Proceedings of the 11th North American International Conference on Industrial Engineering and Operations Management, Milwaukee, Wisconsin, USA, June 9-11, 2026

Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/NA11.20260193
Published: June 9, 2026

FANUC Robotics Labs for Industry 5.0 Manufacturing: A Flipped Learning Approach in Computer Integrated Manufacturing (CIM) Course

Goksu Avdan
Assistant Professor
Department of Engineering Technology
Missouri Western State University
St. Joseph, MO, USA
gavdan@misouriwestern.edu

Abstract

Computer Integrated Manufacturing (CIM) requires graduates who can operate and program automated systems reliably under shop-floor constraints. While industrial robotics is widely adopted across manufacturing and healthcare-related production, many undergraduate experiences still rely on software-based instruction or limited lab exposure, which can leave students underprepared for practical tasks such as safe motion execution, frame setup, and repeatable point teaching. This research paper presents a full-semester, hands-on robotics instructional approach implemented in MET 232 (Computer Integrated Manufacturing) using a FANUC ER-4*i*A robot and primarily iPendant-based programming within a hybrid flipped-learning setting. The lab sequence is scaffolded from safety fundamentals, mastering and calibration, tool and user frames, and program creation with AUTO-mode testing, toward higher-level programming concepts including branching logic, subprogramming, and custom decision-making for pick-and-place tasks. Course observations indicate that student difficulties were most often tied to foundational consistency (Teach-mode discipline, coordinate/frame selection, and point recording) rather than writing program lines, and that structured labs improved student confidence and operational readiness over the semester. The approach aligns with Industry 5.0 expectations by emphasizing human-centered skill development and safe decision-making. Future improvements will integrate FANUC ROBOGUIDE to expand offline practice opportunities and will add real-world applications such as palletizing/depalletizing, vision-based decision making, quality control checks, and a robot-machining demonstration by integrating the FANUC ER-4*i*A with a FANUC ROBODRILL to highlight CIM efficiency gains.

Keywords
Computer Integrated Manufacturing (CIM), Industry 5.0, Flipped Learning, FANUC.

1. Introduction
Computer Integrated Manufacturing (CIM) is increasingly essential because it enables manufacturers to connect design, planning, production, quality, and material handling into a single data-informed workflow (Scheer 2012). This integration improves productivity, quality standards, and response times to customer demand, while also supporting safer and more reliable operations. In parallel, industrial robotics continues to expand across various sectors in both healthcare and manufacturing (Urrea and Kern 2025). Pick-and-place applications are widely used for repetitive, high-throughput handling tasks in pharmaceutical manufacturing and healthcare-related production environments, while broader robotic capabilities support welding, assembly, inspection support, and maintenance activities in aerospace and automotive manufacturing. As these technologies become standard practice, the workforce demand is shifting toward graduates who can program, operate, and troubleshoot automated systems, not only describe them conceptually.

Despite this industry need, many undergraduate learning experiences in automation and robotics remain limited to simulation-based exercises or short demonstrations that do not fully reflect real shop-floor constraints. Students often struggle to translate basic programming concepts into reliable robot behavior when they must handle practical considerations such as tool frames, user frames, safe motion planning, repeatability, and operational discipline at the teach pendant. This gap is especially visible in entry-level robotics training, where learners may memorize commands but still lack confidence in building, validating, and debugging complete robot programs under time, safety, and accuracy constraints.

There are studies in the literature that incorporate FANUC robot operations in both engineering technology and engineering courses at various institutions. However, some of these courses rely solely on software-based instruction (Coletta and Chauhan 2022), while others include only a limited number of lab assignments rather than maintaining a full-semester focus on hands-on robotic operations using FANUC robots (Mikhail and Wang 2018).

This conference proceeding addresses that instructional gap through a hands-on instructional approach embedded in MET 232 (Computer Integrated Manufacturing), where students learn robotic programming fundamentals and robot operation using a FANUC ER-4iA robot and primarily program their lab assignments through the iPendant. The motivation for this work is twofold: (1) to strengthen students' practical competence in automated manufacturing operations through structured lab activities that mirror industrial expectations within the hybrid flipped-learning setup, and (2) to support Missouri Western State University's applied learning philosophy that emphasizes learning by doing and workforce preparation. The problem statement is that, without a clearly structured, hands-on robotics learning sequence with measurable outcomes, students may complete an automation course without developing the practical programming proficiency and operational confidence needed for modern CIM environments.

1.1 Objectives

The objectives of this work are to develop and implement hands-on robotics learning modules within MET 232 using the FANUC ER-4*i*A and iPendant-based programming, and to document measurable learning outcomes that demonstrate improved student competency in automated manufacturing operations. Specifically, this paper aims to:

1. Define a structured set of robot programming and operation skills aligned with CIM needs (for example, safe motion execution, program structure, basic debugging, and repeatable task performance),
2. Design lab activities that progressively build these skills through authentic robot interaction rather than simulation-only exposure, and
3. Evaluate student performance using direct evidence from lab artifacts and practical demonstrations to identify strengths, common failure points, and instructional improvements. The unique contribution is an applied, course-embedded approach that connects CIM workforce needs to repeatable, hands-on robotics training and assessment within an undergraduate engineering technology context, reinforcing applied learning goals and broader workforce development priorities.

