**Multi-Agent Synchronized Collaborative Assembly Replication (MASCAR) Testbed Design Notes**

Jeremy Marvel, Megan Zimmerman, and Shelly Bagchi

**Abstract: The design considerations and measurements for a purpose-built robot testbed for human-robot interaction metrology is discussed. Rationale are given for each design choice, as are iterations on designs as challenges arise during the construction process and early human subject trials.**

[**1. Introduction 2**](#_iql00dhbb0ix)

[1.1. Scope and Impetus 3](#_w8sgvr3rtidk)

[1.2. Intended Use of the MASCAR Testbed 5](#_axji1614eiv)

[1.3. Associated Documents and Files 5](#_d0cwb4ryk2e)

[**2. Physical Components 6**](#_kmc85nte9ce2)

[2.1. Mobile Testbed Table 6](#_z0palo9ve6m7)

[2.1.1. Mobile Table Base 6](#_94jfah5x5yav)

[2.1.2. Tabletop Surface 8](#_kz3ktzrf8nr3)

[2.1.2.1. Tabletop Version “0” 8](#_mhr5cdijbeg0)

[2.1.2.2. Tabletop Version 1 9](#_ap9b4zs8zxhk)

[2.1.2.3. Tabletop Version 2 11](#_iaxbruna8q59)

[2.1.2.4. Tabletop Version 3 11](#_49lo0m2b2h0x)

[2.1.2.5. Tabletop Version 4 11](#_m8f2nd7ue7of)

[2.1.3. Power 13](#_yc4msiyq12yd)

[2.1.4. Communications 13](#_a1csuc3f45fr)

[2.1.4.1. Network Communications 14](#_fqkf4fpr18zl)

[2.1.4.2. Digital I/O 14](#_tlpyxo9dx7uc)

[2.1.4.3. Serial Data Communications 15](#_jizzs99z83jb)

[2.1.4.4. Audio Input 15](#_a12dm8wl9qm7)

[2.2. Robotic Platform 15](#_qfqr4wmdt68c)

[2.2.1. Robot 15](#_dwgnngwgtvmh)

[2.2.2. Ancillary Controller 17](#_3nnx82lz5xrr)

[2.2.3. Gripper 17](#_w8ip7pf0f5j0)

[2.2.3.1. Gripper Selection 18](#_lsjsmvdo00kr)

[2.2.3.2. Fingertips 18](#_veiv7zl5ywos)

[2.2.4. Gripper Control 20](#_g4fg7bt1gl1g)

[2.2.5. Hand-Guided Control Interface 22](#_v9rakxodna46)

[2.2.5.1. Handle Version 1 22](#_3fnov5ennc7l)

[2.2.5.2. Handle Version 2 23](#_8vhhz8f0k8q)

[2.2.5.3. Handle Version 3 23](#_s284yf3u01m2)

[2.2.5.4. Handle Version 4 25](#_nq2z4h3lfgay)

[2.2.5.5. Handle Version 5 26](#_p77rqpkai7bi)

[2.3. Computing Devices 28](#_5ud004plg1w2)

[2.4. Task Components 29](#_iypdy4wmx21m)

[**3. Sensors 29**](#_5y0hjjt9vv3k)

[3.1. Audio 30](#_m6ec81yufbv)

[3.1.1. Postioning 30](#_d6i9s5b8rb47)

[3.2. Vision 32](#_g37u3lo5odzc)

[3.3. Object/Sensor Tracking 32](#_fvcib5rpch2t)

[3.3.1. Rigid body motion capture 32](#_psfdyk9bym3w)

[3.3.2. Human motion capture 33](#_7cng32o0dmgq)

[3.4. Sensor Mounting 33](#_khc0iefjwrp0)

[**4. Communications and Control 34**](#_5oru0dmt0u2d)

[**5. Data Collection and Storage 35**](#_xxf2r6jhkyc2)

[5.1. Data feed collection and synchronization 35](#_jbeiozahjx4a)

[5.1.1. Data Collation 35](#_d0x2x0ig5tb4)

[5.1.1.1. Robot Data 35](#_on2s0itp9kdh)

[5.1.1.2. Gripper State 35](#_u6q9zvqeouqy)

[5.1.1.3. Video 36](#_5e59gg9yxizw)

[5.1.1.3.1. Black Magic 36](#_i5ofyobdzi4t)

[5.1.1.3.2. External Display Feeds 37](#_jti18rdwlu8e)

[5.1.1.4. Audio 37](#_xfumur9pcb09)

[5.2. Storage 37](#_3i2uh3i0m11)

[5.3. File Structure 37](#_px3vnjg6nhds)

[5.4. Tagging 38](#_j1s6ska05ja7)

[5.4.1. Labstreaming player 38](#_vzkqh7g72jj7)

[5.5. Future work 38](#_fcrbi6wjnghw)

[**6. Task Components 39**](#_28qz03boqhwt)

[**7. Digital and Physical Twins 39**](#_rl98b5w2kcvv)

# Introduction

Repeatability and reproducibility continue to present fundamental challenges within the human-robot interaction (HRI) research community. Overwhelmingly, publications surrounding data sets, designs of experiments, and even systems descriptions lack the detail and metadata sufficient for the replication and extension of HRI research to one’s own environment and/or research domain. While the existing documentation is important and meaningful, it is not well-suited for generalizability and reproducibility.

To address this limitation, the National Institute of Standards and Technology (NIST) is working to develop a reporting methodology that establishes the vital information necessary to enable research replication, data scalability, and result generalization. This report reflects an application of this methodology to a new HRI testbed designed and built by NIST to serve as a data collection and reporting apparatus validation tool. In particular, details are provided that thoroughly document design decisions, product[[1]](#footnote-0) selection criteria, impetuses and results of iterative changes, and early steps to verify and validate the integration of a new testbed for human-robot collaboration research.

The human subjects testbed is intended to provide a traceable and repeatable means of collecting human-robot interaction data for the generation of high-quality datasets of human-robot interactions in manufacturing applications. “Traceability” in this context reflects the expectation that all components of the testbed, their capabilities (and limitations), and the data they produce (including measurement uncertainties) are all thoroughly documented. “Repeatability” specifically refers to providing the metadata (including formatting and installation details) in the data sets that captures the nuances of our specific installation (configuration, settings, environmental conditions, etc.) such that researchers can exactly replicate, model, or even modify NIST-generated data sets.

The testbed itself consists of an adjustable-height, mobile cart with a machined table top. On this cart is mounted a small industrial robot system (manipulator + controller) with a two-finger servo-driven gripper, and a NIST assembly task board[[2]](#footnote-1). The robot is controlled via a number of different interfaces, including the manufacturer’s default teach pendant, a commercial off-the-shelf (COTS) robot controller with a human-machine interface (HMI), and custom-built interfaces.

In this section, we describe the research questions, thought processes, and early design decisions that inspired the construction of this testbed and the subsequent data sets. In Section 2, the physical components that comprise the structure and functional capabilities of the testbed are described. Section 3 discusses the tools and mechanisms by which data is collected, including the selection, placement, and configuration of sensors and data-recording mechanisms. The propagation of control and feedback signals are presented in Section 4. And Section 5 discusses the mechanisms by which the physical, mechanical, and electronic components were integrated to produce a cohesive testbed.

## Scope and Impetus

In line with the intended goal of creating a testbed that supports HRI research repeatability and data set generation and generalizability, NIST’s Performance of Human-Robot Interaction project set for itself, a fiscal year 2025 critical milestone: “*Design and document a testbed specifically intended for HRI teaming research, featuring portability and adjustability to accommodate the variability of humans and tasks. This testbed is intended to be modular to accommodate the rigid placement of dedicated sensors and tools for human- and task-monitoring applications*.” In addition to addressing the replication crisis in HRI research, this critical milestone also addresses two fundamental limitations of current laboratory and testbed designs that perpetually plague the NIST robotics program:

* Throughout the published literature, it is evident that testbeds used for HRI research are heavily-modified general-purpose workbenches and tables with sensors on tripods or mounted to the laboratory walls. Research at NIST is no different in this regard.
  + This forces testbed designs to be centered on conforming to and compensating for limitations of the existing laboratory spaces, rather than being focused on fulfilling the intended purpose of the research activity. By taking a bottom-up approach to design and documentation, the new testbed would be constructed based on first establishing the needs of addressing the research question, rather than being limited to the more typical top-down approach of settling for whatever is on hand.
  + Limitations in the existing infrastructure have historically created challenges for human subject studies. By first clearly establishing the research space requirements needed for successfully supporting activities involving the general public, existing laboratory spaces can be thus purposely restructured and reconfigured to be optimal for research quality.
  + Similarly, limitations of existing spaces often necessitate customized solutions to adapting and repurposing equipment rather than specifically acquiring and leveraging the tools and components necessary for the research task. While some existing sensors and data collection tools may be used, their inclusion is mandated by the specific metrology needs rather than convenience. Commercially available robots, equipment, control interfaces, sensor systems, and software tools would be prioritized given the typical wealth of available documentation provided by manufacturers and service providers, while custom-built solutions would be relegated to those cases where established products are unavailable.
  + Tolerances and uncertainties need to be known *a priori*, rather than measured and justified *post hoc*. The testbed would also need to be specifically designed to accommodate robot and sensor calibration and registration using established means[[3]](#footnote-2).
* Because of their nature of being purpose-built for addressing specific research questions, the design and documentation of testbeds frequently focuses only on those aspects that are distinct from preceding work. However, often these design decisions are not created in a vacuum, and design considerations that were inspired or derived from prior research are often omitted. This historical context is crucial to understanding the context of design decisions, and should thus be included in the documentation.
  + As part of the research efforts toward HRI performance metrology, NIST partnered with Michigan Technological University (MTU), and assisted with their design of experiment, including the setup and configuration of their own research testbed. We drew inspiration from their setup, and improved upon their design, and drew inspiration from their setup and improved upon their designs as limitations and potential improvements were identified.
  + Significant efforts at NIST including the compilation of datasets consisting of object definitions and sensor data–such as the datasets associated with the 2011 and 2012 Solutions in Perception Challenges[[4]](#footnote-3), and the Manufacturing Objects and Assemblies Dataset (MOAD[[5]](#footnote-4))--created formalized methods for compiling and documenting application-relevant information for training models and evaluating system performance. The MASCAR testbed draws from these methodologies, including the documentation of pertinent metadata and configuration information, to continue the tradition of traceability and providing meaningful ground truth for novel algorithms.

## Intended Use of the MASCAR Testbed

How humans interact with a robot

How interfaces impact task performance (what aspects of those interfaces have the largest impact on performance?)

Which tasks are impacted most by interface design?

Which tasks are generally challenging?

Buthroyd Duhurst metrics

User preferences

Impacts on users (mental physical effort, etc.)

Quantifiable user actions

COrrections being made

Number of button pushes

Scenario completion time

Software usability

Ergonomics scales?

System usability

Survey selection tool leveraging the scoring criteria from Saad et al[[6]](#footnote-5)

Survey tools

NASA TLX

Programming the robot to complete a simplified assembly task using small components

## Associated Documents and Files

As the MASCAR testbed was designed and constructed, all aspects were documented.

CAD models

SOPs for performing tasks (such as installing/configuring software, troubleshooting, etc.)

Custom code bases

Risk assessments for safety verification and IRB approval

Bill of materials, including specific products and links used in the construction of the MASCAR testbed. Provided for informational purposes only.

<https://github.com/usnistgov/MASCAR/>

../ <root>

<CAD Files>

<Digital Assets for Modeling>

<Lab Furniture>

<Robot Models>

<Testbed Components>

<Design Documentation>

<Human Study Taskboard>

<Safety>

<Human Study Testbed>

<Universal Robots>

<Source Code>

<Arduino and ESP32 Code>

<Python Code>

<URCaps>

# Physical Components

The MASCAR testbed consists of several components that serve specific functions for the collection and support of human subject data. In this section, we discuss the design considerations and iterative changes made to the various physical components that contribute to the overall construction of the testbed.

## Mobile Testbed Table

Although the testbed consists of more than just a single workcell for robotic assembly applications, the collection of HRI data is centered on the interactions at a modular configuration consisting of a wheeled table, a customized table surface, and ancillary equipment necessary to support the functionality of application execution and data collection.

### Mobile Table Base

The NIST HRI laboratory is in a near-constant state of reconfiguration to accommodate changes is the research being conducted. Robots and equipment are moved around as they are needed, and even permanent fixtures can be repurposed when required.

Because of this, it was necessary to have a mobile platform on which the necessary equipment for the human subject study could be centered. The requirements for this table were as follows:

* **Portability**. The table needed to have locking casters such that it could be easily moved to where it was needed, and then locked in place to provide a stable platform. These casters (and possible additional locks) were not necessarily required to be stock with the table, but could also be after-market additions if required.
* **Size Limitations**: In addition to being mobile, the testbed table should have a profile small enough to allow for easy transport through narrow (~1.0m) doorways and aisles, and be able to be stored in a way that was not obtrusive to other traffic and/or equipment in the lab. To this end, it was determined that the table should not be larger than 1.5m (4.92ft) by 1m (3.28ft).
* **Height Adjustability**: Given its intended purpose of human-centric robotic assembly applications, the table was required to be height adjustable to accommodate a spectrum of participant heights. As such, it was necessary to have a base that could be easily modified to adjust the table surface to a height most comfortable for the participants. Assuming 50th percentile male and female adult participants, the height range of the table surface must be in the range of 0.96m (37.8in) to 1.10m (43.3in) to put the surface roughly in line with the elbows. Fine granularity of adjustment was preferable, but coarser adjustments were acceptable if needed. Similarly, automatic, electromechanical adjustments were preferable, but manual adjustments were also acceptable provided 1) it did not involve taking components off of the table (including disassembly of the table, itself), 2) could be repeatable, and 3) could be completed by a single person with minimal upper body strength.
* **Weight capacity**: The table base was required to support the mass of a robot and its controller, a customized tabletop surface, secondary controllers/computers for alternative interfaces, power distribution, communications, sensing, the robot’s tooling, and any components for the collaborative assembly task (task boards, fixturing, parts, etc.).
* **Rigidity:** With its intended purpose of supporting a metrological testbed consisting of a small, industrial robot, the table was expected to be relatively stiff to prevent unwanted movement of the table surface while the robot was moving or while the human operator was interacting with table-top components and equipment. A structural stability to allow less than 1cm lateral or anteroposterior motion from baseline was required.

Multiple table base options are commercially available, including the option of attaching powered, adjustable table legs onto a mobile base. For the sake of simplicity, however, a commercially available solutions was chosen to serve as the basis for the robotic workcell.

