Robotics and Computer–Integrated Manufacturing 70 (2021) 102113

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcm

Assessing Industrial Robot agility through international competitions

Anthony Downs [c,*], Zeid Kootbally [a,b], William Harrison [a], Pavel Pilliptchak [a], Brian Antonishek [a], Murat Aksu [a], Craig Schlenoff [a], Satyandra K. Gupta [b]

[a] National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD, United States
[b] University of Southern California, Los Angeles, CA, United States
[c] 100 Bureau Drive, Mail Stop 8230, Gaithersburg, MD, United States

ARTICLE INFO

Keywords:
Performance evaluation
Industrial Robotics
Artificial intelligence
Agility
Competitions

ABSTRACT

Manufacturing and Industrial Robotics have reached a point where to be more useful to small and medium sized manufacturers, the systems must become more agile and must be able to adapt to changes in the environment. This paper describes the process for creating and the lessons learned over multiple years of the Agile Robotics for Industrial Automation Competition (ARIAC) being run by the National Institute of Standards and Technology.

1. Introduction

Modern manufacturing is under ever increasing pressure to develop solutions for highly complex tasks. The increased number of new models and variants have forced manufacturing firms to shift away from high-volume/low-mix production to low-volume/high-mix production in order to meet the demands of a diversified customer base. Customer satisfaction is crucial in the current economy and requires production as per customer needs at cheaper rates, with reliability, and high quality.

Small and medium manufacturers (SMMs), defined by the National Association of Manufacturers (NAM)[1] as companies with 2500 or fewer employees, represent a very important segment of the manufacturing sector. SMMs make up 99% of all firms, employ over 50% of private sector employees, generate 65% of net new private sector jobs, and contributed $2.18 trillion to the economy in 2016. As we move towards shorter product life cycles and customized products, the future of manufacturing in the U.S. will depend upon the SMMs' ability to remain cost-competitive. Current manufacturing processes in SMMs rely on a significant amount of manual operations. This is the case of tool and die makers which are operated by human workers. Although these companies are capable of offering custom products in a short time (less than a month), the associated manual operations have high running costs. The pressures in today's economic climate leave many SMMs struggling to find ways to contain manufacturing costs. In recent years, many SMMs have turned to automation in order to compete with low-cost sourcing. Coupled with the significant cost reduction, many companies are able to justify return on investment in one to two years. Among the available automation strategies, hard automation usually represents the lowest first-cost option. Hard automation is a robot or machine that is designed to perform a specific highly repetitive task. The task is usually a simple operation or a combination of simple operations. For instance, some automotive parts (e.g., oil pans [1]) stay the same for years before they are redesigned. In such a scenario, hard automation is preferred where the priorities lie in long-term repeatability and quality when retooling and constant redesign are not necessary.

While hard automation has advantages that include low unit cost, automated material handling, and a high production rate, one of the main disadvantages of such a system is its inability to accommodate product changes in order to meet the demands of a diversified customer base. Customer satisfaction is crucial in the current economy and requires production as per customer needs at cheaper rates, with reliability, and high quality. For SMMs to be able to satisfy consumer demand for products with shorter life cycles and a greater variety of products or variants of existing products, they need to rely on agile robotic systems. Agility in this context refers to the ability for robots to think, learn and adapt in order to respond to failures during task process.

To advance the agility performance of manufacturing robotics assembly systems in unstructured and dynamic environments, the Agile Robotics for Industrial Automation Competition (ARIAC) was initiated in June 2017 by the National Institute of Standards and Technology (NIST) in collaboration with the Open Source Robotics Foundation (OSRF). ARIAC is designed to be a simulation-based competition to allow competitors around the world to utilize latest advances in artificial intelligence and robot planning to address real-world industrial

* Corresponding author.
E-mail address: anthony.downs@nist.gov (A. Downs).

[1] https://www.nam.org/facts-about-manufacturing/.

https://doi.org/10.1016/j.rcim.2020.102113
Received 1 October 2020; Received in revised form 17 December 2020; Accepted 21 December 2020
Available online 12 January 2021
0736-5845/Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

challenges pertaining to kitting (or kit building) applications. The latest iteration of the competition was held between April and May 2020 and introduced a more challenging environment, a new robot, new scenarios, and new agility challenges. The annual occurrence of ARIAC is two-fold. First, NIST intends to use the results and knowledge gained from ARIAC to further its efforts to develop metrics and test methods to measure robot agility as well as tools for manufacturers to assess the agility of their robotic systems. Second, ARIAC aims to encourage competitors to develop the most effective solutions to address manufacturing processes while keeping the cost down.

This paper describes ARIAC with a focus on the 2020 iteration. The paper is organized as follows: Section 2 provides a literature review of the different efforts that address robot agility and performance metrics. Section 3 discusses well known robotics competitions which NIST investigated prior to ARIAC. Section 4 describes the evolution of the ARIAC platform and environment. Section 5 provides an overview of ARIAC components and how they interact with competitors' systems. Section 6 focuses on the metrics that were implemented to measure the performance of competitors' systems. Section 7 summarizes lessons learned from past competitions and addresses new ways to approach future competitions.

2. Agility and performance metrics

According to the Oxford dictionary,[2] agility is defined as "the ability to move quickly and easily". In manufacturing terms, agility refers to the idea of responding effectively to changing customer needs in a volatile marketplace by handling product variety and by introducing new products quickly [2,3]. In the context of this paper, we define agility as "the ability of a robot system to succeed in an environment of continuous and unpredictable change by reacting efficiently and effectively to changing factors". While there is no agreed upon definition of robot agility in the literature, this definition is consistent with proposed definitions of agility which involves not only robot agility but also agility of the manufacturing process as a whole [4,5].

Robotic systems need to be able to operate safely in collaboration with humans or other robots, be easily tasked and re-tasked, and be integrated into the rest of the enterprise seamlessly and quickly. These systems can greatly help small and medium manufacturers facing rising raw material and labor costs, stiff prices and offshore competition, quality concerns, skilled worker shortages, worker safety issues, and limited resources that hinder growth and profitability. Once the role and the definition of an agile robotic system are given, we need a way to measure the agility of such a system. Defining and measuring agility will allow manufacturers to select the right system to address challenges such as (1) swapping robots in and out without introducing extended downtime or reprogramming, (2) fast re-planning when a new order is provided to it, or (3) responding to changing environmental conditions (e.g., non-fixtured tray moves), due to new product designs.