2. Literature Review

Industry 4.0 has accelerated the integration of cyber-physical systems, connectivity, and data analytics into manufacturing, strengthening the role of CIM as a foundation for productivity, traceability, and flexible production. Building on this, Industry 5.0 emphasizes human-centric and resilient manufacturing, where automation is designed to amplify human capabilities rather than replace them. Recent studies show that collaborative robotics, digital twins, augmentation technologies (AR/VR/MR), and AI are increasingly treated as complementary enablers of this transition, particularly for systems that require safe and efficient human-robot interaction in real production environments (Zafar et al. 2024).

Even with these advances, robot programming and deployment remain recurring bottlenecks, especially for reconfigurable systems and high-mix production. Recent research demonstrates that AR-assisted digital twin approaches can reduce programming effort and improve usability by enabling more intuitive human guidance, supporting skill-based and low-code workflows for tasks such as robotic assembly (Yin et al. 2024). In parallel, human-centric manufacturing studies emphasize that performance metrics alone are insufficient, and that worker factors such as ergonomics, workload, and interaction quality are becoming essential design targets in Industry 5.0 settings (Bucci et al. 2025). These findings reinforce the educational need to teach both fundamental robot

programming and the operational discipline required for safe, repeatable performance, while also exposing students to the broader direction of modern programming paradigms.
Healthcare 4.0 and Healthcare 5.0 extend similar ideas into medical and healthcare-adjacent systems, where connected sensing, automation, and analytics support monitoring, decision support, and operational efficiency (Avdan and Onal 2024a; Avdan and Onal 2024b). Recent studies describe Healthcare 4.0 as leveraging IoT/IoMT, cloud infrastructures, and data-driven automation, while Healthcare 5.0 places additional emphasis on human-centered design, personalization, and responsible use of AI in clinical workflows and related systems (Rashid and Nemati 2024; Younesi et al. 2025). This perspective is relevant to robotics education because pick-and-place automation is widely connected to pharmaceutical manufacturing and other healthcare-related production and logistics contexts, where consistency, traceability, and safety are critical.

Finally, the education and training literature increasingly supports blended learning structures that combine physical robot access with simulation, XR, or digital twin-enabled practice to increase repetition opportunities and improve safety. Recent studies report that mixed-reality training integrated with digital twins can improve engagement, confidence, and measurable task performance for foundational operations such as pick-and-place (Tran et al. 2025). Similarly, XR-based training frameworks support scalable practice and structured evaluation, helping bridge theoretical knowledge and hands-on readiness in industrial robotics programming (Mulero-Pérez et al. 2025). Collectively, this body of work supports course designs that preserve hands-on teach-pendant programming with real industrial robots, while selectively incorporating Industry 4.0 and 5.0-aligned tools that expand practice time and strengthen competency-based assessment.

3. Methods

In this section, we are going to detail the implementation of the lab assignments that were used in the Fall 2025 semester in the MET-232 course. These assignments were improved and newly developed by taking our lessons from the Fall 2024 semester, where we also used the FANUC ER-4*i*A. Both Fall 2024 and Fall 2025 were instructed under a hybrid flipped learning approach, where students were instructed outside of classroom time for the lecture portion; however, important topics were still covered in person, along with instructor-led discussions. Afterwards, during the lecture time, within the feasible time left, in-class exercises were assigned to the students to ensure they finish on time. If the time was not enough (as anticipated by the instructor), then an extension was given to the students by the end of the day.

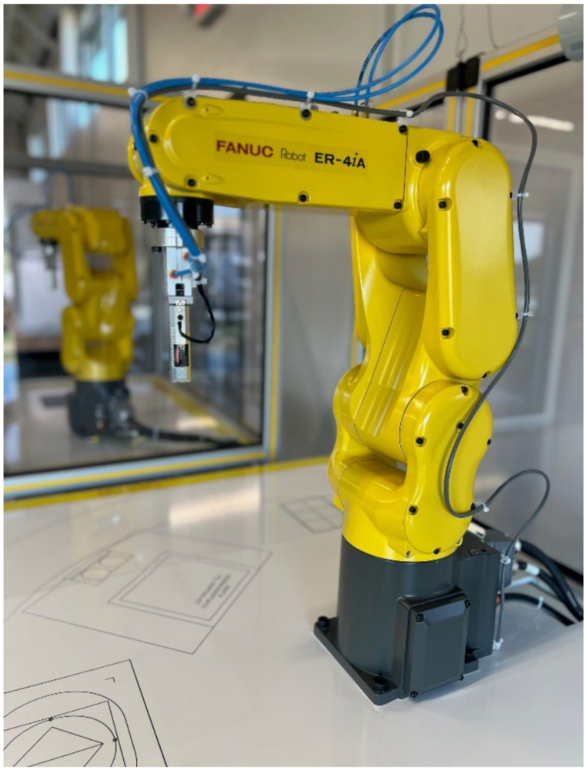

Figure 1. FANUC ER-4*i*A

The lab time, which is the second part of the class, was instructed with the FANUC ER-4*i*A (see Figure 1). Both lectures and labs had their own scheduled times on specific days. It is also important to note that continuous improvement on both the instructional methods and the lab assignments is still ongoing. In the lab assignments, the instructor used the iPendant to teach robot programming. In Fall 2026, the main plan is to utilize ROBOGUIDE, which is an offline programming and simulation software tailored specifically for FANUC robots. This will take the teaching strategy to another level and is expected to increase student experience by providing more hands-on opportunities, aligning with Missouri Western State University's applied learning goals. This effort will also support local workforce development, which was a main goal of establishing the Houlne Center for Convergent Technology in St. Joseph, MO.