A 7-gauge steel, mobile table base with locking casters (see bill of materials for part numbers, link, and cost) was selected to serve as the mobile platform for the testbed. The selected base did not come with a table surface, which reduced the amount of waste given that a custom table surface was to be used. The base itself was 1.2m (47.24in) by 0.7m (27.56in), and was intended to support a table surface of width 1.22m (48.0in) by a depth between 0.76m (30in) and 0.91m (36in). The base surface could be manually raised and lowered using a low-torque crank to be between 0.76m (30in) and 1.07m (42in).

On paper, this testbed met all of the selection criteria originally established. The size and height adjustability readily allow us to easily move, store, and reconfigure the testbed for comfort and safety. However, two significant issues were encountered during the construction of the testbed that required additional modifications to be made:

1. The table base did not come with any shelving on which to mount the robot controller and ancillary equipment.
   1. The solution was to install shelves made of acetal resin, reinforced with aluminum crossbeams
2. While the casters lock to prevent rolling, the caster swivels do not. This allows the table to rock in an oscillatory manner while the robot is accelerating and decelerating, and people leaning on the table can also cause table surface motions within a set area.
   1. The solution was to install floor locks to lift the table up off the casters and onto stable legs.

### Tabletop Surface

With the table base thus selected, a suitable table surface on which to mount the robot and the assembly task components had to be constructed. Given the metrological nature of the testbed, accuracy, precision, and repeatability were required for placement of objects and fixtures on the tabletop surface. However, no suitable table surfaces were commercially available. As such, a customized table surface was constructed that would allow us to reliably mount equipment, component parts, and sensors for the collaborative application.

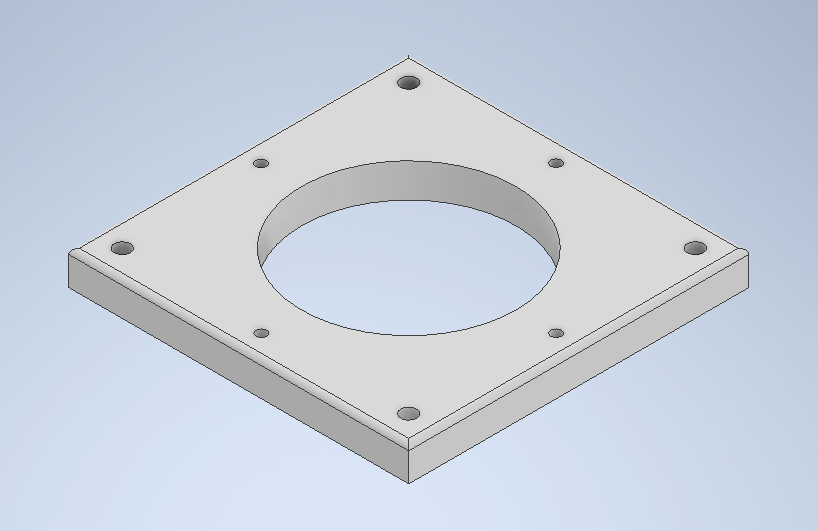
#### Tabletop Version “0”

A suitable tabletop design could not be fabricated until we knew how and where we wanted the robot, sensors, and ancillary equipment mounted. Initial efforts to design on paper were ineffective, so a proxy tabletop was constructed that would allow us to test different configurations and designs.

This Version “0” tabletop consisted of a 24-inch x 48-inch sheet of ½ inch plywood, slightly smaller than the table base’s intended tabletop surface. The plywood was painted green for the eventual application of green-screening video feeds to remove background elements, and an epoxy resin clearcoat was added to increase the table’s durability, ensure a smooth texture, and self-level the tabletop surface. The tabletop was secured to the table base using six wood screws fastened from underneath.

A two-state light tower (a green light that turned on when the robot was powered on, and a red light that turned on when the robot was moving) was added to the table for operator awareness of basic safety functionality.

An aluminum mounting plate was machined so that we could easily move the robot around before settling on a final location, and then secured to the table surface using four wood screws. One of the table’s long edges was selected as the “operator side,” which indicated where the operator was expected to be located. Early experiments determined that the ideal location for the robot was on the left edge (relative to the operator side) of the tabletop surface such that the robot could be readily stowed out of way while the operator was working in the shared work volume, while also providing as much tabletop real estate for unencumbered operator activity. Additionally, the majority of the tabletop surface would have predefined mounting points such that additional fixtures and components could be installed relative to the center of the robot’s base coordinate frame.

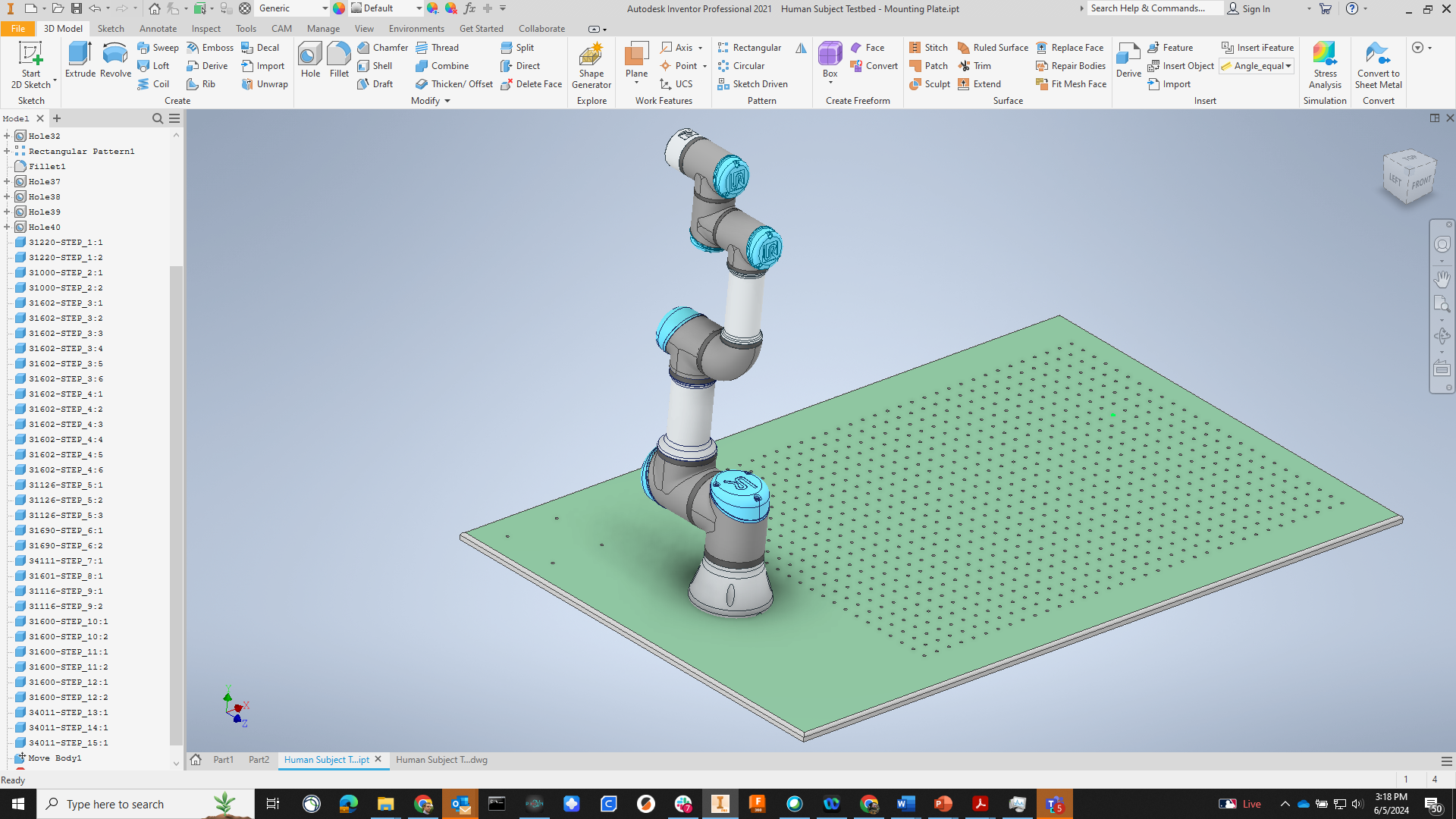


This plywood surface would be leveraged for short-term use while a suitable Version 1 was being manufactured.

#### Tabletop Version 1

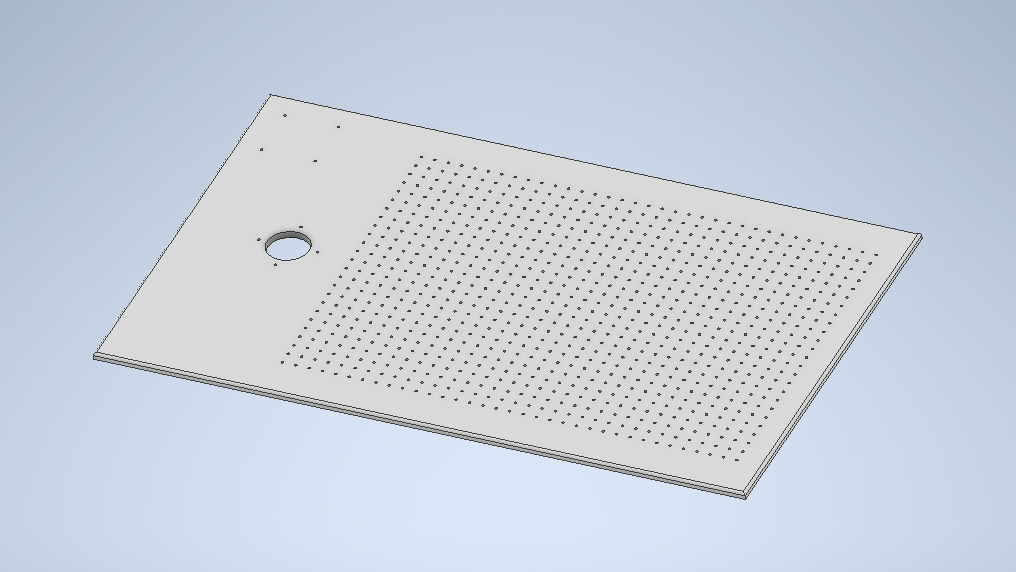
Following the early trials leveraging the plywood surface

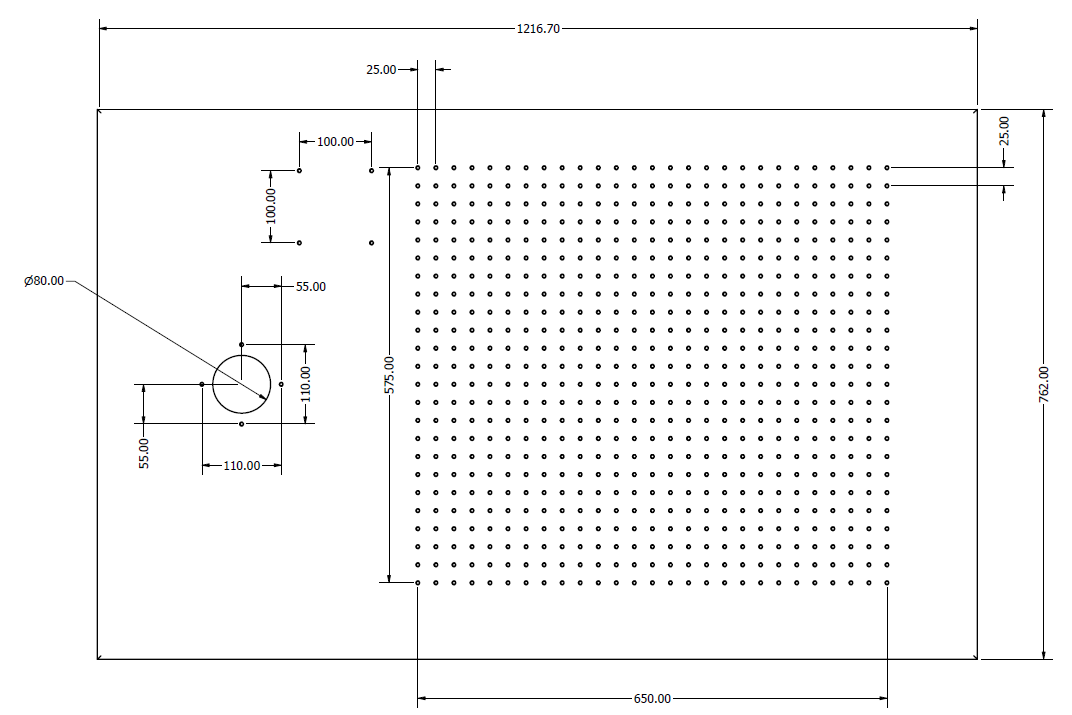
The initial design consisted of a regular grid of drilled-and-tapped M6 holes at 25mm intervals for rigid placement of surface-mounted elements. Fixtures and individual components could then be rigidly attached to the table using these holes, and additional plates could be added if different hole patterns were required. This grid of tapped holes would constitute the majority of the table surface, and a largely bare area was added to one edge of the table to accommodate the pre-defined and immutable mounting of the robot along the medial axis of the lengthwise plane.



Inspired by the general-purpose testbeds in several of the NIST laboratories leveraging optical breadboard with 6mm tapped holes,

* Machined aluminum top for strength, durability, and consistency
* Drilled and tapped holes in 25mm grid pattern to allow for
* Pre-drilled and tapped mounting for robot
* Pre-drilled and tapped holes for registration plate to define origin of table
* Anodized green for green-screening





TO DO: Add dimensional drawing and give transformations from origin to robot base, origin to table corners, and origin to tapped grid

#### Tabletop Version 2

* V1 table top colored flat black to reduce reflectivity and transfer of color to assembly objects with lambertian surfaces

#### Tabletop Version 3

Pilot human trials indicated that the placement of the robot on the left-edge of the tabletop surface were not as convenient for the programming task as originally thought. The working volume was overly limited by the relatively short reach of the robot arm, and the use of fixtures to hold pre-assembled parts consumed most of the robot’s available volume.