A literature review for robot agility shows only a few examples that address assembly-type manufacturing use cases and change cases. For the assembly-type use cases, Quinn et al. [6] describe an example assembly task with four plastic parts that get snapped and inserted together. This is described as a typical light assembly task for the workcell being tested, which includes two robots working together. Frei and Di Marzo Serugendo [7] describe an example assembly of an adhesive tape roll dispenser assembly, which is a slightly more complicated assembly than the first use case as it requires a screw for locking the pieces together. The authors also described a change case in the assembly of the adhesive tape roll dispenser assembly, where the environment gets changed to a different locking method for the assembly process, in this case, changing from a screw-lock assembly method to a snap-fit method of assembly. Another change use case was described by Gou et al. [8], wherein three cases are described. The first case is used as a baseline to compare performance for the other two cases. The second case is a new high priority order coming into the system to invoke a re-prioritization. The third case is a variation on the second, but the re-prioritization is caused in this case by a machine breakdown, causing the system to adjust to absorb the workload.

Measuring the agility of a robotic system provides relevant information to manufacturers for better choosing and using robotic systems. Measuring the efficiency of robotic systems in completing a task is one of the main criteria to assess a robot system's agility. The efficiency is measured with time metrics. Time metrics for agility may consist of cycle time, planning time, and changeover time. The cycle time (amount of time per unit) is the period required to complete one cycle of an operation, or to complete a function, job, or task from start to finish. The planning time is an estimate of time the robotic system spends planning before carrying out any action that performs a task. The changeover time is the time taken by a robot to automate the configuration of the equipment settings for changing over from one product to another. Some of these time metrics have been explored by Downs et al. [9] where the authors describe multiple test methods run in different scenarios for kit building.

Although not discussed in this paper, a comparative study with human workers may be necessary to assess the agility of a robot system. For example, one can study the required person-hours needed to perform a task compared to the time taken by a robotic system to perform the same task. Certain tasks may require only two person-hours with two workers while it may take much longer with a robotic system. On the other hand, a one thousand person-hours job may be performed by a robotic system within just a few hours. There may be some cases where human workers can perform a task faster than a robot but may cost more to the company in the long term. As can be seen, assessing the agility of a robotic system is not trivial and a relative study with human workers must be considered.

3. Robotics competitions

While designing the ARIAC competition, the organizers made sure to adhere to the following guiding principles: (1) Challenges represented in the competition must mimic, as closely as possible, to the challenges that industry is facing in applying robots on their factory floors. (2) There was a low barrier to entry. In other words, the organizers did not want to require that competitors had expensive pieces of equipment in order to participate. (3) The focus must be on robot agility. While the organizers completely understood that there are other key challenges in robotics, such as perception or grasping, they did not want the focus to be on these areas. If competitors had novel approaches to address those challenges they could use them, but would not be required to do so in order to compete. (4) The organizers wanted the competition to be easily accessible to all, and not require traveling to a conference or an event to compete. The goal was to allow a team to participate from their own offices. (5) The organizers wanted the competition to involve industry, and be beneficial to industry, because the hope was that the unique approaches that came from the competition would be adopted by industry and would help to solve their robotic challenges.

Before the organizers designed the competition, they explored other similar competitions to ensure that none of them already addressed the guiding principles.

The Amazon Picking Challenge [10] was a yearly competition from 2015 to 2017, focusing on "picking". In the competition, teams have to develop robotics hardware and software that can recognize objects, grasp them, and move them from place to place. The goal was to use this competition to assess if robots would be able to do some of the menial pick and place operations that are currently performed by humans. As noted, this competition is focusing on perception and grasping, and not as much on the agility of the robot nor its ability to

[2] https://en.oxforddictionaries.com.

replan. It is also a physical competition, which required participating teams to build their own robots and travel to the competition site.

In the Virtual Defense Advanced Research Projects Agency (DARPA) Robotics Challenge, teams competed in a simulated suburban obstacle course. Twenty-six teams from eight countries qualified, which ran from June 17 to 21, 2013. Competing teams applied software of their own design to a simulated robot in an attempt to complete a series of tasks that are prerequisites for the next stages of the grand challenge [11]. The overall DARPA Robotics Challenge, which includes both the virtual and physical challenges, was launched in response to a humanitarian need that became glaringly clear during the nuclear disaster at Fukushima, Japan, in 2011. The DARPA Robotics Challenge consisted of three increasingly demanding competitions over two years. The goal was to accelerate progress in robotics and hasten the day when robots have sufficient dexterity and robustness to enter areas too dangerous for humans and mitigate the impacts of natural or man-made disasters. While the virtual nature of this part of the competition made it very accessible to the community, the focus was on humanitarian and first response robots as opposed to industrial applications.

The Robot Perception Challenge [12] was launched by Willow Garage and NIST to drive improvements in sensing and perception technologies for next-generation robots. The competition debuted at the IEEE International Conference on Robotics and Automation (ICRA) 2011 in Shanghai, China. The competition measures the performance of current algorithms that process and act on data gathered with cameras and other types of sensing devices. While perception was an important challenge in robotics, it is not one of the guiding principles of ARIAC developers.

The RoboCup Logistics League (RCLL) "is a league of the annual international robotics competition RoboCup. It focuses on in-factory logistics applications. Following the RoboCup spirit this league's objective is to enable scientific work in order to achieve a flexible solution of material and informational flow within industrial production using coordinated teams of autonomous mobile robots". [13] While this competition is very relevant, ARIAC goes a step further by introducing a wide array of agility challenges (as described later in this paper) which is outside of the scope of the RCLL competition.

In Europe, there was a recent initiative promoted (and funded) by the European Commission to foster robotic competitions with the aim of gathering advancements in robotics. In particular, The EUropean RObotics Challenge (EUROC) was running under the "Factories of the Future" program. [14] EUROC aims to spur the development of new applicable innovations in European manufacturing. It consists of three industry-relevant challenges within the scenarios of (1) Reconfigurable Interactive Manufacturing Cell, (2) Shop Floor Logistics and Manipulation and (3) Plant Servicing and Inspection. [14] These competitions are looking more at the cell level than at the robot level.

Robot Competitions Kick Innovation In Cognitive Systems and Robotics (RoCKIn) is an EU-funded project aiming to foster scientific progress and innovation in cognitive systems and robotics through the design and implementation of competitions. RoCKIn@Work [15], a subset of this competition, is looking for innovative industrial robots that can help businesses meet increasing demand from their customers. A robot will assist with the assembly of a drive axle — one component of the robot itself and therefore a step towards self-replicating robots. Tasks include locating, transporting and assembling necessary parts, checking their quality and prepping them for other machines and workers. The robots will be working interactively as personal mobile assistants in a highly flexible and continuously changing production line.