Here is the full list of lab assignments involving the FANUC ER-4*i*A robot, as shown in Table 1. Then, we detail each one of them under their own section.

Table 1. Summary of MET 232 lab assignments and topics covered using the FANUC ER-4iA

Labs	**Topics Covered**
Lab 1	Practical understanding of the FANUC ER-4iA, essential iPendant functions, introduction to robot axes, range of motion, payload limits, and safety.
Lab 2	Mastering and calibrating the FANUC ER-4iA.
Lab 3	Tool frames, including setup using the Direct Entry and Four-Point methods.
Lab 4	User frames, including setup using the Three-Point and Four-Point methods.
Lab 5	Creating and editing programs, recording positions, and testing programs in AUTO mode on the FANUC ER-4iA.
Lab 6	Additional practice on recording positions, using the COPY function to reuse recorded positions, basic pick-and-place instructions, and testing programs in AUTO mode on the FANUC ER-4iA.

Lab 7	Branching instructions (LBL[ ], JMP LBL[ ], CALL[ ], IF) and using pre-recorded positions in a new program on the FANUC ER-4iA.
Lab 8	Creating an original pick-and-place operation.
Lab 9	Using subprograms within pick-and-place operations.
Lab 10	Applying learning through custom logic and instructions for pick-and-place operations (Part 1).
Lab 11	Applying learning through custom logic and instructions for pick-and-place operations (Part 2).

Lab 1: Robot Overview and Safety Fundamentals

In Lab 1, students are introduced to the FANUC ER-4*i*A and the iPendant as the primary interface for basic robot operation and motion control. The lab begins with iPendant-focused tasks that help students become comfortable with core functions required for safe jogging and interaction with the robot. Students answer guided questions covering key concepts such as who receives motion control when the iPendant is enabled, how the DEADMAN switch must be held to allow motion, the purpose of the SHIFT button, and how to adjust the jogging speed during operation. The lab also familiarizes students with the iPendant interface layout by having them identify the number of soft keys and the correct key combination needed to access the split-screen display menu.

Lab 2: Mastering and Calibration

In Lab 2, students learn why mastering and calibration are required for accurate and repeatable robot motion on the FANUC ER-4*i*A. The lab starts by explaining that the robot controller relies on a closed-loop servo system, where the serial pulse coder (encoder) sends position feedback to the controller so it can compare actual motion to command signals. For the robot to move correctly to recorded points, the controller must *know* the true joint positions. This is established by mastering, which records encoder counts at a known mechanical reference point (witness marks) and stores that information as user data maintained by the backup battery. Students also learn that calibration is the automatic process performed at power-on, where the controller retrieves the stored mastering data and synchronizes the servo system for normal operation (Figure 2).

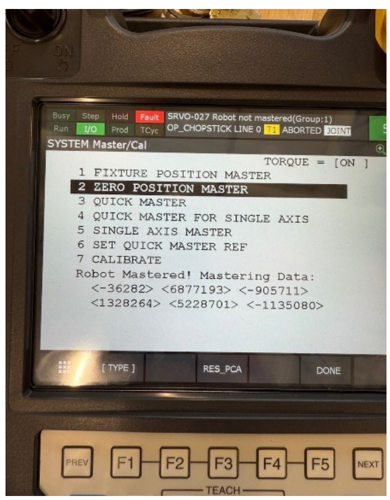

Figure 2. Mastering and Calibration

Before any movement, the lab requires safety and readiness checks, including verifying the cell guarding, confirming e-stops are functional, switching the iPendant to Teach mode, using the DEADMAN switch properly, and clearing the workspace. The technical portion begins by enabling the Master/Cal function (if needed) through the $MASTER_ENB system variable. Students then confirm the current mastering values by checking $DMR_GRP[1] and comparing $MASTER_COUNT and $SPC_COUNT for each axis using the double-view feature. Next, students perform Zero Position Master by jogging each joint to align witness marks and then executing the mastering procedure through the Master/Cal menu. They record the updated counts, complete the CALIBRATE step, and document the mastered position with a photo as evidence. Finally, students verify and apply DCS updates (which typically triggers a system alarm), enter the required code, and perform a Cold Start to ensure the updated safety parameters and mastering data are properly applied. The lab ends with short conceptual questions that reinforce why mastering is needed, what changes after Zero Position Master (see Figure 2), the difference between mastering and calibration, and what happens if encoder backup power is lost.