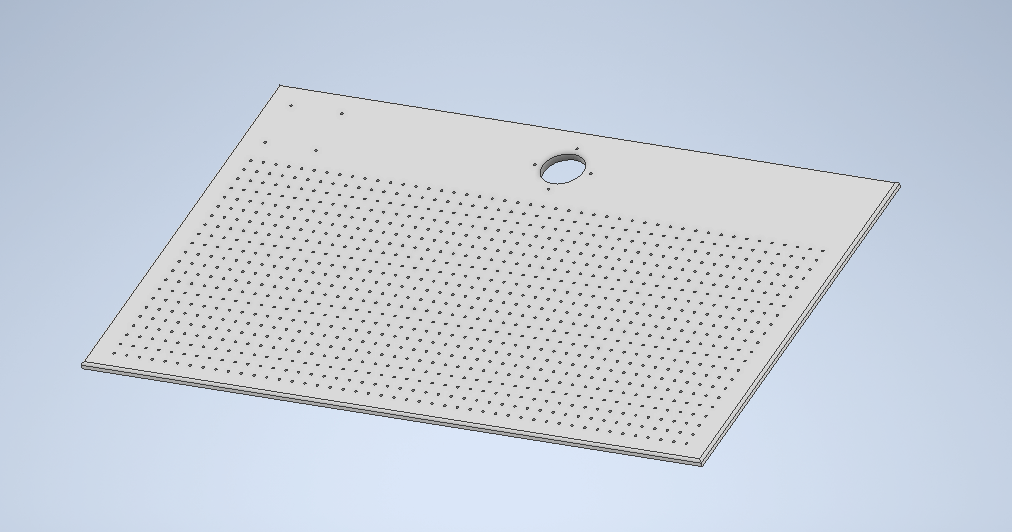
* Modification of V2 to place the robot central to the long edge of the table, toward the “back” to allow for a larger effective work volume.

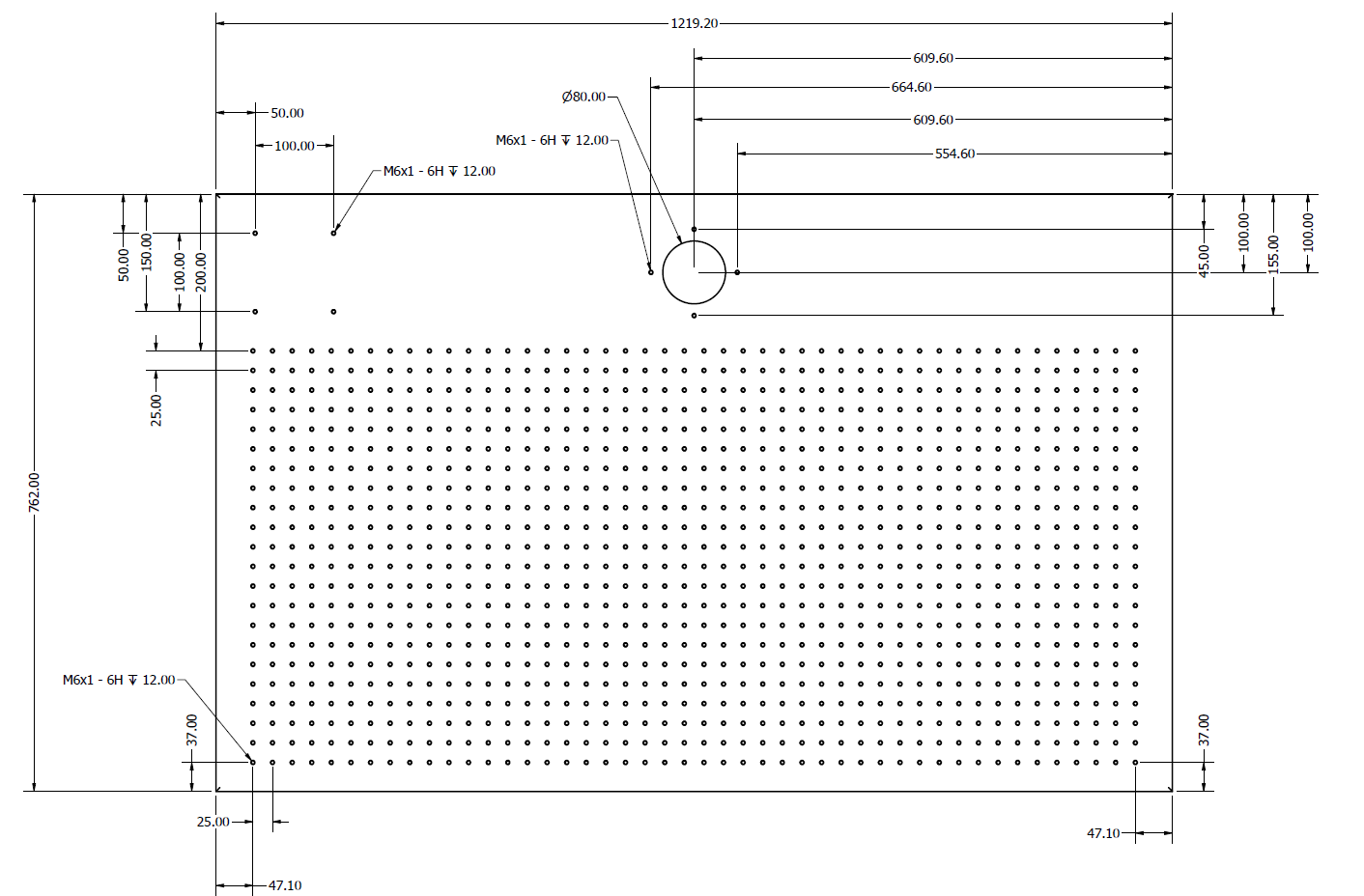
#### Tabletop Version 4

Following the early trials levera

V4

* Grid pattern extended across the entire width of the table.
* Table anodized black for more robust finish.





Sensor armatures

Wanted repositionable and adjustable, with enough support for multiple cameras/microphones/sensors

Three options being explored:

Rigid arms that connect to the table

More stable, less flexible in terms of configurations

Modular mounting kits with ball sockets

More flexible in terms of configurations, less stable

Tripods

’ Alternative option: Externally-mounted sensors

(not rigid relative to testbed surface, but not subject to wobble)

### Power

Some of the audio and video equipment is sensitive to fluctuations in input power, so it is necessary to integrate a power supply system that conditions the input voltage and filters out noise in the power line.

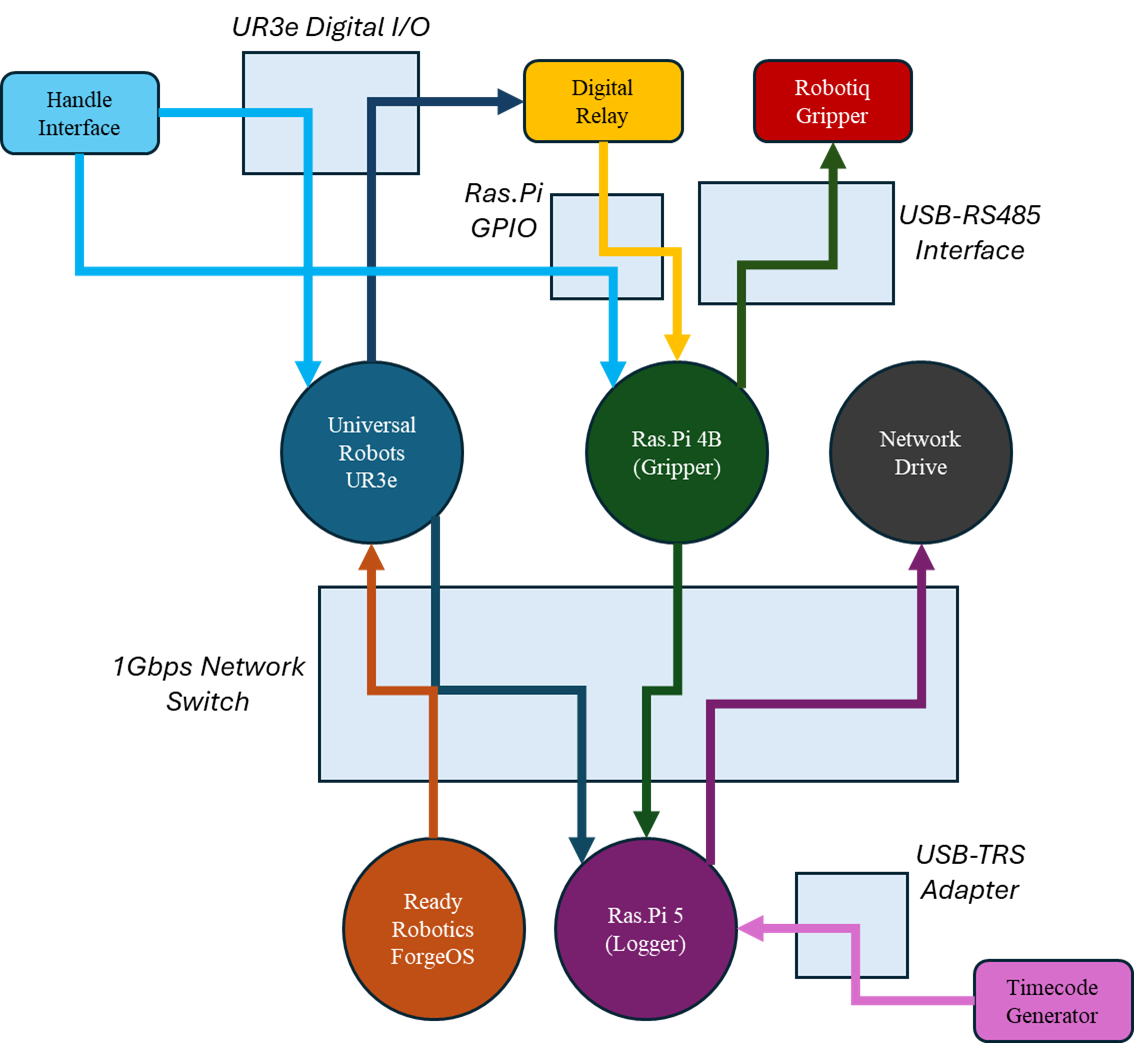
Studio-grade power conditioner and surge protector added to the system

Serves as a centralized power distribution system, with 12 filtered AC outputs, and voltage and amperage metering

Mounting on the testbed table also allows for power cords to be minimized, reducing the risk of additional noise being added after the power had been conditioned.

### Communications

Multiple interfaces were used throughout the testbed to facilitate the exchange of information between components. Some of these interfaces are intrinsic to the equipment being used (e.g., the Universal Robots e-series controller has several digital inputs and outputs that can be used to send information, control digital devices, or receive sensor data), while other interfaces are external and are used to send and convert information between devices. In this section, we will discuss the various communication interfaces used throughout the MASCAR testbed.



The primary testbed connectivity and communication chart. External computers and sensors are not shown here as they are not permanently connected to and associated with the MASCAR testbed.

#### Network Communications

The HRI Laboratory at NIST has leveraged a lab local area network (LAN) for communicating with robots and ancillary equipment. This LAN does not have Internet access, and is air-gapped from the NIST network for safety and security reasons. For the MASCAR testbed, a separate LAN was set up using a 24-port 1Gbps Ethernet switch, and would be used only for inter-process communications between computing devices used for the MASCAR assembly application and data collection. This network switch was kept separate from the laboratory’s existing LAN to limit possible noise and bandwidth issues with multiple computers sending and receiving data.

Currently, the dedicated MASCAR LAN supports only wired networked devices. A wireless component is planned for future updates to enable logging and data exchange to and from devices that communicate via 802.11 protocols.

#### Digital I/O

Both the robot controller (see Section 2.2.1) and an ancillary micro-computer (see Section 2.2.4) leveraged their onboard digital I/O for sending and receiving binary signal data to trigger specific actions for the MASCAR application. The controller commands the motions of the attached robotic manipulator, while the micro-computer is used to interface with an integrated robotic gripper (see Section 2.2.3) and command grasp/release (close/open) behaviors.

Switches on a robot-mounted handle for direct robot control (see Section 2.2.5) triggered different signals on the digital I/O of the robot controller and the micro-computer. A signal going into the robot controller triggered a gravity-compensation mode on the robotic manipulator, allowing the operator to physically drag the robot into different poses. A separate signal going into the micro-computer resulted in the gripper being commanded to open or close. A separate signal from the robot controller was used to actuate a digital relay, which acted as a virtual button feeding a signal into the micro-computer as a secondary gripper actuation command.

#### Serial Data Communications

The servo-driven gripper communicates through a recommended standard (RS) 485 physical layer serial interface using Modbus protocols. The micro-computer used to command gripper actions, however, does not natively support RS485, so a separate adapter had to be used to convert digital signals from one of the four universal serial bus (USB) ports built into the micro-computer main board.

#### Audio Input

For the purpose of data synchronization and alignment, the MASCAR testbed leverages timecode generators (see Section 4) that create audio signals with embedded timestamp information. Each timecode generator can interface with a single device using a variety of different audio inputs, and is synchronized with a centrally-controlled application that communicates via Bluetooth. The audio coming from the timecode generators are higher-voltage, line-level signals, and require “line-in” interfaces on the devices taking the timecode data.

While the computers used in the HRI laboratory all have a 3.5mm TRS audio jack, the audio inputs are used for low-voltage “microphone input” signals, and as such cannot accommodate the line-level outputs from the timecode generators. Separate USB-based audio interfaces with line-line ports thus had to be used to read and record the signals coming from the timecode generators.

## Robotic Platform

For the purpose of this testbed, a robotic arm with a gripper capable of acquiring, holding, and placing parts within the application work volume is required. This section describes the selection criteria for choosing which platforms to leverage for the MASCAR testbed, and iterative design and integration changes that occurred during the design and construction of the testbed.

### Robot

The robot is a central component to the HRI application, and must be intended for use with and around humans. Similarly, given the relative small working volume of the MASCAR table surface, the robot does not have to be particularly large (in fact, for operator safety purposes, the smaller the robot is, the smaller the risks presented to the study participants and the researchers), nor does it require significant payload capabilities.

While a multitude of hobbyist and open-source robot platforms would fit these high-level needs, a commercial solution was ultimately required due to the necessity for replicability and repeatability of the testbed integration. Commercial availability of completed robots would reduce control and performance uncertainties stemming from consumer-built systems and components. Similarly, robotic systems constructed of higher quality parts (e.g., metal links rather than plastic, high-resolution harmonic drive motors rather than DC motors or hobbyist servos, and controls with more functionality and complexity enabling increased performance and capabilities) improve not only the expected performance and functional longevity of the robotic system, but also lend to the study participants’ trust in the capabilities of the system based on the perceived quality of the robot[[7]](#footnote-6).

