_d representing the delay introduced by the ET2000 from the captured times of the data and ACK packets, T_{data} and T_{ACK}, which are timestamped within the ET2000 with the manufacture specified <1us delay component. The actual RTT KPI is calculated from the time difference of the ACK and data packets, shown in Eq. 5. What we can measure includes the offset and uncertainty introduced by the tap device, shown in Eq. 6. Lastly, the time introduced from the RTT measurement using the ET2000 device, which is denoted by M_{ET2000} in Eq. 7, which is a symmetric measurement equation with a zero bias. This equation was formulated by taking the difference between Eq. 6 and Eq. 5.

$$RTT = T_{ACK} - T_{data} \tag{5}$$

$$\overline{RTT} = (T_{ACK} + T_d) - (T_{data} + T_d) \tag{6}$$

$$M_{ET2000} = T_d - T_d \tag{7}$$

4.2. Uncertainty of RTT from the SharkTap, a Type A uncertainty

Since the SharkTap does not timestamp the packets locally, the timestamp used is from the PTP hardware clock (PHC) in the Ethernet port from the collection machine that is connected to the collection machine's measurement port, TAP. This PHC synchronizes with the main system clock (SYS) to maintain its time. In addition, SYS was synchronized to a grand leader clock (GCL) that synchronizes many nodes in the measurement system by using PTP. Therefore, there are three steps for synchronization that were made for the RTT packets captured: GCL to PHC on the PTP port on the collection machine, PHC to SYS, and SYS to PHC for the TAP's PTP clock. The measured RTT is calculated similarly to Eq. 6, but in Eq. 8 there is an additional clock synchronization error introduced by PTP time synchronization, denoted as E_{PTP}. Then, taking the difference between the measured and actual RTT (Eq. 8 – Eq. 5) yields Eq. 9, where $M_{SharkTap}$ is the time introduced from the RTT measurement using the SharkTap device.

$$\overline{RTT} = (T_{ACK} + T_d + E_{PTP}) - (T_{data} + T_d + E_{PTP}) \tag{8}$$

$$M_{SharkTap} = (T_d + E_{PTP}) - (T_d + E_{PTP}) \tag{9}$$

4.3. Results

To calculate the total expanded uncertainty, the standard deviations and means of each data set or reported values were analyzed. For the SharkTap, a latency test was performed by cycling traffic through the TAP and measuring the difference in time between the 10000 captured TAP packets and outbound packets using two ports on the collection machine. This test resulted in latency values that were introduced from the SharkTap device. The total error from PTP, E_{PTP}, in Eq. 9 was calculated by the addition of the offset from the grand leader clock to the clock on tapped Ethernet port, a PHC. The value of E_{PTP} is the summation of the errors mentioned previously mentioned in Section 4.2, labeled as $E_{GCL-PHC}$, $E_{PHC-SYS}$, and $E_{SYS-PHC}$. The resulting mean and standard deviation values for each variable can be seen in Table 4.

$$E_{PTP} = E_{GCL-PHC} + E_{PHC-SYS} + E_{SYS-PHC} \quad (10)$$

Table 4. Type A input values to calculate RTT uncertainty using the SharkTap

	T_d	$E_{GCL-PHC}$	$E_{PHC-SYS}$	$E_{SYS-PHC}$
Mean (ns)	69265	-0.03	-0.04	-0.26
Standard deviation (ns)	3225	121	244	266

We were able to calculate the delay in the received TAP and output packets to determine the $\boldsymbol{T_d}$. The clock error mean and standard deviations are the same for both the data and ACK packets for the TCP messages, as the point in the network the packets are collected is the same, and the length of the packets are not included in this time. We measured the time introduced by the SharkTap's TAP port using a traffic generation test, measuring the TAP and passthrough of the SharkTap on a singular device. The difference in time between the TAP and passthrough was taken to determine $\boldsymbol{T_d}$.