Lab 3: Tool Frame Setup
Students learn how to define and validate tool frames (TCP) on the FANUC ER-4*i*A using two common methods: Direct Entry and the Four-Point method. The lab begins with a short technical overview explaining when each approach is appropriate. Direct Entry is used when the tool geometry is known and students can manually enter the TCP offset values relative to the faceplate, which is typically efficient for rigid and symmetric tools. In contrast, the Four-Point method is used when tool dimensions are not precisely known but the tool tip can be touched to a fixed reference point. By recording the same physical point from multiple wrist orientations, students allow the controller to compute the TCP and tool frame based on measured samples, reducing measurement error and guesswork.

After completing required safety checks (guarding, e-stops, Teach mode, DEADMAN use, and workspace clearing), students complete both tool-frame setup methods in structured steps. For Direct Entry, students operate the gripper through the I/O Robot Out screen, secure an object, and then define a tool frame in the Frames menu by selecting Direct Entry, adding an identifying comment, measuring the faceplate-to-tip distance, and entering the Z value (or additional values if needed). For the Four-Point method, students again secure the object and select Four Point as the tool-frame method, then record four approach points to the same fixed reference point using different wrist orientations. They document the calculated results (Mean Error and Max Error) and validate the frame by activating the newly created tool frame and rotating about X, Y, and Z in TOOL coordinates. The lab ends with a short set of conceptual questions to confirm that students understand safe jogging requirements, coordinate selection using COORD, correct conditions for choosing Direct Entry vs. Four Point, and how to interpret validation behavior (for example, TCP drift indicating that the tool frame needs correction).

Lab 4: User Frame Setup
Students develop a practical understanding of User Frames on the FANUC ER-4*i*A and learn why user frames are critical for recording and reusing positional data in robot programs. The lab introduces the idea that a user frame is the programmer-defined reference frame for the workpiece or fixture, and if a user frame is not defined, the robot defaults to the World frame. Students review how a user frame includes both a physical offset (X, Y, Z) and an orientation (W, P, R), and they learn that selecting User as the jog method changes jogging behavior so that motion aligns with the active user frame rather than the robot base coordinates.

After completing safety and readiness checks (guarding, Teach mode, DEADMAN use, and workspace clearing), students create user frames using both the Three-Point and Four-Point methods. In the Three-Point method, students record three taught points (origin, +X direction point, and a point on the X–Y plane for +Y direction) to define the frame location and orientation, then test the frame by activating it and jogging in User coordinates to confirm expected motion behavior. In the Four-Point method, students repeat a similar process but also record a System Origin point, which explicitly sets the origin of the user frame and improves control over the frame's location (offset) relative to the World frame. Students then compare the resulting X, Y, and Z values from both methods and answer concept questions focused on real-world use cases (fixture offsets, multi-station setups, conveyors/workcells), common teaching errors (mis-taught origin or direction points), and the difference between User Frames and Tool Frames.

Lab 5: Program Creation and AUTO Mode Testing
In Lab 5, students develop a practical understanding of User Frames and why they matter for recording positional data in robot programs. The lab emphasizes that a user frame is the reference frame for all recorded positions, it is taught by the programmer, and only one user frame can be active at a time. Students also learn that if a user frame is not defined, the robot defaults to the World frame, which can make programming harder when the workpiece is rotated or relocated in the cell.

After completing safety and lab readiness checks, students create user frames using both the Three-Point and Four-Point methods. In the Three-Point method, they teach the frame by recording an Orient Origin Point, an X Direction Point, and a Y Direction Point (on the X–Y plane), and then test the frame by jogging in User coordinates to verify that motion follows the intended workpiece directions. In the Four-Point method, they repeat the same process but also record a System Origin Point, which explicitly sets the frame origin and helps separate "frame orientation" from "frame location (offset)." The lab closes by having students compare the resulting X, Y, and Z values between the two methods and reflect on common teaching errors and practical use cases such as fixtures, multi-station pallets, and workcells.

Lab 6: Position Management and Pick-and-Place Basics

In Lab 6, students build on their earlier frame setup skills by focusing on recording positions, reusing previously taught points, and writing a basic pick-and-place sequence on the FANUC ER-4*i*A (see Figure 3). The lab starts with creating a new program, then introduces motion instructions and what they control in a program (motion type, positional data, speed, and termination). Students are also required to confirm the correct frame setup for the exercise (Tool Number and User Frame settings) before they begin recording motion points, since the frame selection directly affects the accuracy and repeatability of recorded positions.