From a functionality perspective, the following requirements of the robot and its controller were identified for the collaborative application:

* **Gravity compensation mode or minimal-torque, back-drivable joints for hand guiding applications.**  To minimize the negative user experience encumbered by requiring teach-pendant-based jogging, we desired to have a robot with some degree of hand-guiding capabilities so that study participants could directly and physically move the robot to a desired pose. The base assumption was that the responsiveness of the robot to the users’ applied forces and torques would be directly correlated to the perceived ease of use in terms of mental and physical effort required to get the robot to do what the user wanted it to do.
* **Digital I/O for sensor and external actuator integration.** Being able to send and receive digital signals is integral to the functionality of controlling third party grippers, either to directly control said grippers from the built-in programming interface, or to provide simple means of communications to a connected controller that is responsible for the actuation of the gripper.
* Well-documented tool flange to allow for swapping and/or customizing end-of-arm tooling
* Network connectivity for state reporting and/or control
* 5+ degrees of freedom
* Sub-millimeter pose repeatability
* Lift capacity > 1kg at full extension

Moreover, the following safety requirements were deemed necessary for the MASCAR testbed:

* Integrated software-based safety protocols such as collision detection (to identify when unexpected contact with the human, workpiece components, or the testbed itself) and soft axes to prevent overreach and motion within regions not relevant to the subject study application,
* Integrated dual-channel safety signals for emergency stops and external safeguards,
* Reduced velocity modes to minimize injury potential,
* Relatively low mass (< 15kg) to minimize injury potential,

### Ancillary Controller

A secondary programming and control interface is necessary for the evaluation of interface designs provides it presents a sufficiently distinct user experience. While NIST intends to develop prototype interfaces that represent significantly different control paradigms (e.g., virtual reality interfaces, and teach by demonstration).

Commercial secondary interface was purchased

ForgeOS selected

COTS support for selected robot

Easy integration

Significant limitation, however, is that there is no video out from the ForgeOS controller, that would allow us to record the actions of the user on the teach pendant

Logging into the underlying system did not provide usable screen data, and the Docker container that serves as the ROS interface for robot communication and control only allows for the use of a teach pendant or a virtual, on-screen teach pendant, not both.

Physical teach pendant might possibly be modified to allow for display output. LVDS signal from the teach pendant motherboard to the IPS LCD screen was documented and we tried to tap into that signal using a LVDS to HDMI adapter board, but we were unable to get this to work properly. All efforts were documented, along with the failure modes and ultimately abandoned. The teach pendant was then restored to near-stock settings.

Backup plan is to use a camera pointed at the teach pendant screen, and recording the user’s finger actions as they interact with the interface. Three different options being explored:

* ESP32-Cam module
* Raspberry Pi W with a camera module
* GoPro Hero 7 (hdmi output)

These trials are still ongoing as of the current version of this document.

### Gripper

The gripper is used to acquire and hold components for the assembly task

#### Gripper Selection

Either pneumatic parallel or servo-driven

Pneumatic 2-fingered gripper

* Reliable and easy to control
* Simple open/close action
* Custom fingers easily configured for application needs
* Can adjust gripping force by turning the air pressure up or down, but too little pressure might not overcome lateral stiction, thus no guarantee that the fingers will open or close properly
* Rapid actuation could damage parts
* Needs external air compressor (not horribly expensive, though) and hoses
* No built-in sensing, contact would have to be measured using pressure-sensitive pads

Servo-driven gripper options:

Many different options on the market

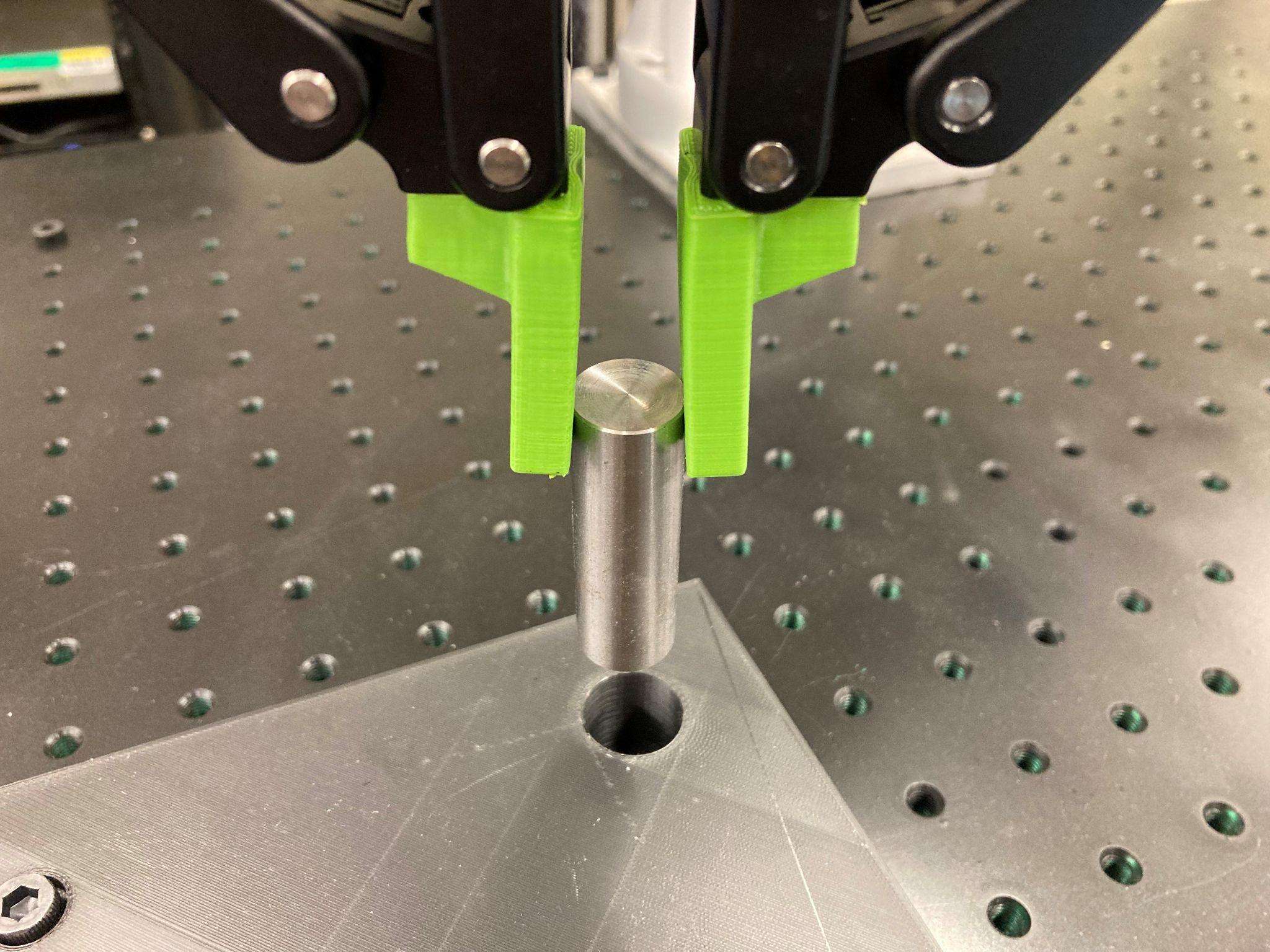
In this case, we defaulted to the same model being used by MTU, a Robotiq 2-fingered gripper

#### Gripper Fingertips

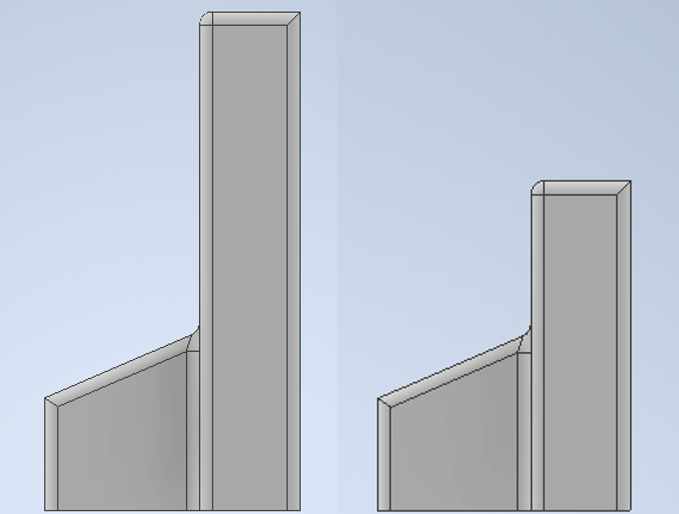
The Robotiq 2-fingered gripper ships with rubber-coated metal fingertips attached to the finger flanges. While robust, these fingertips do wear out over time, and can be expensive to replace. As the fingertips wear down, noticeable changes in the gripper’s performance may manifest, necessitating the replacement of the fingertips on a fairly regular basis depending on the wear. However, replacing the fingertips mid-study can negatively impact the results and traceability of the investigation if the replacements have not been verified as being one-to-one substitutes for the parts being replaced. Specifically, two different samples with different fingertips may not necessarily be directly comparable.

To alleviate this, we chose to omit the fingertips that came with the gripper, and use a custom-designed replacement that can be cheaply replicated and replaced, and quickly verified to be identical to one another at the time of manufacture. This fingertip design was largely based on the pre-existing fingertip design, except that the construction material was changed from metal to 3D printed plastic.

By 3D-printing the fingertips, we can replace worn fingertips frequently as needed with minimal cost. Custom-building the fingertips has tee added benefit of being highly customizable for the task, allowing us to vary the different variables such as fingertip length, material, and infill density (e.g., adjusting infill for safety purposes, where too much force would cause the fingertips to snap to reduce the severity of pinching hazards).



It is worth noting, however, that certain design variations will have performance consequences. For example, varying the infill density impacts the flexion along the moment arm of the gripper fingertip. A risk assessment determined the gripper force with the current fingertip design was insufficient to cause injury due to pinching. The fingertip design was therefore modified to shorten the tip component to increase stiffness and reduce uncertainty associated with the fingertip’s flexibility. Specifically, the fingertip length below the connection point was reduced from 37.5mm (similar to the OEM gripper finger) to 25mm, a ⅔ reduction in length.



### Gripper Control

Considerations:

Three possible solutions

UR controller

Direct control from the polyscope programming interface

Easy integration with Robotiq-provided URCaps

Robotiq sells packages specifically for UR integration

Cannot be controlled externally - has to be controlled programmatically through the Robotiq URCap

Robotiq Universal Controller

$1,700

Intended to be a common interface between Robotiq peripherals and external robot controllers and PLCs

When there are multiple communication protocols used in an industrial setting

Supported protocols include:

DeviceNet

CANopen

Profibus

Ethernet-based protocols

Ethernet/IP

Modbus TCP

EtherCAT

PROFINET

We previously used a Raspberry Pi for controlling a 3-fingered Robotiq gripper for a HRI tech demo. The 3-fingered gripper has a slightly different control interface than the 2 finger gripper (2-finger is easier to implement)

Raspberry Pi

Raspberry Pi 3B $35 for the base unit

Would need RS485 interface, which can be purchased for as little as $12 from some vendors. Average price is between $30 and $80.

Went with Pi option due to ease of integration and relatively low cost

Initial logic: press and hold a momentary button for 0.5s to toggle open/closed. Timing is easily adjusted in code

New challenge: allowing external controllers to signal to the Pi the need to open/close gripper

However, the robot controller would need to signal the Raspberry Pi somehow through the polyscope

Robot controller could send command to Pi using a digital output connected to a relay (when high, relay closes, pulling down the signal on the Pi’s GPIO, indicating an external signal to actuate the gripper)

Limitation of this is the increased number of steps to program the gripper actuation

Gripper activation

In polyscope:

URCaps

Gripper

Edit Action

Open/Close

Save Action

Open/close selection automatically causes the gripper to actuate. Five steps total.

In contrast with DIO action

Basic

Set

Set digital output

Select output

Set high/low

Test

Six steps total.

Custom URCap was written that could be used in the Polyscope programming environment to send commands to the Raspberry Pi

With new URCap interface:

URCaps

Gripper Actuation

Open/Close Gripper

Reduced to three steps

Keeping both hand- and robot-controlled actuation, there is a potential for signal conflict

### Hand-Guided Control Interface

Although the UR3e’s included teach pendant was designed such that it was relatively easy and comfortable to use, early experiments demonstrated that always having to hold onto the teach pendant led to strain and mild discomfort, especially when trying to manually move the robot around in freedrive mode. In some cases, two hands were required to move the robot to a specific pose in freedrive mode, which was physically impossible given the need to constantly hold down the freedrive enable button on the teach pendant.

To reduce this strain, we decided to add a handle with buttons that would enable local control of different functions of the robot. Specifically, we wanted to create a manual interface that would allow us to enable/disable freedrive, and actuate the gripper open or closed. Freedrive actuation on the UR3e’s controller, was relatively straightforward, since the controller natively allows digital inputs to be configured as external triggers for many activation-related functions, including Freedrive. Controlling the gripper remotely from the physical robot, however, would require a custom solution that must necessarily be compatible with the selected gripper control system (see Section 2.2.4).

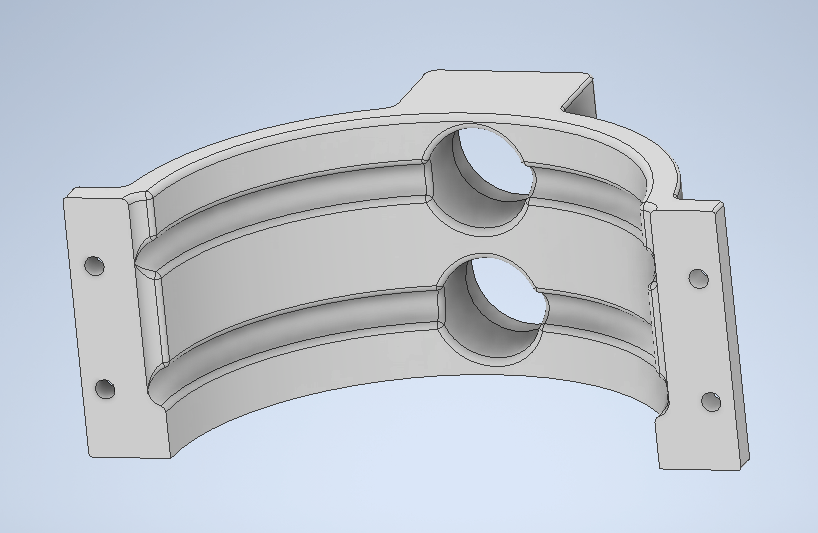
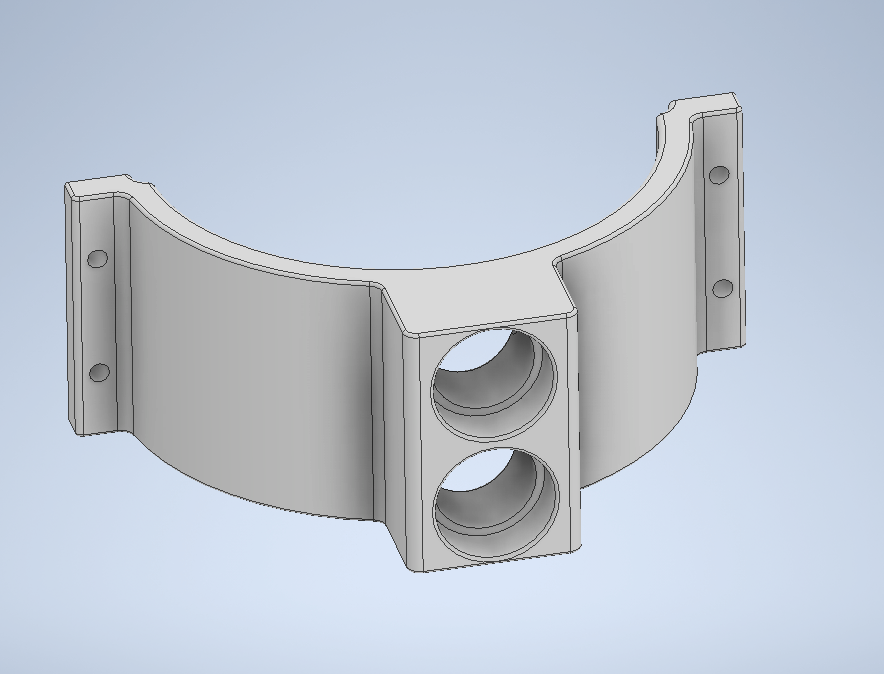
In this section, we discuss how the manual interface was designed, and how these designs changed as a result of pilot studies volunteer inputs.