For the ET2000, the standard deviations and delay means introduced by the ET2000 are 20 and 1000 ns respectively. Using Eq. 7, the ET2000's combined standard uncertainty was calculated to be 28.28 ns, with no bias on the output value, as the 1000 ns offsets cancel to zero from the measurement equation, Eq. 7. The k value at a 99% confidence interval is 2.58. Therefore, the resulting expanded uncertainty is reported to be **0 ± 73 ns**. For the RTT uncertainty using the SharkTap, the combined standard uncertainty is 4592 ns with no bias on the output value, due to the symmetry in the RTT measurement equation, Eq. 9. The k value at 99% confidence is 2.58. Therefore, the resulting expanded uncertainty is reported at **0 ± 11849 ns**. The standard deviation from the TAP's latency and jitter is much higher than the clock synchronization error, since only 1.4% of the uncertainty value stems from the clock error components, E_{PTP}.

5. Conclusion

This report introduced how to calculate expanded uncertainty utilizing the GUM process, provided a description of the dual-lift use case, and presented the process of calculating the uncertainty for the physical and network measurement KPIs. The measurement systems have been shown to be accurate, with the expanded uncertainties of ±0.027 mm for the Cartesian error magnitude using the 3D tracking system and ±73 ns for the RTT using the ET2000 at a 99%

confidence interval. Note that the SharkTap's expanded uncertainty at ±11849 ns has much higher uncertainty compared to the ET2000, however this value is sufficient for determining latencies in the network in the 100's of microseconds. The wireless RTT latencies are typically above 1 ms. Overall, the data produced in experiments using these measurement systems have shown relatively small position and latency uncertainties, compared to the movements and latencies in communications from the operation of the robots.

References

[1] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. Evaluation of measurement data | Guide to the expression of uncertainty in measurement. Joint Committee for Guides in Metrology, JCGM 100:2008. URL: https://www.bipm.org/documents/20126/2071204/JCGM_100_2008_E.pdf/cb0ef43f-baa5-11cf-3f85-4dcd86f77bd6.

[2] Stanford Artificial Intelligence Laboratory et al. (2018). *Robotic Operating System*. Retrieved from https://www.ros.org

[3] R. Candell, K. Montgomery, M. Kashef Hany, S. Sudhakaran and D. Cavalcanti, "Scheduling for Time-Critical Applications Utilizing TCP in Software-Based 802.1Qbv Wireless TSN," *2023 IEEE 19th International Conference on Factory Communication Systems (WFCS)*, Pavia, Italy, 2023, pp. 1-8, doi: 10.1109/WFCS57264.2023.10144232.

[4] *Prime 13W*. (2023). https://optitrack.com/cameras/prime-13w/

[5] *ET2000 | Industrial Ethernet multi-channel probe*. (2023). https://www.beckhoff.com/en-us/products/i-o/ethercat-development-products/elxxxx-etxxxx-fbxxxx-hardware/et2000.html

[6] *SharkTap 10/100/1G features*. (n.d.). http://midbittech.com/files/GigE/Details.htm

[7] Candell, R. , Montgomery, K. , Hany, M. , Sudhakaran, S. , Albrecht, J. and Cavalcanti, D. (2022), Operational Impacts of IEEE 802.1Qbv Scheduling on a Collaborative Robotic Scenario, 48th Annual Conference of the Industrial Electronics Society IECON 2022 Conference, Brussels, BE, [online], https://doi.org/10.1109/IECON49645.2022.9968494

[8] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," in *IEEE Std 1588-2019 (Revision ofIEEE Std 1588-2008)* , vol., no., pp.1-499, 16 June 2020, doi: 10.1109/IEEESTD.2020.9120376.Taylor BN (2011) The current SI seen from the perspective of the proposed new SI. *Journal of Research of the National Institute of Standards and Technology* 116(6):797. https://doi.org/10.6028/jres.116.022