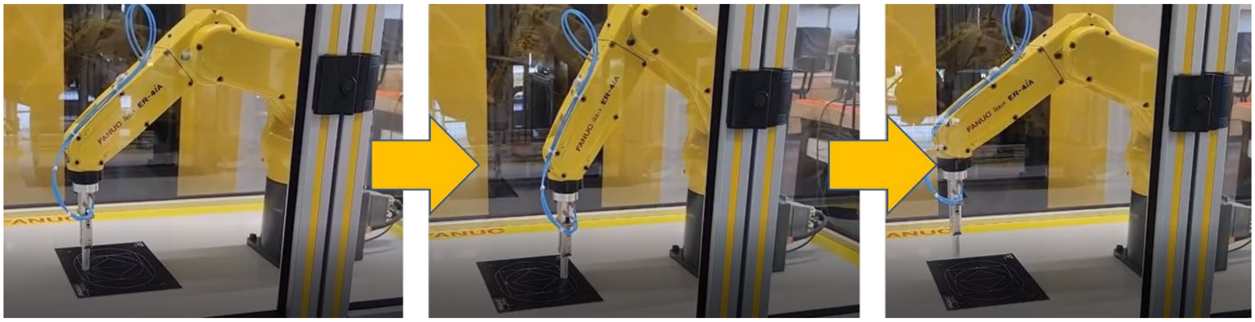

Figure 3. Pick-and-place example from a student lab assignment

After the program setup, students record several target locations for the pick-and-place operation, insert I/O commands to control the gripper (open and close), and include a short WAIT instruction to improve timing and consistency. A key learning goal in this lab is using the COPY function to reuse previously recorded positions instead of reteaching the same points repeatedly, which reinforces efficiency and good programming habits. Finally, students test their program in AUTO mode by disabling the iPendant, resetting faults, and running Cycle Start, and they also learn the ABORT function to safely stop a program that did not reach the END statement before running another program. The lab ends with reflection questions that reinforce differences between STEP vs. AUTO modes, FINE vs. CNT termination, and Joint vs. Linear motion types.

Lab 7: Branching Logic and Program Flow

In Lab 7, students learn how to control program flow on the FANUC ER-4*i*A by using branching instructions and logical decision-making. The lab introduces the purpose and usage of LBL[ ], JMP LBL[ ], and CALL[ ] instructions, and explains how these tools allow a program to repeat sections, skip lines based on conditions, or execute a separate routine and then return to the main program automatically at the next line after the call (see Figure 4). Students also learn how data registers (R[ ]) function as numeric storage locations that can be used for counters, timers, part counts, and logic variables depending on the programming goal.

After creating a new program using the required naming format, students record two taught points and then build a simple logic sequence using a register (for example, R[200]) as a counter. They define LBL[1] as the looping point, use an IF statement to compare the register value to a constant, and then jump to LBL[2] when the condition is met. If the condition is not met, they increment the register value using an arithmetic operation, call a subprogram (for example, SUBPROGRAM1), and then jump back to LBL[1] to repeat the loop. Students also practice using a predefined Position Register (PR[ ]) and compare it to program-local taught points P[n], reinforcing the advantage of reusable global positions across multiple programs. The lab ends with running the program in AUTO mode and learning how to use ABORT (ALL) if a previously executed program did not reach the END statement, so that a new program can be safely made active.

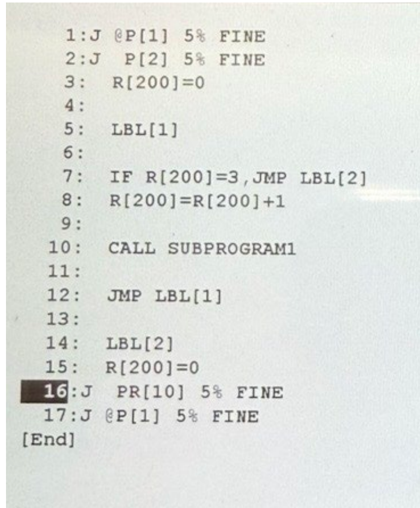

Figure 4. Example of FANUC program flow control using LBL[ ], JMP LBL[ ], and CALL[ ] instructions for looping, conditional branching, and subprogram execution.

Lab 8: Custom Pick-and-Place Operation

In Lab 8, students apply their accumulated skills on the FANUC ER-4*i*A by writing their own pick-and-place program from scratch, rather than following a fully guided template. The lab requires students to create a complete sequence that picks an object from a defined reference point, places it at a second reference point, waits for a fixed time, and then picks the object again from the same location without moving before returning it back to the original pickup point.. This structure reinforces the importance of program flow, repeatability, and consistent point usage, especially for the “wait then pick again without moving” step, which tests whether students truly understand how taught points and motion instructions behave during execution.

In addition to writing the program, students must document their work through multiple deliverables, including the program file, a line-by-line explanation of how the code executes, and iPendant evidence (screenshot or typed code in the report), and they also test the program in AUTO mode and then ABORT and switch back to Teach mode at the end. The lab also includes short conceptual questions connected to automated assembly topics (for example, assembly system configurations, parts delivery components, and how defective parts affect performance), helping students connect hands-on robot programming to broader CIM concepts covered in the course.

Lab 9: Subprogramming for Pick-and-Place

In Lab 9, students extend their pick-and-place programming skills by incorporating subprogramming on the FANUC ER-4*i*A to improve code organization and reusability. The lab requires students to write their own main program that uses a loop to call SUBPROGRAM1 (see Figure 5) and execute it two times, reinforcing the idea that repeated actions should be handled through structured logic rather than duplicated lines of code. After completing the loop, students continue the main sequence by picking up an object at Reference Point 1, moving to Reference Point 3 to pick another object and drop it into a cup, and then repeating a similar sequence at Reference Point 4 to place another object into the cup. They then test the full program in AUTO mode, and properly ABORT the program and return to Teach mode at the end of the activity.