#### Handle Version 1

The first version of the handle interface consisted of a two-part shackle design, allowing buttons to be situated on both sides of the gripper. Each part of the shackle consisted of identical halves of a cylindrical design, which helped reduce development time and complexity. Each shackle half would support two buttons, one for Freedrive activation, and one for gripper actuation. The inside curve had channels for wire management to and from the buttons to their respective controllers.

Early prototypes were 3D printed in ABS plastic and test-fitted onto the wrist of the Robotiq gripper. This design, however, proved to be impractical for three significant reasons:

1. The two-part shackle design was difficult to install by a single person, and required the use of temporary adhesives to hold parts in place. Moreover, even with the locking bolts in place and tightened completely, the shackle had a tendency to shift easily.
2. Even with the cable channels, the bolt positions at either end of the shackle were prone to pinching wires. And with the completed assembly also being allowed to shift, the wires were not guaranteed to stay in their respective channels.
3. The front- and back-mounted buttons were difficult to use, and early attestations claimed that extended use was uncomfortable and “unnatural.”



#### Handle Version 2

Addressed the issues regarding the shackle design.

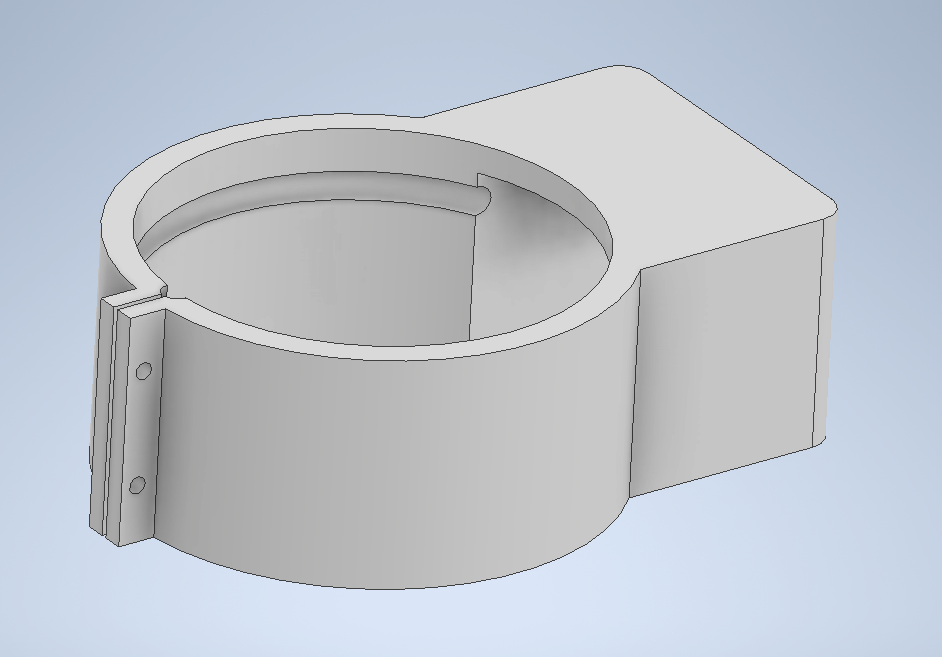
V2 tested single-ring design with a dummy button protrusion (not out of prototype stage)

Ring design without button flange to test fitting and be used for adjusting design/size parameters

#### Handle Version 3

V3.1

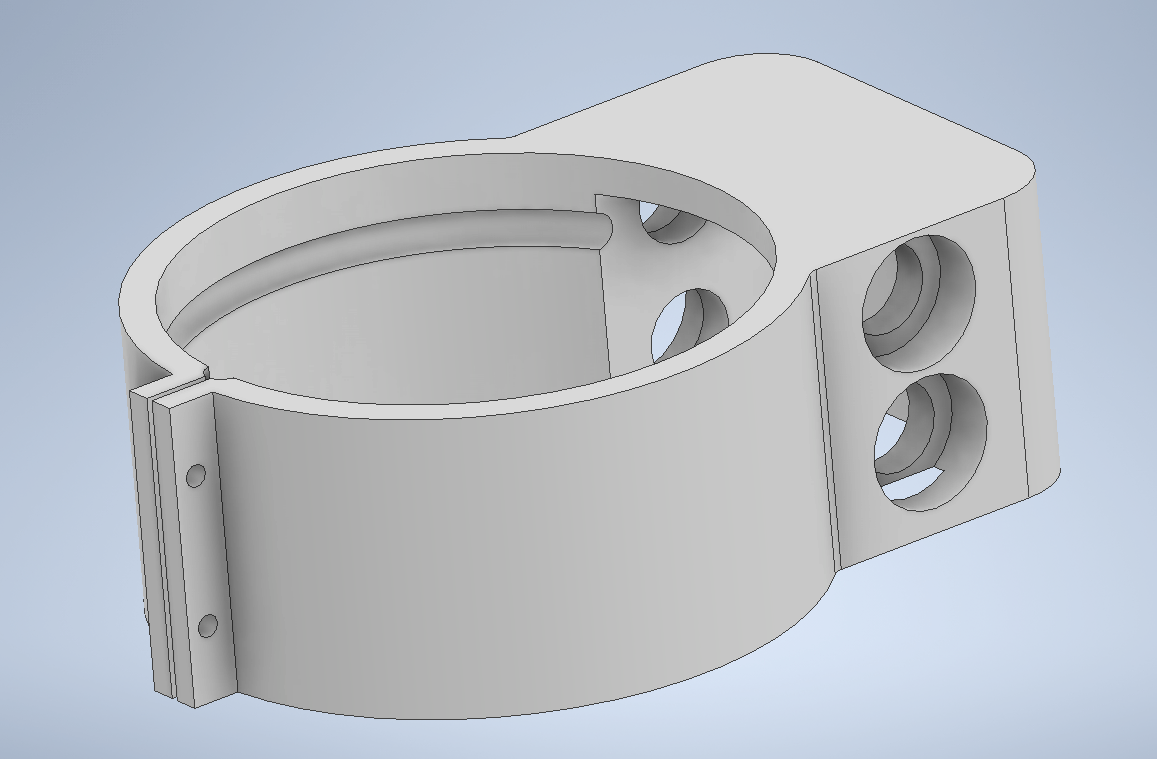
tested wiring management

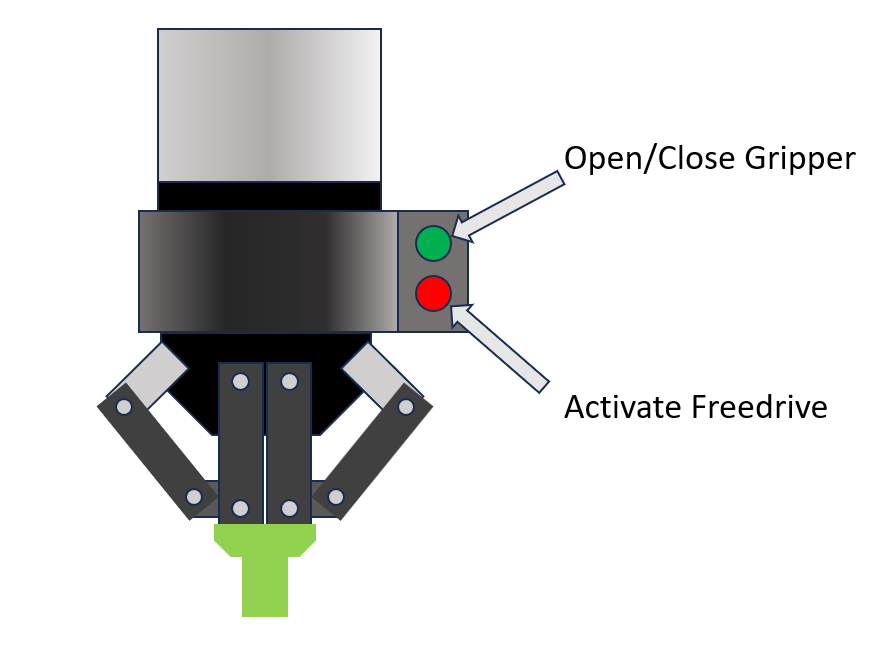


V3.2

Buttons added to both sides

First prototype to go into long-term use, and was used during NIST MOP & phase 0 trial of new wearable-based data collection trial





#### Handle Version 4

Extended handles for a yoke design

Separate primary and secondary functionality on both handles, integrated switches in addition to the momentary buttons

Only early aesthetic prototypes of V4 were ever built to test feasibility and comfort, and a functional prototype was never developed or installed. While these early prototypes were cumbersome and unwieldy, they provided useful insights into how the design could be improved. These improvements would be directly implemented on the next version of the handle interface.

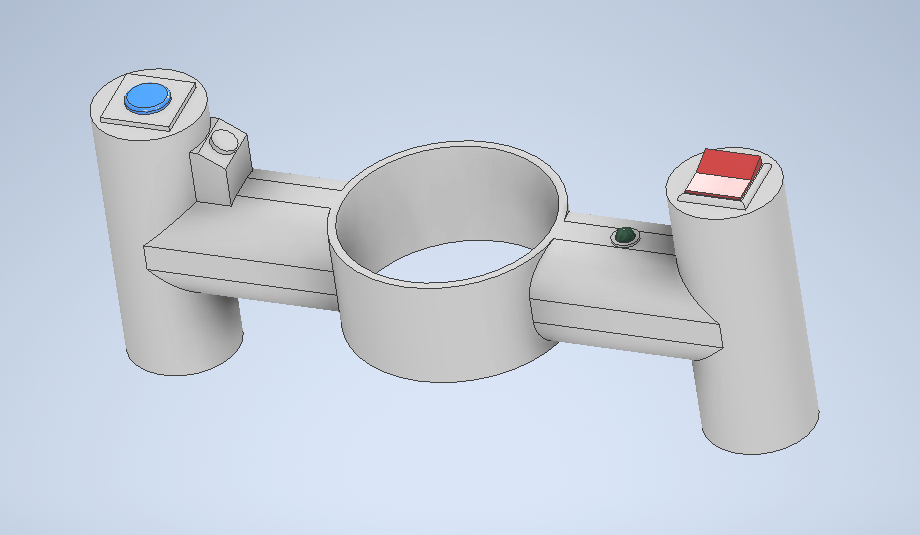
Designed to accommodate more buttons and functionality

thinner handles

much thinner cross-beam

placement and type of buttons (combination of latching and momentary switches)

needed modularity to allow for swapping out handles if they are too thick/thin,

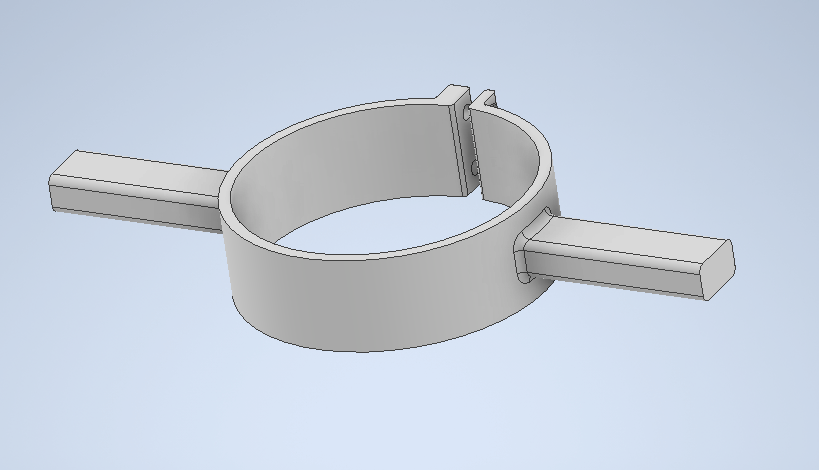


#### Handle Version 5

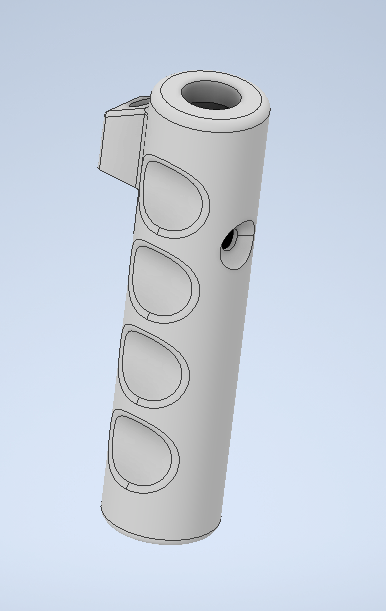
5.1

Streamlined design of central core for comfort

Designed to be modular, with the central core to be machined out of aluminum or acetal resin