In addition to surveying different competitions, the organizers wanted to be sure that the challenges that were captured within the ARIAC simulated environment were representative of the challenges faced by industry. As such, NIST reached out to industry. Each challenge was ranked with respect to its difficulty in representing it in Gazebo (based on OSRF's feedback) from 1 to 5 with 1 being the easiest to represent, as well as its importance to industry (based on industry's feedback) from 1 to 5 with 1 being the most important. During the investigation phase, thirty-nine challenges were identified among which, six were selected to be focused on in ARIAC. These six challenges are listed in Table 1 along with their respective ratings. All of these challenges have been represented in ARIAC at some point in the past, and many of them have been represented in all of the previous competitions. Detailed descriptions of the challenges shown in Table 1 can be found in Section 5.2.

Table 1
Agility challenges in ARIAC.

Challenges	Difficulty rating	Importance rating
Parts Re-Orientation	1	1
Faulty Gripper	1	2
New Order	1	2
Faulty Part	2	2
Faulty Sensors	1	3
Human Presence	3	1

4. Evolution of ARIAC

As stated earlier, ARIAC's original purpose was to test the applicability and usefulness of the robot agility metrics developed at NIST. While a worthwhile goal in and of itself, it begged the question of how to go about this. From this need, ARIAC was born.

From the beginning, ARIAC was intended to be a simulation-based competition open to everyone. Many existing engineering competitions require large teams with deep technical knowledge of proprietary systems, or substantial funding, however, the ARIAC organizers wanted this competition to have no barriers due to resources. Research teams, hobbyists, and undergrads should all have the chance to participate. For this reason, the organizers chose both the Robot Operating System (ROS) [16] and Gazebo [17] as the platform for the competition.

4.1. ARIAC platform

The list of viable software suitable for ARIAC turned out to be relatively short. It was important that the software is free to all users, customizable, and familiar to the robotics community.

While there are simulation environments that can satisfy the first two aforementioned criteria, Gazebo and ROS are the best options for the robotics community. ROS also has the added benefit of making competition code more transferable to actual robotic control systems provided those systems also use ROS.

As the competition has progressed, ARIAC has stayed with ROS and Gazebo because no software or communication protocol has surpassed either since ARIAC 2017.

4.2. Environment

ARIAC's technical development is largely thanks to Open Robotics [18], which also created the environment for the 2017, 2018, and 2019 iterations of ARIAC. For these years, NIST supplied a 3D model of the environment and Open Robotics would create and deploy a functional simulation congruent with the model and scoring parameters. In ARIAC 2020, NIST assumed both administrative and technical responsibilities, making it the first year ARIAC was completely managed, created, and designed by NIST.

The makeup of the environment has had two major iterations (Fig. 1). The first iteration in 2017 included a single robot on a single rail used to build kits. While this central theme of building kits has not changed for all four ARIAC iterations, the narrative driving the environment has. ARIAC 2018 saw a progressive scenario where a robot is mounted inside of a shipping container fulfilling orders. This still

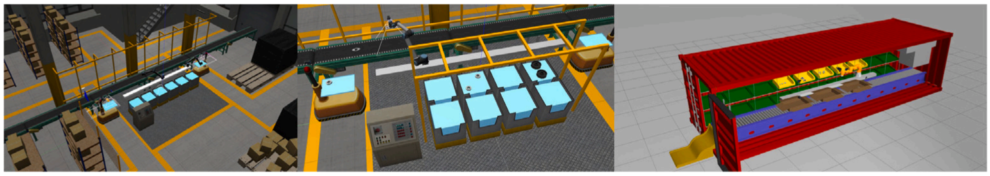

Fig. 1. Screen Renders of ARIAC 2017, 2018 and 2019 respectively.

included a single rail and single robot, however, the kit location stayed in constant motion.

The narrative for 2019 and 2020 would be the same but vary in robot degrees of freedom. ARIAC 2019 consisted of two robotic arms on a single rail while the 2020 iteration consisted of two robotic arms mounted on a single torso on a two-dimensional rail. ARIAC 2020 was also the first time robots had to avoid workers in the environment.

4.3. Control interface

The interface for ARIAC has remained the Open Robotics created GEAR (Gazebo Environment for Agile Robotics) interface [19]. The GEAR interface allowed for a controlled standardized means of communication between competitors and the simulation environment.

To maximize flexibility, GEAR was implemented to be a ROS-based interface. While the number of communication topics may have changed to accommodate a changing set of challenges, the structure has remained consistent across competitions. With GEAR, competitors implement their system in a variety of supported programming languages. Additionally, this approach was chosen to isolate the use of a simulated environment as an implementation detail. Competitors' systems never communicated directly with the Gazebo simulator, but instead, with GEAR which in turn communicated with the simulator via a Gazebo-ROS integration layer. Correctly-designed kitting systems developed to work in a simulated environment should be usable on a physical robot with minimal software modifications due to the use of an abstract ROS interface. Similarly, kitting systems developed to control a particular manipulator can be used to control another manipulator with minimal modifications if designed appropriately.

The competition interface was implemented in GEAR in a way that competitors would only have access to the permitted information from the ARIAC server during "competition mode". However, competitors could enable "development mode" to access extra information useful for debugging during the testing phase. ROS has in-built functionality for distributed systems, which facilitated blocking of non-permitted communication with the simulation during the finals.

5. The ARIAC infrastructure

This section describes the mechanisms used in trials of the competition to allow communications between competitors' systems and the competition interface. A **trial** is a single run of the simulation in which at least one order is described. The environment setup (e.g., part and part vessel locations) and some agility challenges are also defined in the trial. An **order** is an instruction with the type, the color, and the pose of each part to be placed in a kit. The order also specifies which automated guided vehicle (AGV) to use to build and deliver kits. An order has at least one shipment. A **kit** is the result of a process which groups separate but related items (parts in ARIAC) as one unit. A **shipment** is an instance of an order. If an order must be built and delivered multiple times then the order consists of multiple shipments.