In addition to the programming task, students document their work through a structured set of deliverables, including the final program file, a line-by-line explanation describing how the program executes, and iPendant evidence (either a screenshot or typed code in the report). The lab also requires a short execution video submission to confirm the program runs correctly in the physical cell environment. Finally, the lab includes a brief set of quality-related questions (for example, control charts, cause-and-effect diagrams, and Six Sigma concepts) to connect hands-on automation tasks with broader CIM topics tied to process control and continuous improvement (Figure 5).

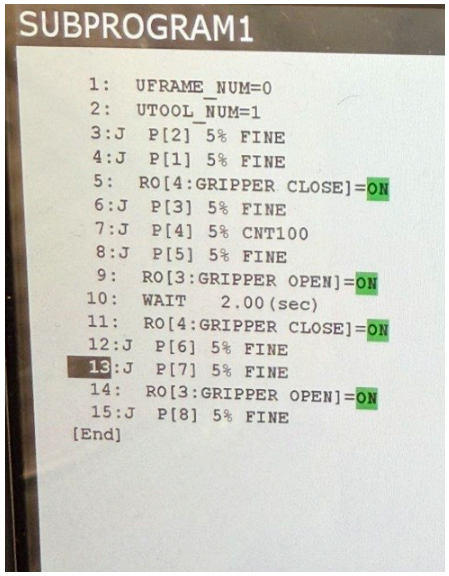

Figure 5. Subprogram

Lab 10: Custom Logic for Pick-and-Place Part 1

In Lab 10, students apply their learning on the FANUC ER-4*i*A by creating a program that uses custom logic to control how many objects are picked and placed into a cup. At the beginning of the program, students define a user-selected number using a data register (R[200]). Based on the value of this register, the robot performs different actions: if the number is 1, it picks and places Object 1 into the cup; if the number is 2, it places Objects 1 and 2; if the number is 3, it places Objects 1, 2, and 3. If the number is greater than 3, the robot does not move at all, which reinforces the importance of safe decision-making and proper condition handling in automated systems (Figure 6).

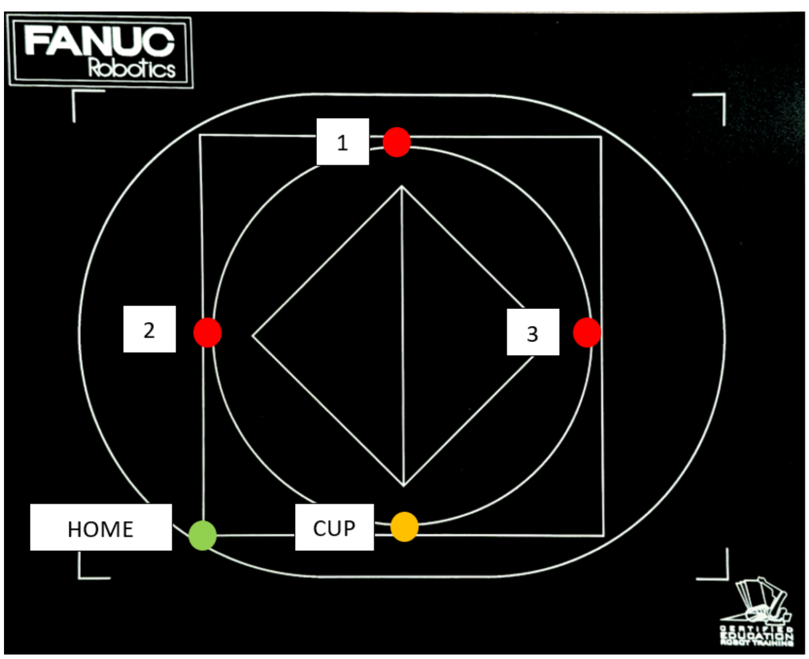

Figure 6. Reference points for Lab 10

The lab also emphasizes good programming practice through required deliverables. Students submit the final program file, provide a line-by-line explanation describing how the program executes, and include iPendant evidence (either a screenshot or typed code) following the course formatting guidelines. As shown in Figure 6, the lab uses clearly defined reference points (HOME, CUP, and object locations) so students must combine accurate taught positions with decision logic and a reliable return to HOME as the final step in the program.

Lab 11: Custom Logic for Pick-and-Place Part 2

In Lab 11, students further strengthen their programming and logic skills on the FANUC ER-4*i*A by building a more structured pick-and-place program using input validation, loop logic, and error handling. At the beginning of the program, students assume R[200] already contains a user-selected value, then they validate whether this value is within an acceptable range (between 1 and 3). If the value is invalid (less than 1 or greater than 3), students must use an IF statement and JMP LBL instruction to redirect the program to a dedicated error-handling label (for example, LBL[90]). If the value is valid, the program continues into the main logic, where students simulate a FOR-loop using LBL, IF, and JMP LBL instructions while treating R[200] as a counter that is decremented after each operation.