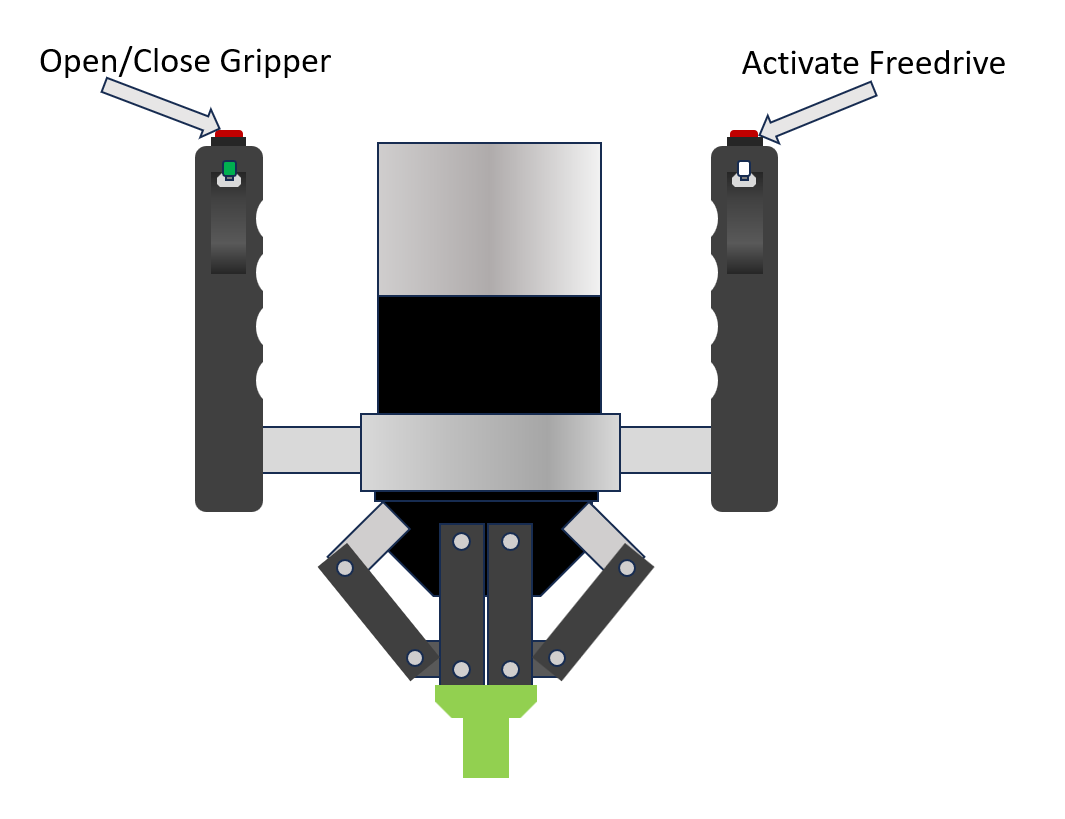


Handles intended to be replaceable with different button configurations as needed



V 5.2

Moved the connection point to be at the bottom of the handle such that the operator’s hands are above the connection point. This is to avoid obstructions with the base and pinch points with the gripper’s fingers.



As of the time of this document’s writing, only two of the four buttons are currently wired for functionality. For the left handle, the latching button is used to open and close the gripper. The latching functionality reduces the complexity of the actuation functionality, as the open/closed state of the gripper are directly related to the button state rather than tracking operational states and triggering actuation based on durations of button presses.

And, for the right handle, the latching button is used to enable Freedrive mode. Rather than forcing the user to hold down a button for the duration of the Freedrive motion, a simple toggle can be used to turn on Freedrive and keep it on as long as it is needed.

This mapping leaves both momentary buttons unassigned. However, it is expected that future iterations of the teach-by-demonstration assembly task will find practical use for momentary functionality (e.g., such as signaling the system to record robot positions, or editing stored points).

## Computing Devices

Precision 5860 Towers

* Upgrade (Processor): Intel® Xeon® W7-2495X (45 MB cache, 24C / 48T, 2.5 GHz to 4.8 GHz Turbo, 225 W)
* Upgrade (Graphics Card): Nvidia RTX 6000 Ada Generation, 48 GB GDDR6, 4 DP
* Upgrade (Memory): Upgrade Memory to 128GB, 4x32GB, 4800MT/s ECC RDIMM (CSP INSTALL)
* Upgrade (Hard Drive): Upgrade Internal Boot Drive - 2 TB, M.2, PCIe NVMe, SSD, Class 40
* Upgrade (2nd Hard Drive): Add 2nd Internal SSD - 2 TB, M.2, PCIe NVMe, SSD, Class 40
* Upgrade (PCIe Card): Thunderbolt 4 PCIe Card (2x Type-C, 2x DP-in)
* Upgrade (Network Cards): Intel® X710 10GbE NIC, Dual Port, Copper

Dedicated Raspberry Pi 5 for data logging

Raspberry Pi SC1112

Manufacturer: Raspberry Pi

Product Category: Single Board Computers

RoHS:

Series: Raspberry Pi 5

Form Factor: Raspberry Pi 5

Processor Brand: Broadcom

Processor Type: BCM2712

Core: ARM Cortex A76

Frequency: 2.4 GHz

Maximum RAM Capacity: 8 GB

Installed RAM: 8 GB

Operating Supply Voltage: 5 V

Interface Type: BLE, Ethernet, HDMI, Micro SD, MIPI, PCIe, USB 2.0, USB 3.0, USB Type-C, Video, WiFi

Dimensions: 85 mm x 56 mm

Packaging: Bulk

Brand: Raspberry Pi

Cache Memory: 2 MB

Product Type: Single Board Computers

180

Subcategory: Computing

Part # Aliases: SC1432

Raspberry Pi SC1635 7” touch display

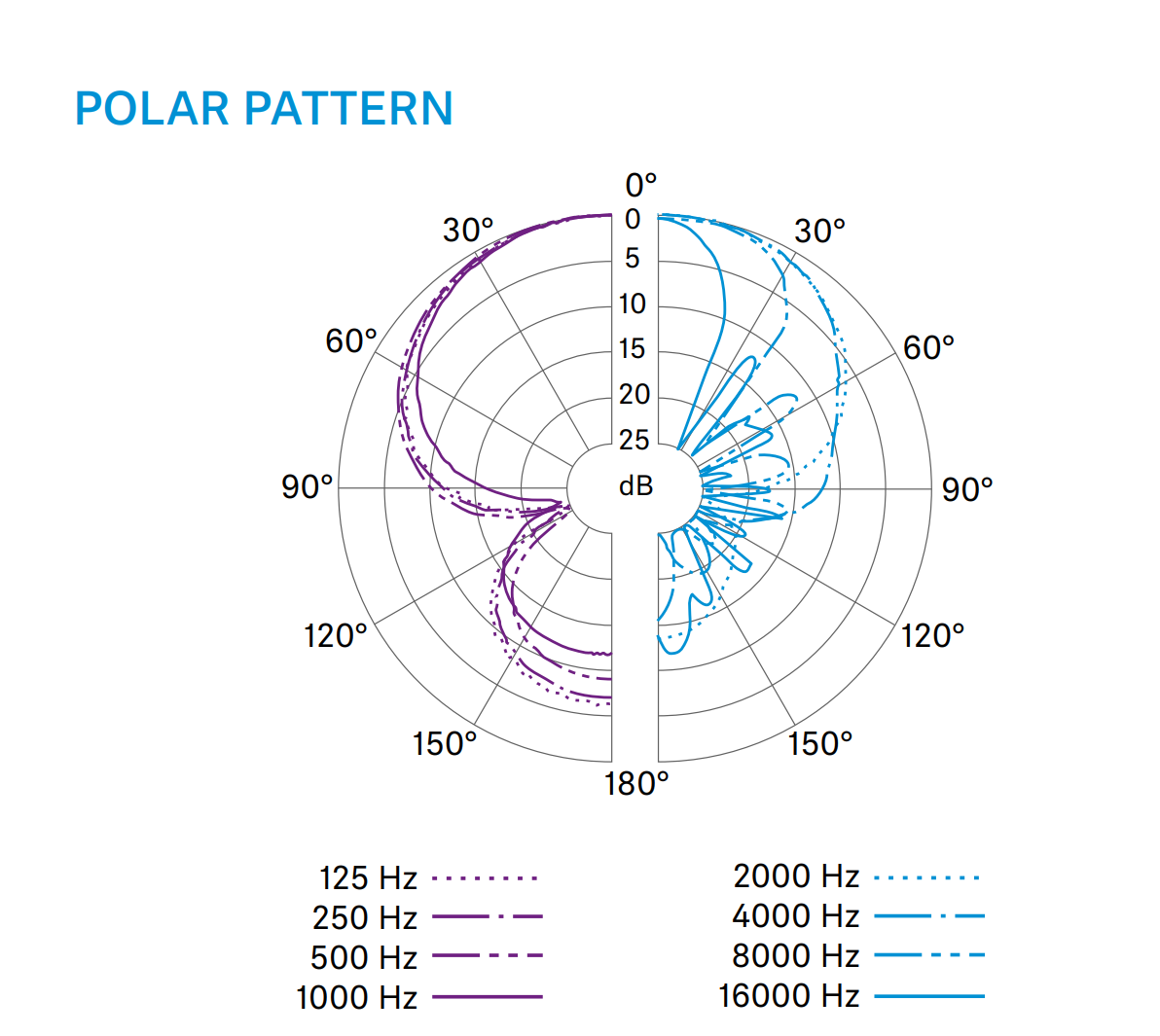
# Sensors

The principal function of the MASCAR testbed is to support the collection of high quality multimodal sensor data

## Audio

The testbed consists of two standalone Sennheiser MKE 600 Shotgun Microphones and two Senal MC24-ES Short Shotgun Microphones connected and positioned on top of two Canon XA15 camcorders.

### Postioning

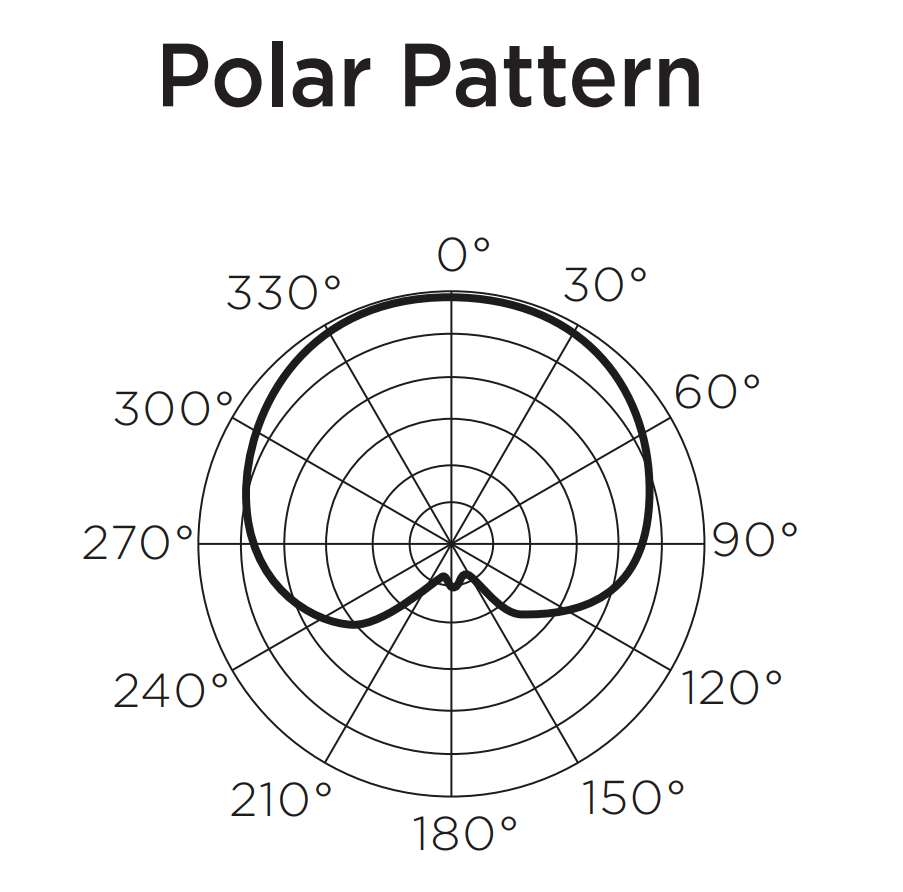
The polar pattern for the Sennheiser MKE 600 directional microphone performs best when recording directly in front of the target source (in this case, the subject or user) but will perform well within ±30° of the target source.  


Polar patterns of Sennheiser MKE 600 microphones [Product specification sheet MKE 600 - Sennheiser electronic GmbH & Co. KG · Am Labor 1 · 30900 Wedemark · Germany · [www.sennheiser.com](http://www.sennheiser.com)]

For our application the MKE 600 microphones will be positioned overhead of the subject on either side not to exceed ±30°, just out of frame of the front facing camcorder with the approximate distance from the subject standing normally to be determined following testing with the audio interface. However, distance from the subject should be large enough to ensure no collision with equipment for safety purposes.