Fig. 2 will be used as a description reference for the events occurring during a typical competition trial. To start a trial, a competitor's system signals that it is ready to receive orders by using a ros::ServiceClient to call the service /ariac/start_competition. Next, the first order is published on the ROS topic /ariac/orders. Competitors can retrieve the order by subscribing to the topic. Once a competitor's system receives

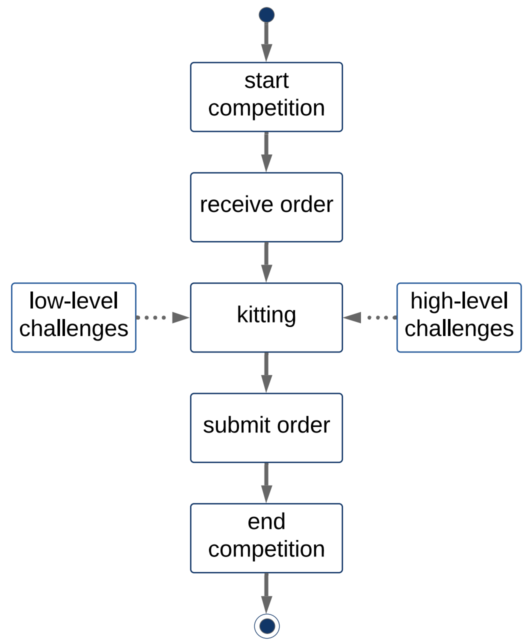

Fig. 2. Flow diagram showing the chain of events which are performed by a competitor's system during a trial. Note that there is a possibility that low-level and high-level challenges may not occur at all during a trial.

an order, it can task the robot to build the order using the competitors' methodology. During order fulfillment, agility challenges may start at specific time or in specified regions in the workcell. Once the order is completed, the AGV may then take the parts away and return with an empty tray. The competition ends when a ros::ServiceClient calls the service /ariac/end_competition. If a successful response is received, the competition ends and a score breakdown for the trial is both printed out on the standard output and logged on the competitor's machine.

5.1. Scenarios and trials

The competition was made up of a number of separately configured trials during both the qualifiers and the finals. During the qualifiers and finals, each competitor is only allowed one control approach, therefore, each control approach had to be capable of automatically re-planning itself based on the changing environment present in the trials. Trials for the finals were split into three main scenarios: Baseline Kit Building, High-priority Kit Change, and Moving Obstacle.

5.1.1. Baseline Kit Building

Baseline Kit Building scenarios are used to evaluate competitors' systems on rather simple kitting tasks where the focus is on whether or not competitors' systems are capable of performing pick and place using sensor data. These scenarios also include elementary agility challenges.

5.1.2. High-priority Kit Change

High-priority Kit Change scenarios consist of introducing a new order while the robot is already working on an order. The robot must

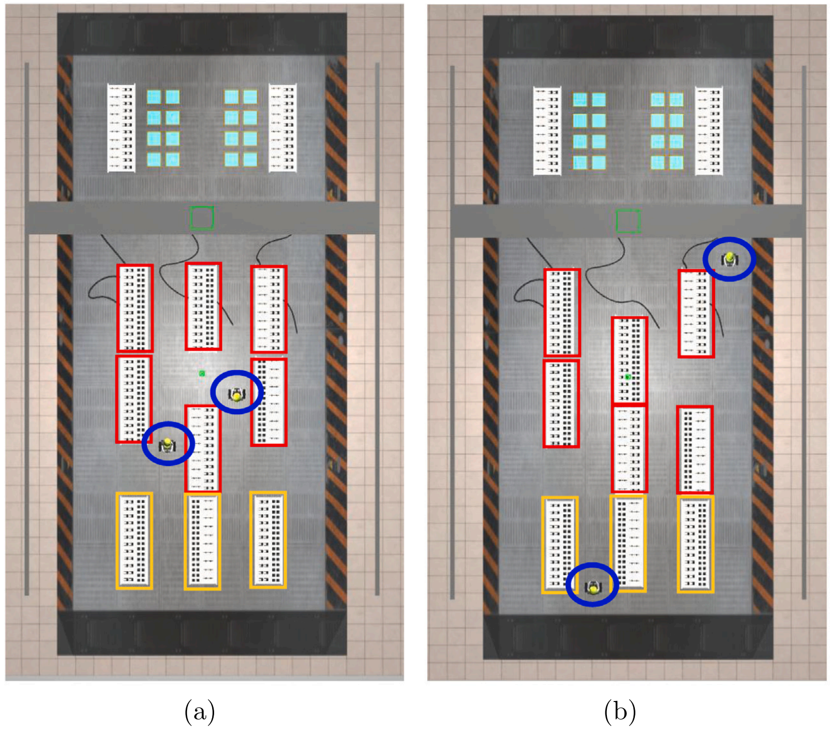

Fig. 3. In Moving Obstacle scenarios, the robot has to access parts located on stationary shelves (surrounded with yellow rectangles) while avoiding any collision with moving obstacles (depicted with blue circles). An example is provided for two configurations ((a) and (b)) for re-configurable shelves (surrounded with red rectangles). Each possible configuration of re-configurable shelves ensures there is at least one gap in the set of shelves located in the same column. For instance, in configuration (a), the gap for the set of shelves in the middle column is located right before the stationary shelf (in yellow). In configuration (b), the gap for the set of shelves in the middle row is located at the top of the shelves. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

complete the new order as fast as possible before resuming the first order. These scenarios also include low-level agility challenges and focus on competitors' systems ability to perform re-tasking when a new order is introduced.

5.1.3. Moving Obstacle

In the Moving Obstacle scenarios, competitors' systems face situations where sections of the workcell are populated with humans going back and forth in a linear pattern. These scenarios are used to test competitors' systems ability to detect human workers, to plan paths, and to generate collision-free motion commands to access parts located on shelves. Another level of complexity was added to these scenarios with the introduction of re-configurable shelves (see Fig. 3). The location of some shelves in the workcell are re-configured between trials to prevent competitors from scripting the robot path.

5.2. Agility challenges

As seen in Table 1, the latest version of ARIAC consists of six agility challenges. Each individual trial is made up of a combination of specifications of **configurable characteristics**, i.e., input variables to the *trial configuration files* that are used to initiate agility challenges. A *trial configuration file* is written in a YAML (YAML Ain't Markup Language) [20] format.

5.2.1. Part re-orientation

A part is presented to the robot in an orientation that is different than its desired final orientation. The robot needs to rotate the part around the part's x-axis before it is placed in the tray. The pulley part (see Fig. 4) is the only part in the environment designed to be used in this agility challenge. The pulley part has a flat collision surface on its top and bottom ends, making it ideal for grasping with a vacuum gripper. However, the side of the part is hollow, creating a more difficult grasp because of the small contact patch that the edges provide. Competitors are not permitted to directly grasp this part from the side when a part re-orientation is required.