The loop behavior is intentionally designed to test conditional execution and program flow. When R[200] > 1, students implement a pick-and-place cycle that returns the object back to the original reference point, including a short wait and gripper actions to reinforce consistent sequencing and timing. When R[200] = 1, the robot performs the final cycle by gently placing the object into the cup and returning to HOME. When R[200] = 0, the loop exits and the robot returns to HOME to end the program. In the separate error-handling section, students move the robot to HOME, use a brief wait, bring the robot to the zero position (witness marks aligned at 0 degrees), and then return to HOME to end the program safely. The lab reinforces that real robotic automation requires not only correct motion points, but also safe logic for unexpected inputs and controlled recovery behavior.

4. Results and Discussion

Across the full-semester sequence, the most common student struggles were related to foundational setup and consistency, rather than typing the program lines. Early on, students tended to underestimate how sensitive the robot behavior is to correct Teach-mode discipline, proper use of the DEADMAN, and consistent jogging habits. As the labs progressed into tool frames and user frames, a frequent challenge was mixing coordinate systems (World vs. Tool vs. User) and then recording points under the wrong active frame, which later caused “mysterious” offsets during execution. Another recurring issue was debugging under time pressure, where students would change multiple things at once (frame, speed, termination, I/O timing) and then struggle to isolate the true root cause of an error.

One clear advantage of the assignment design is that it provides a scaffolded pathway from robot awareness and safety into reliable program execution, and then into more advanced programming concepts (branching, subprogramming, and custom logic). Students generally improved their confidence once they saw that a correct frame setup and consistent point teaching produce repeatable motion, especially when they moved from manual jogging to AUTO mode testing. The structure also encouraged good habits such as reusing taught points (COPY, PR usage), documenting work clearly (line-by-line explanation), and connecting robot behavior to CIM concepts such as quality and process control. At the same time, a limitation of a hands-on, teach-pendant-centered approach is that learning can be constrained by physical access, lab scheduling, and the time required for safe one-by-one practice, which can slow repetition opportunities for students who need extra attempts.

From a pedagogical perspective, our approach reflects key findings reported in recent education and training studies: students benefit when they can practice meaningful robot tasks repeatedly, receive structured feedback, and connect programming steps to real cell behavior rather than only simulated motion (Tran et al. 2025; Mulero-Pérez et al. 2025). Other recent studies also highlight that digital-twin and AR/XR-supported workflows can reduce barriers and increase training scalability, which speaks directly to the main weakness of strictly physical-lab delivery (limited seat time) (Yin et al. 2024; Tran et al. 2025). In that sense, our course design is strong in hands-on realism and operational discipline, but it can be further strengthened by adding structured offline or simulated practice opportunities to increase repetitions without increasing safety risk or lab congestion.

Conceptually, our approach also matches broader Industry 4.0 and Industry 5.0 directions that emphasize both technical capability and human-centered outcomes (Li and Duan 2025). In addition to programming accuracy and task completion, CIM education should also help students recognize that automation performance is connected to workplace performance, ergonomics, and process improvement, which are increasingly important in human-centered manufacturing environments (Roy et al. 2024). Recent studies highlight the growing role of human-robot collaboration, digital twins, and human-centric system design as manufacturing shifts toward more resilient and worker-aware automation environments (Zafar et al. 2024; Bucci et al. 2025; Avdan 2026). By requiring students to validate frames, handle branching logic safely, and implement input validation and error-handling behaviors, the assignments move beyond "making the robot move" and toward reliable automation behavior that resembles real shop-floor expectations. The remaining challenge is to systematically scale learning access while maintaining the same level of hands-on competency and safety discipline.

5. Future Directions

In future offerings, the main improvement plan is to integrate FANUC ROBOGUIDE more intentionally to provide offline access and increase repetition opportunities, while still keeping the iPendant-based lab experience as the core competency target. Additional real-life examples will also be incorporated, such as palletizing and depalletizing patterns, basic vision-based detection to support decision-making, and simple quality control checks that connect robot motion to inspection outcomes. Another major direction is to integrate the FANUC ER-4iA with a FANUC ROBODRILL demonstration to show how a coordinated autonomous system (robot tending plus machining) can increase efficiency and consistency in CIM environments. These additions will better align the course with Industry 4.0 and Industry 5.0 expectations and also strengthen the link between robotics programming, data-informed decision making, and practical workforce skills (Zafar et al. 2024; Yin et al. 2024).

6. Conclusion

This conference proceeding presented a full-semester, hands-on robotics instructional approach implemented in MET 232 (Computer Integrated Manufacturing) using the FANUC ER-4*i*A and iPendant-based programming. The lab sequence was designed to move students from safety and operational fundamentals into frame setup, accurate position teaching, program creation and AUTO-mode testing, and finally into higher-level programming concepts such as branching logic, subprogramming, and custom decision-making. Overall, the approach addresses a common gap in robotics education where students may learn concepts through software-based instruction or limited lab exposure but still struggle to develop reliable, shop-floor-ready programming and troubleshooting skills. The results and observations from this implementation indicate that a structured and progressive set of lab assignments can strengthen student confidence and competency in automated manufacturing operations, while aligning with applied learning and workforce development goals.