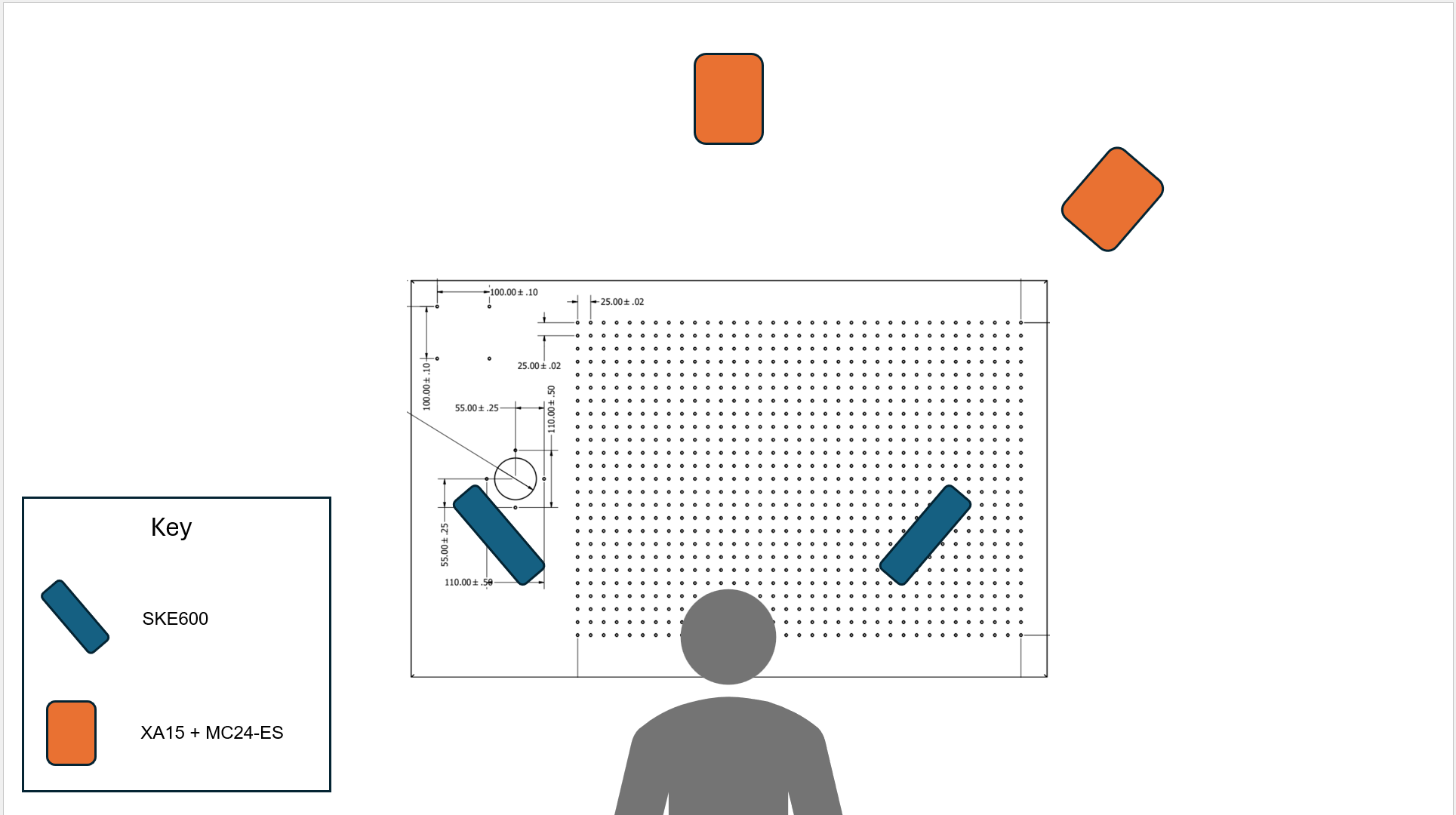
As the camcorders will be positioned per the needs of the desired camera angle and field of view - and the MC24-ES microphones are mounted on top of said camcorders, the distance between the subject and the sensor is expected to result in a small degradation in sample quality. These sensors are intended to replicate a combination camera and microphone setup and thus the optimal positioning aside from that for the camera itself is not of particular concern.

The polar pattern for the MC24-ES Cameras is optimal at 0° and preferred to be within ±30° of the target source.



Polar patterns of the Senal MC24-ES Short Shotgun Microphones [User Instruction Manual <https://www.senalsound.com/product/8774/Senal-MC24_ES-Professional-Condenser-Short-Shotgun-Microphone> ]

The diagram below shows the approximate planned positioning of the microphones from an overhead perspective.



Overhead diagram of approximate positioning (subject to change).

All four microphones require phantom power and a pre-amplifier and will connect to the f8n pro recorder by XLR cables through their respective pre-amplifiers.

## Vision

While the testbed also collects vision information in the form of motion capture systems described in the following sections, the testbed also collects high resolution Video and depth information from multiple sources and angles. The testbed records video from two Canon XA-15 camcorders recording at Full HD (1080p), and two Intel Realsense 455 RGB-D sensors.

Each camcorder is equipped with a Senal MC24-ES shotgun microphone recording separately into the F8N field recorder as described in the previous section.   
  
There is one camcorder that is pointed directly at the user from the front, and one camcorder pointed at the user from an angle. There is a intel realsense positioned below the front facing camcorder, and another intel realsense positioned overhead of the testbed positioned on a c-stand as these are the two most relevant points for collecting depth data external to the robot, as one captures information regarding the human and the other captures information about the task.

## Object/Sensor Tracking

Two motion capture systems

### Rigid body motion capture

Leveraging two motion capture systems for rigid bodies, but currently only planning to use for extrinsic calibration of sensors within the MASCAR testbed environment

System 1: OptiTrack trio

measuring extrinsic configuration of sensors relative to the testbed table

System 2: Vicon Bonita 10

An older motion capture system already being used in the HRI laboratory

measuring 6DoF pose of objects within testbed

8x 1.0 MP PoE IR cameras

## Human motion capture

The human motion capture is

Wanted a sensor system that could track people within the testbed, and not require participants to wear markers, fiducials, or active sensors to encourage more natural interaction with the robot

Captury

Dedicated PC

10Gbps network

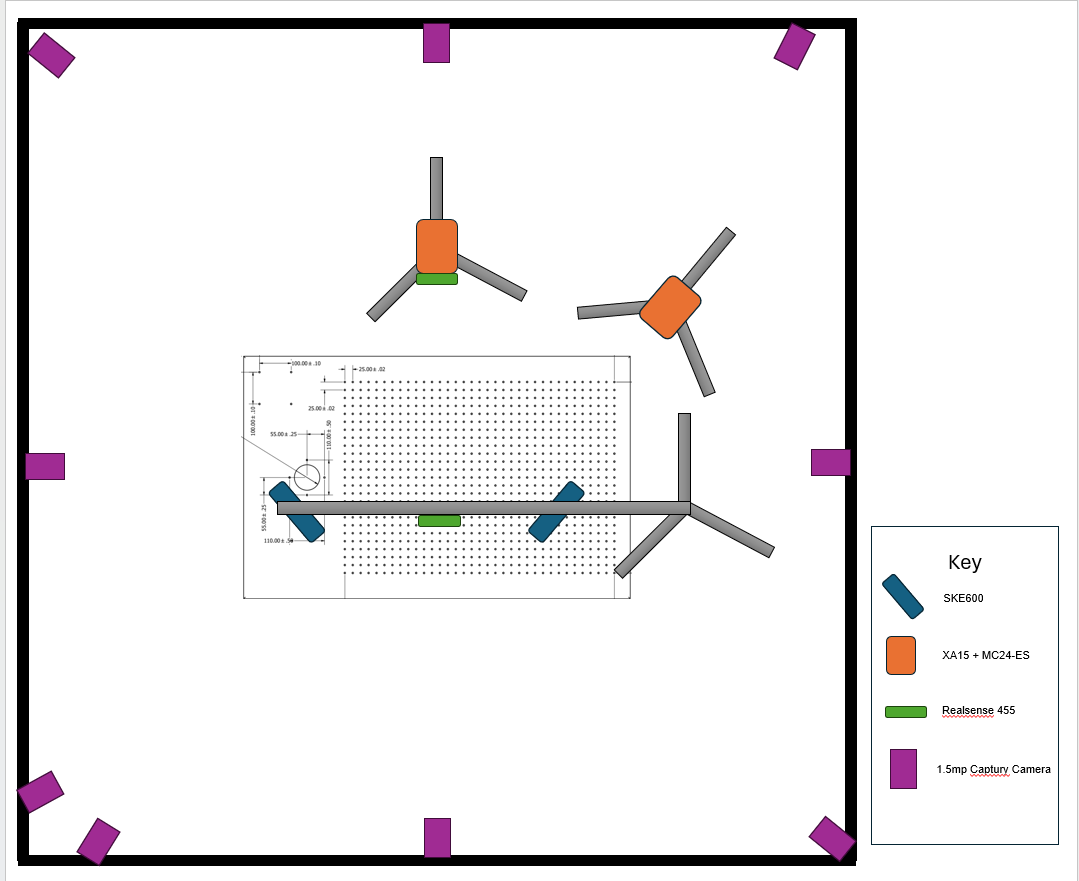
11 1.3MP PoE color cameras

## Sensor Mounting

Static, overhead sensors are on ceiling and wall mounted overhead racks that span the lab space surrounding the testbed. These racks consist of cable trays that are located about 10 inches from the ceiling, and are aligned in a square surrounding the testbed with an additional cable tray serving as a horizontal bridge in the center, parallel to the testbed. These racks mount 9 ethernet cameras for the captury motion capture system with different angles provided by arms holding said cameras extending downward from the racks in different positions. In addition to the ethernet cameras, the laboratory’s VICON motion capture system is mounted overhead on the racks as well, pointed in the general direction of the testbed. These VICON sensors are able to be used if needed for additional calibration information, but there are no plans to implement these sensors in the near future, as calibration information is provided by the OPTITRACK trio system separately.  
  
Additional sensor mounting has gone through several iterations. The initial design had sensors directly mounted to the testbed table. Though the testbed is sufficiently static and stabilized for the safe and repeatable operation of the robot , this method proved to be inadequate for the needs of the project as the robot motions on the table and mobile nature of the testbed table resulted in positional noise and movement of cameras and sensors.

The sensors were moved to tripods with optical motion capture points mounted to each sensor and the testbed table to characterize the position relative to the testbed table. While this solved the stability problem, the more mobile nature and size of the tripods caused new concerns for positioning, stability, and floorspace constraints.

The current setup has solved these issues by migrating from tripods to mounting sensors on C-Stands, which are stabilized with sandbag weights. The floorprint is similar to a tripod, but this method frees up workable horizontal space previously occupied by tripod legs while also being more stable with the option to add additional weights as needed. The height and extension provided by c-stands with additional overhead arms also enabled the use of the overhead realsense camera with a much shorter USB-C cable than would be needed to route from the ceiling mounts to the linux box.   
  
There are four C-stands used in this current configuration, one for the front facing camcorder and realsense 455. One for the side facing camcorder, one for the overhead boom microphones and the overhead realsense 455, and one for the optitrack trio (not shown in current graphic).



The above diagram describes the positions of all visual and audio sensors in relation to the testbed.

# Communications and Control

Network backplane for inter-process communications a dedicated 1Gbps network switch set up as a LAN to reduce transmission distance/time and minimize network noise

1Gbps network LAN

Robot

ForgeOS

Gripper Control and State Reporting Raspberry Pi

Integrated Logger Raspberry Pi

External Computer(s) for recording

Backup Hard Drives

NTP vs PTP

Timecode generators

# Data Collection and Storage

## Data feed collection and synchronization

### Data Collation

Each data stream will have an associated timecode generator audio file, originating from separate Deity TC-1 devices synchronized via the Deity Sidus Audio[[8]](#footnote-7) app. The resulting audio files for all feeds are then analyzed to determine the synchronization time for each feed. Then all feeds are added to the Lab Streaming Layer ecosystem which uses the sample rate per feed, and the synchronization times to determine the time offsets and then aligns the feeds temporally.

#### Data Collection

This effort involves the collection of a variety of different formats of data with varying sample rates.

#### Robot Data

Robot data is collected and parsed in Python using the Universal Robots real-time data exchange (RTDE) python library[[9]](#footnote-8), and saved in CSV format with 119 data points per entry

Documentation regarding data points collected is provided by UR in their wiki. [Real-Time Data Exchange (RTDE) Guide — PolyScope Tutorials documentation](https://docs.universal-robots.com/tutorials/communication-protocol-tutorials/rtde-guide.html)

timestamp,, target\_q\_0, target\_q\_1, target\_q\_2, target\_q\_3, target\_q\_4, target\_q\_5, target\_qd\_0, target\_qd\_1, target\_qd\_2, target\_qd\_3, target\_qd\_4, target\_qd\_5, target\_qdd\_0, target\_qdd\_1, target\_qdd\_2, target\_qdd\_3, target\_qdd\_4, target\_qdd\_5, target\_current\_0, target\_current\_1, target\_current\_2, target\_current\_3, target\_current\_4, target\_current\_5, target\_moment\_0, target\_moment\_1, target\_moment\_2, target\_moment\_3, target\_moment\_4, target\_moment\_5, actual\_q\_0, actual\_q\_1, actual\_q\_2, actual\_q\_3, actual\_q\_4, actual\_q\_5, actual\_qd\_0, actual\_qd\_1, actual\_qd\_2, actual\_qd\_3, actual\_qd\_4, actual\_qd\_5, actual\_current\_0, actual\_current\_1, actual\_current\_2, actual\_current\_3, actual\_current\_4, actual\_current\_5, joint\_control\_output\_0, joint\_control\_output\_1, joint\_control\_output\_2, joint\_control\_output\_3, joint\_control\_output\_4, joint\_control\_output\_5, actual\_TCP\_pose\_0, actual\_TCP\_pose\_1, actual\_TCP\_pose\_2, actual\_TCP\_pose\_3, actual\_TCP\_pose\_4, actual\_TCP\_pose\_5, actual\_TCP\_speed\_0, actual\_TCP\_speed\_1, actual\_TCP\_speed\_2, actual\_TCP\_speed\_3, actual\_TCP\_speed\_4, actual\_TCP\_speed\_5, actual\_TCP\_force\_0, actual\_TCP\_force\_1, actual\_TCP\_force\_2, actual\_TCP\_force\_3, actual\_TCP\_force\_4, actual\_TCP\_force\_5, target\_TCP\_pose\_0, target\_TCP\_pose\_1, target\_TCP\_pose\_2, target\_TCP\_pose\_3, target\_TCP\_pose\_4, target\_TCP\_pose\_5, target\_TCP\_speed\_0, target\_TCP\_speed\_1, target\_TCP\_speed\_2, target\_TCP\_speed\_3, target\_TCP\_speed\_4, target\_TCP\_speed\_5, actual\_digital\_input\_bits, joint\_temperatures\_0, joint\_temperatures\_1, joint\_temperatures\_2, joint\_temperatures\_3, joint\_temperatures\_4, joint\_temperatures\_5, actual\_execution\_time, robot\_mode, joint\_mode\_0, joint\_mode\_1, joint\_mode\_2, joint\_mode\_3, joint\_mode\_4, joint\_mode\_5, safety\_status, actual\_tool\_accelerometer\_0, actual\_tool\_accelerometer\_1, actual\_tool\_accelerometer\_2, speed\_scaling, target\_speed\_fraction, actual\_momentum, actual\_main\_voltage, actual\_robot\_voltage, actual\_robot\_current, actual\_joint\_voltage\_0, actual\_joint\_voltage\_1, actual\_joint\_voltage\_2, actual\_joint\_voltage\_3, actual\_joint\_voltage\_4, actual\_joint\_voltage\_5, actual\_digital\_output\_bits, runtime\_state, joint\_position\_deviation\_ratio