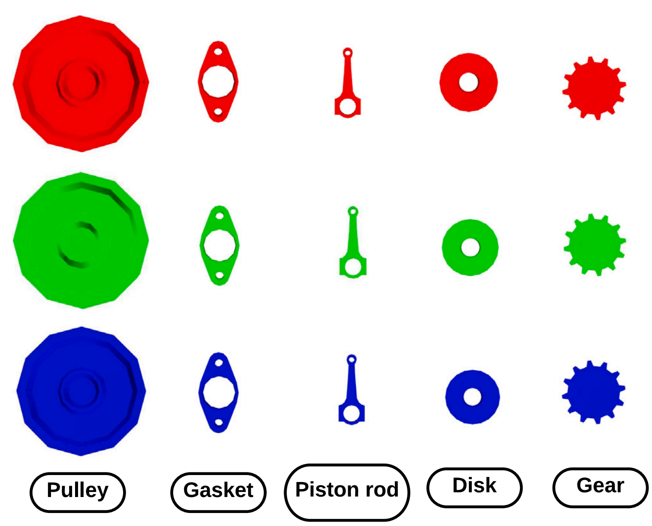

Fig. 4. All part types and colors involved in ARIAC 2020. The pulley parts are the only part types used in the part re-orientation agility challenge. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.2.2. Faulty gripper

In this challenge, as the robot is performing motions to place a part in a tray, the part drops out of the gripper and lands in the tray at a wrong location. The robot needs to determine whether to re-grasp the dropped part and replace it in the tray or to get a new one from one of the part vessels. The *trial configuration file* describes the region in the workcell and the part type the robot must be holding to activate this challenge.

5.2.3. New order

Order announcements during trials are controlled in the *trial configuration file* with an announcement condition and an announcement value. The first order is announced at the start of the competition with time and 0 for condition and value, respectively. An announcement condition can take two other separated values, namely wanted products and unwanted products. The value for each of these two conditions is an integer number n, which is used to control when a new order is announced. This agility challenge is mainly used in the High-priority Kit Change scenarios to tests the ability of competitors' systems to put the previous order on hold, to quickly complete the new order, and to resume the previous order.

How wanted and unwanted products are useful depends on how much overlap there is between the previous order and the new one. When the condition is set to wanted products, the previous order is interrupted when n products have been placed in the tray of the previous order that are also in the new order. When the condition is set to unwanted products, the previous order is interrupted when n products not in the next order have been placed in the tray of the previous order. These conditions can make interesting scenarios, such as guaranteeing competitors have to remove parts or have to re-arrange parts in the tray of the previous order.

5.2.4. Faulty part

The *trial configuration file* designates defective parts in part vessels through part IDs. Competitors are not aware of defective parts during trials. Once the robot places a part in a kit tray, a quality control sensor determines that the part is defective. The robot must dispose of the faulty part, as it does not count towards the trial score, and must get a new one from one the part vessels.

5.2.5. Faulty sensors

For a finite period of time, all sensors in the factory stop working as to mimic a sensor blackout. Competitors' systems have to use an internal world model to continue kitting. Through the *trial configuration file*, ARIAC developers have control on the duration of this agility challenge.

5.2.6. Human presence

This agility challenge is probably the most exigent challenge in the competition and is the main challenge used in the Moving Obstacle scenarios. In this challenge, up to two moving human workers can be present in the workcell at the same time. During their motions, human workers will temporarily obstruct the access to parts required in some orders. Competitors are aware of the four possible locations where workers can spawn as well as their type of motion. However, competitors need to reason about the workers location during trials. To access parts located in the workers' vicinity, competitors need (1) paramount sensor placements to detect the presence of human workers and to compute their velocity and (2) path planning for collision avoidance.

5.3. Simulation architecture

The simulation architecture consists of three main components: The GEAR interface, the ARIAC server, and plugins. Details on the GEAR interface can be found in Section 4. The ARIAC server is used to run an instance simulation of a trial, including the management of agility challenges. Plugins are programmable behaviors which can be embedded into a Gazebo simulation. Plugins are used in ARIAC to (1) initiate some agility challenges, (2) update sensor rates and publish sensor state messages, and (3) control mobile elements in the workcell, e.g., conveyor belt, human workers, and AGVs. Fig. 5 outlines the different modules for an instance of an ARIAC trial simulation run with the GEAR interface. The description of Fig. 5 is given for one competitor's system. During the qualifiers and the finals, this instance is run $n \times m$ times where n is the number of trials and m is the number of competitors.

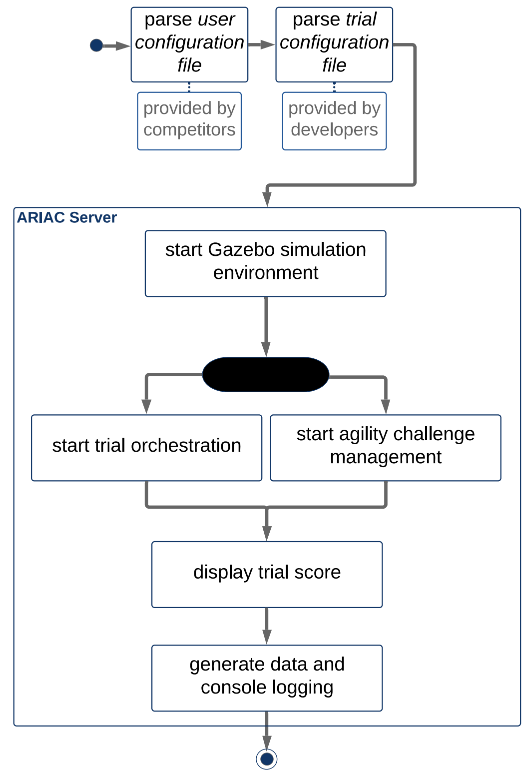

Fig. 5. An instance of an ARIAC trial run with the GEAR interface. The flowchart takes as inputs a user configuration file which for sensor configurations in the workcell as well as a trial configuration file for challenges and parts in the workcell. These two files are passed to the ARIAC server to start the simulation and to trigger the challenges. At the end of the trial a score for the trial and data logs are generated for post hoc review and evaluation.

First, the ARIAC server processes the *user configuration file* and the *trial configuration file*. The Gazebo simulation environment spawns models with poses described in the two configuration files. Sensor models are set in the workcell based on their pose information from the *user configuration file*. Models for parts, human workers, and reconfigurable shelves are set in the workcell using information from the *trial configuration file*.