References

Avdan, G., Ergonomics Sensing in Industry 5.0 Human-Robot Collaboration: Current Trends and Future Directions, *Proceedings of the 11th North American International Conference on Industrial Engineering and Operations Management (IEOM)*, Milwaukee, Wisconsin, USA, June 8–11, 2026

Avdan, G. and Onal, S., Lean thinking in Healthcare 5.0 technologies: An exploratory review, *Proceedings of the 9th North American International Conference on Industrial Engineering and Operations Management (IEOM)*, 2024a.

Avdan, G. and Onal, S., Resilient Healthcare 5.0: Advancing human-centric and sustainable practices in smart healthcare systems, *IISE Annual Conference Proceedings*, pp. 1-6, 2024b.

Bucci, I., Fani, V. and Bandinelli, R., Towards human-centric manufacturing: Exploring the role of human digital twins in Industry 5.0, *Sustainability*, vol. 17, no. 1, Art. no. 129, 2025, https://doi.org/10.3390/su17010129

Coletta, J. A. and Chauhan, V., Teaching industrial robot programming using FANUC ROBOGUIDE and iRVision software, *Proceedings of the 16th International Multi-Conference on Society, Cybernetics and Informatics (IMSCI)*, pp. 12-15, Virtual, July 2022.

Li, L. and Duan, L., Human centric innovation at the heart of Industry 5.0: Exploring research challenges and opportunities, *International Journal of Production Research*, pp. 1-33, 2025.

Mikhail, M. and Wang, S., Lab design of FANUC robot operation for engineering technology major students, *Proceedings of the ASEE Annual Conference & Exposition*, June 2018.

Mulero-Pérez, D., Zambrano-Serrano, B., Ruiz Zúñiga, E., Fernandez-Vega, M. and Garcia-Rodriguez, J., Enhancing robotics education through XR simulation: Insights from the X-RAPT training framework, *Applied Sciences*, vol. 15, no. 18, Art. no. 10020, 2025, https://doi.org/10.3390/app151810020

Rashid, S. and Nemati, A., Human-centered IoT-based health monitoring in the Healthcare 5.0 era: Literature descriptive analysis and future research guidelines, *Discover Internet of Things*, vol. 4, Art. no. 26, 2024, https://doi.org/10.1007/s43926-024-00082-5.

Roy, D., Cho, S. and Avdan, G., Data Envelopment Analysis (DEA) for improving ergonomics and workplace performance, *Journal of Applied Research on Industrial Engineering*, vol. 11, no. 1, pp. 24–36, 2024.
Scheer, A. W., *CIM Computer Integrated Manufacturing: Towards the Factory of the Future*, Springer Science & Business Media, 2012.
Tran, T. et al., Empowering industrial robotic training with kinesthetic learning and digital twins in human-centric industrial systems, 2025.
Urrea, C. and Kern, J., Recent advances and challenges in industrial robotics: A systematic review of technological trends and emerging applications, *Processes*, vol. 13, no. 3, Art. no. 832, 2025.
Yin, Y., Zheng, P., Li, C. and Wan, K., Enhancing human-guided robotic assembly: AR-assisted digital twin for skill-based and low-code programming, *Journal of Manufacturing Systems*, vol. 74, pp. 676-689, 2024, https://doi.org/10.1016/j.jmsy.2024.04.016
Younesi, A. et al., HealthCare 5.0: An Industry 5.0 perspective for next-generation medical systems with synergistic integration of IoT, AI, and 6G, *Internet of Things*, 2025.
Zafar, M. H., Langås, E. F. and Sanfilippo, F., Exploring the synergies between collaborative robotics, digital twins, augmentation, and Industry 5.0 for smart manufacturing: A state-of-the-art review, *Robotics and Computer-Integrated Manufacturing*, vol. 86, Art. no. 102769, 2024, https://doi.org/10.1016/j.rcim.2024.102769

Biography

Dr. Goksu Avdan is a tenure-track Assistant Professor in the Department of Engineering Technology at Missouri Western State University. He earned his PhD in Engineering Science with a specialization in Industrial and Quality Engineering from Southern Illinois University Carbondale, and he holds an MS in Industrial Engineering from Southern Illinois University Edwardsville. He completed his BS in Industrial Engineering through a dual diploma program with Southern Illinois University Edwardsville and Istanbul Technical University, along with a minor in Mathematical Studies. Before joining Missouri Western, he served as an Instructor in the Department of Industrial Engineering at Southern Illinois University Edwardsville. His research spans human factors and ergonomics (HFE), human-robot collaboration (HRC), artificial intelligence and machine learning (AI/ML), biomechanics, and biomedical signal processing, with applications in Industry 5.0/Healthcare 5.0 systems. His work has been supported by competitive funding, including a Missouri Western Summer Fellowship Faculty Research Grant and co-authored support through the SIUE Research Equipment and Tools Funding Program. He has also received multiple research and travel awards, including an IEEE BHI NSF student travel award and SIUE graduate research grants. He was recognized with the Outstanding Teaching Assistant Award at the doctoral level and has remained active in professional service through peer review and conference leadership roles. He was also recognized with the Alpha Chi Compass Award as a Most Influential Professor.