/\*

| Element | ID | Format | Units | Note |
| --- | --- | --- | --- | --- |
| 1 | timestamp | Double | sec | Time elapsed since the controller was started |
| 2-7 | target\_q | Vector of 6 doubles | rad | Target joint positions |
| 8-13 | target\_qd | Vector of 6 doubles | rad/sec | Target joint velocities |
| 14-19 | target\_qdd | Vector of 6 doubles | rad/sec/sec | Target joint accelerations |
| 20-25 | target\_current | Vector of 6 doubles | amp | Target joint currents |
| 26-31 | Target\_moment | Vector of 6 doubles |  | Target joint torques |
| 32-37 | actual\_q | Vector of 6 doubles | rad | Actual joint positions |
| 38-43 | actual\_qd | Vector of 6 doubles | rad/sec | Actual joint velocities |
| 44-49 | actual\_current | Vector of 6 doubles | amp | Actual joint currents |
| 50-55 | joint\_control\_output | Vector of 6 doubles |  | Joint control currents |
| 56-61 | actual\_TCP\_pose | Vector of 6 doubles |  | Actual Cartesian coordinates of the tool (x, y, z, rx, ry, rz), where rx, ry, and rz represent the axis-angle vector, and the magnitude of (rx, ry, rz) is the rotation around that vector |
| 62-67 | actual\_TCP\_speed | Vector of 6 doubles | m/s  rad/s | Actual velocity of the tool in the Cartesian space |
| 68-73 | actual \_TCP\_force | Vector of 6 doubles |  | Generalize forces at the TCP, compensating for forces and torques imposed by the defined payload |
| 74-79 | target\_TCP\_pose | Vector of 6 doubles |  | TargetCartesian coordinates of the tool (x, y, z, rx, ry, rz), where rx, ry, and rz represent the axis-angle vector, and the magnitude of (rx, ry, rz) is the rotation around that vector |
| 80-85 | target\_TCP\_speed | Vector of 6 doubles |  | Target velocity of the tool in the Cartesian space |
| 86 | actual\_digital\_input\_bits | 64-bit unsigned integer | binary | Current state of the digital inputs (0-7: Standard, 8-15: Configurable, 16-17: Tool) |
| 87-92 | joint\_temperatures | Vector of 6 doubles | Celcius | Temperature of each joint |
| 93 | actual\_execution\_time | Double | ms | Controller real-time thread execution time |
| 94 | robot\_mode | 32-bit integer |  | Robot mode |
| 95-100 | joint\_mode | Vector of 6 32-bit integers |  | Joints control mode |
| 101 | safety\_status | 32-bit integer |  | Safety status |
| 102-104 | actual\_tool\_accelerometer | Vector of 3 doubles |  | Tool x, y, and z accelerometer values |
| 105 | speed\_scaling | Double |  | Speed scaling of the trajectory limiter |
| 106 | target\_speed\_fraction | Double |  | Target speed fraction |
| 107 | actual\_momentum | Double |  | Norm of cartesian linear momentum |
| 108 | actual\_main\_voltage | Double |  | Safety Control Board: Main voltage |
| 109 | actual\_robot\_voltage | Double |  | Safety Control Board: Robot voltage (48V) |
| 110 | actual\_robot\_current | Double |  | Safety Control Board: Robot current |
| 111-116 | actual\_joint\_voltage | Vector of 6 doubles | Volts | Actual joint voltages |
| 117 | actual\_digital\_output\_bits | 64-bit unsigned integer |  | Current state of the digital outputs (0-7: Standard, 8-15: Configurable, 16-17: Tool) |
| 118 | runtime\_state | Unsi |  | Program state |
| 119 | joint\_position\_deviation\_ratio | Double |  | A C153 or C159 protective stop is triggered if the value is ever equal to or exceeds 1.0 |

A Raspberry Pi 5 is attached to the MASCAR LAN, and connects to the UR3e controller via TCP/IP

#### Gripper State

Same Raspberry Pi 5 that is used to collect the robot data is also used to stream-record the state data from the gripper Pi 4B via TCP/IP

Gripper data is collected with the following script:

strData\_ = '''{{"GripperData": {{

"act": "{}",

"calib": "{}",

"pbits": "{}",

"pmm": "{}",

"button": "{}",

"grip": "{}",

"actTime": "{}"}}

}}'''.format(isActive, isCalib, pos\_bits, pos\_mm, button\_state, grip\_state, button\_time)



Gripper state data

{"GripperData": {

"act": "True",

"calib": "False",

"pbits": "230",

"pmm": "-1.0",

"button": "False",

"grip": "False",

"actTime": "1812410.666974608"}

}



#### Video

##### Black Magic

To simplify the synchronization of visual feeds, standard video feeds with HDMI output capabilities are recorded in tandem with a single time code generator using the ATEM Mini Extreme HDMI Live Stream Switcher onto a portable SSD drive. The switcher outputs a single synchronized Davinci Resolve File which contains all video outputs on separate tracks. This video is then rendered separately and exported as the files for the respective feeds and aligned with the respective audio recorded as outlined in sections 5.1.2.4 and 3.1.

The following feeds will be synchronized through the Blackmagic ATEM Mini Extreme HDMI Live Stream Switcher.

* Front facing camcorder (Canon XA15)
* Side view camcorder (Canon XA15)
* UR3e Teach Pendant (Via Stream or wired connection)
* Alternate view points or angles including additional screen views if necessary (GoPro Hero7)

A timecode generator is attached to the switcher so an audio track is actively recording for synchronization purposes.

##### RGB-D

RGB-D Data is recorded to a Linux machine through two intel realsense 455 sensors into a ROSBAG file containing both the sensor data and a timecode generator audio file. One sensor is positioned directly overhead of the workstation on a C-Stand that also holds the overhead booms, and the other is positioned in front of the testbed beneath the front facing Canon XA15 camcorder.  
The sensors are each connected to the ROS enabled Linux machine via USB-C cables and can be hardware synchronized to capture at the same rate and clock time using a multicamera synchronization method described by intel. ([Using the Intel® RealSense™™ Depth cameras D4xx in Multi-Camera Configurations](https://www.intel.com/content/www/us/en/content-details/841998/using-the-intel-realsense-depth-cameras-d4xx-in-multi-camera-configurations.html)). A ROS node will run using the intel realsense API through the Realsense ROS wrapper. ([GitHub - IntelRealSense/realsense-ros: ROS Wrapper for Intel(R) RealSense(TM) Cameras](https://github.com/IntelRealSense/realsense-ros)).  
Analysis is being run on the practicality and benefits of using hardware synchronization, or if it is sufficient to provide synchronization using the Linux machine’s timecode generator as input to the LabStreamingLayer node.

##### External Display Feeds

UR3e Teach pendant

Streaming the teach pendant screen via virtual network computing (VNC), and stored on a separate computer and simultaneously recorded through the blackmagic switcher for synchronization purposes.

ForgeOS feed

Future research efforts that leverage the ForgeOS interface for controlling and programming the robot will require a video feed that captures how the volunteers are interacting with the ForgeOS teach pendant.

By default, there is no way to stream the teach pendant screen to an external display. We have made efforts to modify the ForgeOS hardware and software to gain access to the display signals, but so far all attempts to stream this information from the stock hardware have failed. These efforts have been documented, along with the particular failure modes encountered.

We are currently investigating the feasibility of mounting an external camera to the ForgeOS teach pendant, but do not anticipate having a workable video stream until after initial human trials have been collected in FY 26.

#### Audio

The dataset will include Four environmental audio tracks. This audio will be recorded locally on the Zoom F8N field recorder with an additional timecode generator track as described in section 3.1 and transferred to the centralized file folders manually prior to being synchronized using the LabStreaming Layer environment. Audio files will be analyzed intermittently with Audacity and other DAWs.

The F8N will record two multitrack audio files. Each .wav file will have two tracks corresponding to an individual microphone. The files will be separated based on the type of microphone being used, with one file for the two SKE600 microphones and one file for the MC24-ES microphones positioned on top of the camcorders. Each file will be recorded on a separate SD card as follows:

File 1 - SKE600 Microphones

Track 1 - SKE600 A

Track 2 - SKE600 B

File 2 - MC24-ES Microphones

Track 1 - MC24-ES A

Track 2 - MC24-ES B

The exact positioning and designation of the individual microphones as A and B will be noted at the time of setup and documented.

## Storage

During the development of the dataset, the data will be stored on the local HRI lab servers. The team will consult with the HSPO regarding the protection of subject data during the collection process. All team members with access to the data have completed Human Subjects Training and will be included in the IRB for the data collection protocol.

The HRI lab servers consist of two physically lockable multi-bay network storage stations with 32 terabytes of storage space, one Synology DiskStation DS3612xs (8-bay network storage) with 8x 4 TB hard drives, Synology DiskStation DS418 (4-bay network storage) with 4x 8TB hard drives.

To ensure privacy and proper data handling, key access and network access during and following the collection period will be limited to federal staff named on the planned IRB protocol directly working on the project, with the proper Human Subjects Training.

## File Structure

Due to the restrictions of the motion capture software on longer recording times, The proposed file structure for entries in the database is dictated by the subject number, interaction number or task, and recording segment. This design is subject to change.  
The following json file format will be used to describe

{SubjectID: "X1",

Scenario: 1,

Files: [

optitrak: CalibrationFileName

scenarioFeeds:{

mocap: "capturyFileName",

topRealsense: "Top.bag",

FrontRealsense: "Front.bag"

},

continousFeeds: {

scenarioStart: TIMECODE\_OFFSET,

frontVideo: "recording1\_DD\_TT\_YY.mov",

sidevideo: "recording2\_DD\_TT\_YY.mov",

UR3Pendant: "U\_pendant\_DD\_TT\_YY.mov",

ForgePendant: "F\_pendant\_DD\_TT\_YY.mov",

frontAudio: "fntAudio\_DD\_TT\_YY.mp3",

sideAudio: "sideAudio\_DD\_TT\_YY.mp3",

boomAudio: "bmAudio\_DD\_TT\_YY.mp3",

robotData: "robotData.json",

gripperData: "gripperData.json"

}

]

}



## Tagging

### Labstreaming player

Lab Streaming Layer (abbreviated as LSL) is a system for collating many streams of data into a synchronized and centralized collection. LSL is used in this testbed as a way to combine all of the datastreams we have access to into one location of the semantic event tagging of the time series.  
  
LSL will be used to tag events like “Subject picks up part with robot hand”, “Subject toggles freedrive button”,“robot moves towards goal”,“robot reaches goal point”, or “Gripper closed” among many others in the series across all streams. Utterances and subtitles will be described as well.

Guidelines on tagging will be developed to enable consistent tagging across scenarios and tasks and provided in future documentation. Earlier research efforts when compiling and collating metadata demonstrated the necessity of establishing guidelines for categorization, particularly when multiple people/organizations are intended to contribute to the classification.  
  
Additional tagging scope will be defined depending on funding and staffing capabilities. Should funding increase and more staff be allocated to this effort, more tagging capabilities will be possible, including bounding boxes, 3d cuboid tagging, and quality controlled automated tagging.

## Future work

HSPO information about sharing datasets that contain human subject information

Dissemination plan of data

# Task Components

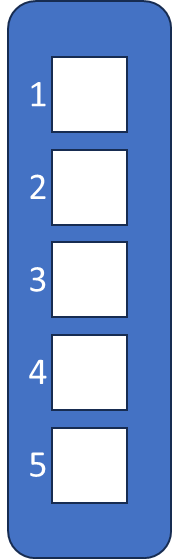
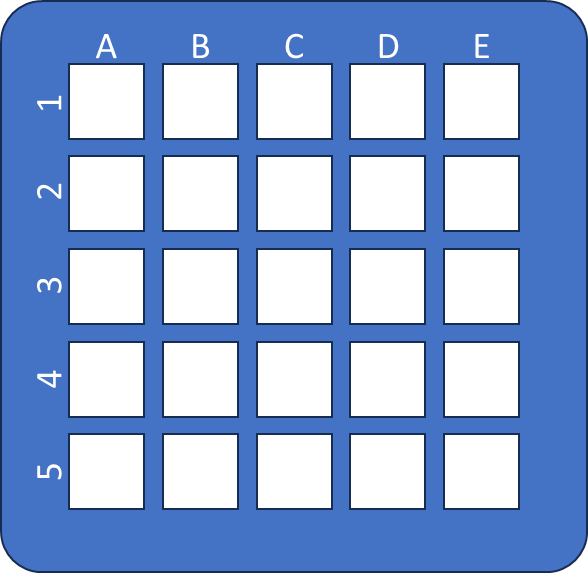
Assembly task, programming the robot to complete the assembly

## Preliminary Assembly Task (Block Grid)

(TEXT FROM THE MOP SOP. REPLACE WITH TEXT CUSTOM FOR THIS REPORT)

During this study, your task will be to move cubes from a staging area (left column) to specific locations on the target board (right grid). You will be instructed which block to move from the staging area using its numerical position in the column, and where on the grid to move the block using its column and row combination. Each block will have a colored indentation on the top, but this is not relevant to this test.

For example, you may be asked to move block #2 to position E1 on the grid, in which case you will use the robot to pick up the block from the staging area in space 2 (second from the top), and then move it to location E1 (top right square) on the grid. While each block will have a color associated with it, these colors are *not* part of this test.

The order of the pick-and-place operations does matter, so please try to do each action in the order it is given to you.

Example assembly scenario

|  | **Block** | | | | |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** |
| Target Location | **C1** | **A3** | **E4** | **B2** | **D5** |

# 

## Assembly Task Board

Leveraged NIST Task Boards

Modified specifically for the human study

Leveraged NIST task boards

ATB 1

Subset of NTB options:

Round peg-in-hole

Square peg-in-hole

We will need IDs for parts for tagging

# Digital and Physical Twins

All physical assets had CAD models either provided or custom-constructed to enable twinning, both physically and digitally

Digital twins for offline programming, testing, etc.

Majority of mechanical components have CAD counterparts such that we can replace

1. Certain commercial equipment, instruments, or materials are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose. [↑](#footnote-ref-0)
2. <https://www.nist.gov/el/intelligent-systems-division-73500/robotic-grasping-and-manipulation-assembly/assembly> [↑](#footnote-ref-1)
3. E.g., Marvel, Jeremy A., and Karl Van Wyk. "Simplified framework for robot coordinate registration for manufacturing applications." *2016 IEEE International Symposium on Assembly and Manufacturing (ISAM)*. IEEE, 2016. [↑](#footnote-ref-2)
4. Marvel, Jeremy A., Tsai-Hong Hong, and Elena Messina. "2011 solutions in perception challenge performance metrics and results." *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*. 2012. [↑](#footnote-ref-3)
5. <https://www.robot-manipulation.org/nist-moad> [↑](#footnote-ref-4)
6. <https://hriscaledatabase.psychology.gmu.edu/> [↑](#footnote-ref-5)
7. Ashby, Mike, and Kara Johnson. "The art of materials selection." *Materials today* 6.12 (2003): 24-35.; Karana, E. L. V. I. N., and I. L. S. E. Van Kesteren. "Materials affect: The role of materials in product experience." *Design and emotion moves* (2008): 221-245. [↑](#footnote-ref-6)
8. <https://deitymic.com/products/sidus-audio/> [↑](#footnote-ref-7)
9. <https://github.com/UniversalRobots/RTDE_Python_Client_Library> [↑](#footnote-ref-8)