Next, two modules are started in parallel. Trial orchestration allows communications between GEAR and the competitor's system, such as order announcements or order submissions. An agility challenge manager relies on the *trial configuration file* to initiate some agility challenges.

Competitors can submit an order at any time, usually when a kit is complete or partially built, using the "end competition" command. This is followed by a score break down for the current trial. An example of a score breakdown is provided in Listing 1.

Table 2
List of sensors/cameras along their outputs and costs in ARIAC 2020.

Sensor/Camera	Output	Cost
Break beam	Signal when a beam is broken by an object.	100
Laser profiler	Array of distances to a sensed object.	100
Proximity sensor	Distance of objects from sensor.	100
Depth camera	Point clouds.	200
Logical camera	Pose and type of models.	500
RGB-D camera	Point clouds and images.	500

```
<game_score>
    Total game score: [12]
    Total process time: [66.257]
    Arms collision: [0]
    <order_score order_0>
      Total order score: [12]
      Completion score: [12]
      Time taken: [66.255]
      Complete: [true]
      Priority: [1]
      <shipment_score>
        Completion score: [12]
        Complete: [true]
        Submitted: [true]
        Product type presence score: [3]
        Product color presence score: [3]
        All products bonus: [3]
        Product pose score: [3]
        Delivered to correct agv: [true]
      </shipment_score>
    </order_score>
</game_score>
```

Listing 1: Example of a score breakdown for a trial. The results for `Total game score`, `Total process time`, and `Arms collision` are used in some of the metrics Equations described in Section 6. For this trial, the competitor's system submitted only one shipment, depicted with the <shipment_score> tag, for only one order: `order_0`.

5.3.1. Sensor configurations

Through the user configuration file, competitors have control over the quantity, the type, and the pose of sensors in the workcell. An excerpt of a competitor's user configuration file is illustrated in Listing 2. There is a total of six types of sensor that are made available to competitors. Table 2 presents the different sensor/camera types with their cost and functionality.

Although competitors are free to use as many sensors as they wish, they must consider the cost of each sensor as to not end up with a sensor configuration which may be too expensive. The cost of the overall competitor system is used during scoring and is compared with other competitors' system cost. As one of the main objectives of ARIAC, competitors need to keep the cost of their system down while still demonstrating great agility of their system.

```
sensors:
  break_beam_1:
    type: break_beam
    pose:
      xyz: [1.6, 2.25, 0.91]
      rpy: [0, 0, 'pi']
  logical_camera_1:
    type: logical_camera
    pose:
      xyz: [-0.3, 0.383, 1.27]
      rpy: [0.2, 1.5707, 0]
```

Listing 2: Excerpt of a user configuration file which describes sensor types along with their locations (xyz) and orientations (rpy) in the workcell. In this example, break_beam_1 and logical_camera_1 are names chosen by the participant.

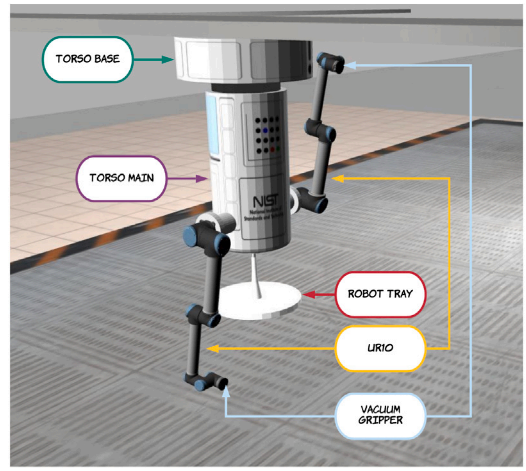

Fig. 6. Visual breakdown of the competition robot. The UR10 models were developed as part of the ROS-Industrial project, while the torso was developed in-house at NIST.

5.3.2. Robot configurations

The competition features a custom robot consisting of two UR10 industrial robotic arms mounted on a rotating torso. This assembly is connected to an overhead gantry that enables the robot to move throughout the XY plane of the workcell. In total, this system has 15 degrees-of-freedom for competitors to consider. Both arms are equipped with vacuum grippers that are independently controllable via ROS topics. Additionally, a tray is attached below the torso of the robot for extra part storage (see Fig. 6).

The kinematic properties of this robot are defined using the Universal Robot Description Format (URDF), following the typical ROS workflow. In addition to this description, competitors were also provided with a ROS package for interfacing the robot with the MoveIt motion planning framework, with support for combined and individual planning of both arms as well as the gantry. Finally, actuating the robot in simulation was accomplished using standard ROS-Gazebo compatibility plugins and ROS JointTrajectory controllers.

6. Measuring agility through evolving automated and human metrics

With the challenges determined through the help of industry and OSRF, the organizers needed a way to be able to measure the effectiveness of how a robot system was implemented to handle these agility challenges. Because of the variety of trade-offs and options that a robot system developer chooses during design, the organizers needed a set of metrics to compare the systems both against other systems as well as comparing the system across different trials where different agility challenges are in play.

6.1. Current metrics

There are currently three general metrics that can be used individually to compare systems in terms of their agility performance: Cost Factor, Completion Score, and Efficiency Factor. These three individual metrics are also combined along with some constant factors into the Ranking Score, which is the equation used for ranking the teams during the finals.

6.1.1. Cost factor

The first general metric is to compare how expensive the systems are based on the choices made. The main idea for this metric is that a lower cost system is better than a higher cost system, all other factors being equal. This cost would be a combination of the costs of the robot, the sensors, and the infrastructure. In the specific cases of ARIAC, the competitors are limited to only one robot and thus the main cost comes from the choice of sensors. Each sensor is assigned a nominal cost value based on its usefulness in the scenarios and other factors by the competition organizers. The costs of the competitor's chosen sensors are summed up and represented as shown in Eq. (1). A baseline cost (BC) is determined by the organizers by using a representative, mid-range number of sensors in the environment. This baseline cost as well as the total cost (TC) is used as shown in Eq. (2) to calculate a team's cost factor (CF).

$$TC = \Sigma_{i=1}^{n} Cost_i \tag{1}$$

$$CF = \left(\frac{BC}{TC}\right) \tag{2}$$

6.1.2. Completion score

The next general metric for comparison is the kit completion score (CS). The idea for this metric is that a submitted order should score higher. For an individual kit order submission, S_j, with i parts in the order, the following points are available:

- 1 point (up to i points) for each part of the correct type being placed in the kit tray.
- 1 point (up to i points) for each part being placed in the correct position (± 3 cm) and orientation (± 0.1 rad).
- i points awarded if the above two cases are maxed out.
- 1 point (up to i points) for each part of the correct color being placed in the kit tray.

The completion score is calculated by adding up the four categories above, and with each category having a maximum of i points, there is a maximum possible completion score of $i \times 4$, as shown in Eq. (3). In the case where a trial contains multiple different orders, each order would have a separate completion score and will be used in different places in the overall scoring equation.

$$CS_{S_j} \leq i \times 4 \tag{3}$$

6.1.3. Efficiency factor

The last of the three main comparison metrics is the efficiency factor (EF). The main idea on this metric is that in general, a faster system is better than a slower system, which leads to greater throughput overall. When an order is sent to the competitor's system, a timer is started that begins counting up until that order is completed and delivered. For a given trial, j, all of the competing teams have their time, T_j, averaged together, represented as AT_j. As an edge case, if a team's system times out (taking longer than 500 simulation seconds) their efficiency factor is set to 0 and the trial's time is not included in the average. Otherwise, the efficiency factor is calculated as shown in Eq. (4).

$$EF_j = \left(\frac{AT_j}{T_j}\right) \tag{4}$$

In the trials where there is a changeover happening, there is a separate timer being run for both the original kit order as well as the new higher priority order, so the choices that the competitors make for which kit to finish first will be measurable in the timings for both orders.

6.1.4. Constant factors and the ranking score

The above three metrics are the main factors of the ranking score formula used to rank the teams before the Judges' scores are added. The cost factor is applied to the average of the completion scores for all kits in the orders in the trial. Then, for each order within the trial, the efficiency factor is applied to the completion score for that order. For trials where a high-priority order is present, a high-priority factor ($h = 3$) is used as a bonus given for the team to prioritize finishing that order more quickly. The ranking score (RS) is calculated as shown in Eq. (5).

$$RS = (CF \times AVG(CS)) + (EF_1) \times (CS_{S_1}) + h \times (EF_2) \times (CS_{S_2}) \tag{5}$$

The ranking score for each trial for a given team is summed to get a total ranking score for the team. The total ranking score is then used to rank each team and to award points based on the rank. The team with the highest ranking score receives 80 points, the team with the second highest ranking score receives 70 points, and so on. These overall points are the majority of the final points used to determine the competition winners. The last bit of these final scores comes from the human judges, which were added starting in the 2018 version of ARIAC.

6.2. Changing cost factor over the years

The first year of the competition, in 2017, the cost factor was calculated in a different method using the average of the competitors costs and applying an exponential factor to it to attempt to spread the cost factors apart. However, at the end of the year, it was noted that one of the competitors had found a way to "game" the scoring system by using only a single sensor and some unanticipated intuition of the placement of parts within the part bins. Once the 2017 competition was over, the cost factor equation was changed to the current method. The organizers also added a panel of three human judges to provide an additional subjective evaluation of the competitors' performance and approaches. A more detailed description of the judges can be found in the following Section.

6.3. Judging panel

With the addition of the judging panel, ARIAC has added a method for the competition final scores to include some human subjective judgment to the mix. A panel of three judges are chosen from industry for each year and are asked to provide their own individual judgment on the innovativeness and feasibility of the competitors' approach. The judging panel is given access to both videos of the competitors' trials, typically narrowed down by the organizers to highlights in order to be cognizant of the judges time as well as a one page document from the competitors describing what they were intending to be innovative about their approach, or describing how they approached the general problems.

For innovativeness, the judges start with a default score of 0 out of a maximum of 10 and add points to the score for how the competitors showed themselves to have an innovative approach to the scenarios. For feasibility, the judges start with a score of 10 out of a maximum of 10 points and subtract away points based on their subjective evaluation of how feasible competitors approaches would be to implement in a real-world manufacturing plant. The three judges' scores are added together and averaged to provide the final 20% of the final score.

6.4. New metrics

Thanks to the IEEE Standards Association Study Group (soon to be a Working Group) on Measuring Robot Agility, there is a list of 10 aspects of agility that are being discussed and will likely become involved in ARIAC at some point in the future. These 10 aspects include hardware reconfigurability, software reconfigurability, communications, task representation, sensing, perception, reasoning, planning, tasking, and execution.

7. Lessons learned and the future of the competition

7.1. Lessons learned

After four years of running ARIAC, the NIST team has learned a lot and has tried to integrate what was learned in each subsequent iteration. The most recent lessons learned are as follows:

- Since NIST took over the back-end development effort from OSRF, additional development time was needed to ensure that the system ran reliably. NIST is planning to start the development process about 2–3 months earlier than was done previously to ensure that a stable system is available to the participants well before the competition is run.
- When teams identify faulty parts, they often just toss them to the side to remove them from the kit tray. While this works from a scoring perspective, it is not realistic in a factory environment. Future environments will have a faulty parts bin in which to place these parts with bonus points for doing so.
- Robots had to avoid people in ARIAC 2020, but there was no penalty for getting very close to people. In future iterations there will be a safety buffer distance from humans with a penalty for getting too close, even if the robot does not strike the people.
- On the back end, NIST will provide a better mechanism to inform participants of when a change is made to the interfaces and the environment.
- The organizers will explore additional domains in addition to order fulfillment and kitting, such as assembly and agile disaster response (process can change over to a related task to support global and national needs for public health and safety).
- The organizers will explore the possibility of an open world environment, where builders can create their own robot and perhaps other aspects of their environment to creatively solve the ARIAC challenges.

These lessons, combined with the lessons that were learned through previous years of the competition, both by OSRF and NIST have been used as the competition has progressed through the years. The results of what the organizers have observed in terms of methods and quirks of strategy used by the teams have led to the changes in the competition so far. For instance, the strategies of competitors through the years have adjusted the scoring of both the cost factor in addition to the completion scores of having the colors of parts being considered in the scoring.

Through a combination of observing the competitors each year as well as the input from the IEEE RAS Standards Working Group on Measuring Robot Agility, the metrics, scoring, and the overall themes of the competition will continue to morph and adjust as the competition continues. In this way, the competition will evolve and continue to promote greater agility within the industry.

7.2. The future of ARIAC

ARIAC remains an evolving entity with a close ear to both industry, robotics, and simulation software advancements, however, the future trajectory of ARIAC can be summed up with 4 major tenants: (1) Accessibility, (2) virtual to real deployability, (3) open medium for realistic and creative solutions, and (4) relevance to today's robotics challenges.

7.2.1. Accessibility

Accessibility in this case means maximizing the number of people capable of participation. This includes making sure the competition remains free and does not require too much specialized knowledge. For this reason, ARIAC organizers continue evaluating free simulation software solutions in addition to Gazebo. While Gazebo is a great free option, future competitions should be operating system independent. Unity 3D and other game engines and simulation environments are among those that the organizers continue to monitor.

7.2.2. Virtual to real deploy-ability

Future competitions will have a clear path to actual physical robot deployment. NIST is currently in the process of creating a real robotic system that has a parallel in the ARIAC Environment. Future teams will be able to see how their control code functions in a real robotic process.

In the future, ARIAC will use a robot communication protocol that is real–virtual and robot agnostic. NIST has been working in the area of robot communication for quite some time. In fact, the Canonical Robot Control Language (CRCL), developed by NIST, is a robot agnostic command language [21]. Future competitions will leverage NIST's expertise in this area making competitor code more relevant to current industry challenges. In the future, NIST also will be working to transition some of the winning code submissions working on physical robots as well as facilitating the use of some of these strategies into real world plants.

7.2.3. Open medium for realistic and creative solutions

Future ARIACs will seek to allow more freedom for robot system designers by implementing a more open environment. This would allow system designers to select not only the control strategy but also the robot morphology, and general system layout. While requiring more from designers, this open-world will allow for greater freedom and creativity.

7.2.4. Relevance to today's robotics challenges

The ARIAC organizers continue to look to both industry and academia for relevant challenges that address the industrial challenges of today. Future ARIACs will include assembly at various levels of abstraction. Initially, this will include placing assembly pieces in 3-dimensional orientations that do not lie on a single plane. Later iterations of ARIAC will also include how these parts are assembled, like screwing, placing, or pressing parts.

ARIAC organizers are also looking at new robot tasks such as finishing and welding. In both cases, teams will need to intelligently solve the dynamic challenges associated with these operations.

CRediT authorship contribution statement

Anthony Downs: Writing - original draft, Writing - review & editing, Methodology. **Zeid Kootbally:** Writing - original draft, Writing - review & editing, Software. **William Harrison:** Writing - original draft, Software. **Pavel Pilliptchak:** Writing - original draft, Software, Validation. **Brian Antonishek:** Writing - original draft, Resources. **Murat Aksu:** Writing - original draft. **Craig Schlenoff:** Writing - original draft, Conceptualization, Project administration, Supervision. **Satyandra K. Gupta:** Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Dr. Kootbally acknowledges support for this work under grant 70NANB19H009 from the National Institute of Standards and Technology, USA to the University of Southern California.

References

[1] W. Tuszynski, R. Michalczewski, W. Piekoszewski, M. Szczerek, Modern automotive gear oils - classification, characteristics, market analysis, and some aspects of lubrication, 2011, http://dx.doi.org/10.5772/13014.

[2] P. Lindbergh, Strategic manufacturing management: A proactive approach, Int. J. Oper. Prod. Manage. 10 (2) (1990) 94–106.

[3] H. Sharafi, Z. Zhang, A method for achieving agility in manufacturing organisations: An introduction, Int. J. Prod. Econ. 62 (1–2) (1999) 7–22.

[4] L. Jin-Hai, A. Anderson, R. Harrison, The evolution of agile manufacturing, Bus. Process. Manag. J. 9 (2003) 170–189, http://dx.doi.org/10.1108/14637150310468380.

[5] N. Bagul, M.P. Ahire, Agile manufacturing system, Int. J. Adv. Res. Sci. Eng. 5 (2) (2016) 282–290.

[6] R.D. Quinn, G.C. Causey, F.L. Merat, D.M. Sargent, N.A. Barendt, W.S. Newman, V.B. Velasco Jr, A. Podgurski, J.-Y. Jo, L.S. Sterling, Y. Kim, An agile manufacturing workcell design, IEEE Trans. 29 (10) (1997) 901–909.

[7] R. Frei, G. Di Marzo Serugendo, Designing self-organization for evolvable assembly systems, in: IEEE International Conference Self-Adaptive Self-Orgnizing Systems, Venice, Italy, 2008, pp. 97–106.

[8] L. Gou, T. Hasegawa, P.B. Luh, S. Tamura, J.M. Oblak, Holonic planning and scheduling for a robotic assembly testbed, in: Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology, 1994, pp. 142–149, doi:10.1109/CIMAT.1994.389081.

[9] A. Downs, W. Harrison, C. Schlenoff, Test methods for robot agility in manufacturing, Ind. Robot. Int. J. 43 (5) (2016) 563–572, http://dx.doi.org/10.1108/IR-01-2016-0032.

[10] N. Correll, K.E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J.M. Romano, P.R. Wurman, Analysis and observations from the first amazon picking challenge, 2016, arXiv:1601.05484.

[11] DARPA, DARPA Virtual robotics challenge results, 2013.

[12] J.A. Marvel, T.-H. Hong, E. Messina, 2011 solutions in perception challenge performance metrics and results, in: Proceedings of the Workshop on Performance Metrics for Intelligent Systems, in: PerMIS '12, Association for Computing Machinery, New York, NY, USA, 2012, pp. 59–63, http://dx.doi.org/10.1145/2393091.2393104, https://doi.org/10.1145/2393091.2393104.

[13] R.L. League, Robocup logistics league, 2018, http://www.robocup-logistics.org/.

[14] European Factories of the Future Research Association (EFFRA), EUROC, 2017, https://portal.effra.eu/project/1084.

[15] RoCKIn Project, RoCKIn, 2013, http://rockinrobotchallenge.eu/.

[16] A. Koubâa, Robot Operating System (ROS). vol. 1, Springer, 2019.

[17] N. Koenig, A. Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator, in: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), vol. 3, IEEE, 2004, pp. 2149–2154.

[18] Open Robotics, Open Robotics, 2019, https://www.openrobotics.org/.

[19] Open Robotics, ARIAC Interface, 2019, https://bitbucket.org/osrf/ariac/wiki/2019/competition_interface_documentation.

[20] O. Ben-Kiki, C. Evans, I. döt Net, YAML Ain't Markup Language (YAML), 2020, https://yaml.org/.

[21] F. Proctor, S. Balakirsky, Z. Kootbally, T. Kramer, C. Schlenoff, W. Shackleford, The canonical robot command language (crcl), Ind. Robot Int. J. (2016).