THE TEACH PENDANT IS DEAD,

Long Live the Teach Pendant!

READY ROBOTICS

Introduction

The tools for programming and controlling industrial robots have evolved over the last several years, but despite recent advances in robot control, one component has remained fairly constant - the robot teach pendant. Today's teach pendants are brand specific, hard to use, and lack basic functions and features we expect from our everyday devices. Even a function such as "Copy/Paste" is an impossibility on many teach pendants being sold on the market today. It's time for the teach pendant to transform into something befitting the capabilities of modern technology, since direct user control of a robot remains a critical part of robotic operations.

One might ask, why is a teach pendant even necessary in light of tools such as Offline Robotic Programming (OLRP) software? In this whitepaper we'll share 10 reasons the robot teach pendant will continue to play an important role in robot control systems. As such, by making teach pendants and their software better in a revolutionary way, we can:

- Reduce or eliminate the training necessary to work with robotic systems
- Increase the speed at which robots can be programmed
- Reduce downtime by simplifying the analysis and resolution of problems
- Increase the number of robotic systems deployed, including into industries where they don't exist today

We can realize these benefits with teach pendant and control software that:

- Provides a single way to control all industrial robots. There are over 70 brands of robots, each with their own teach pendants and control software.

- Eliminates the need for complex robot programming languages. Almost every robot has a programming or scripting language that requires extensive training to use.

- Has an intuitive touch screen interface rather than complex text-based menus accessed through the keypad. A teach pendant should be as easy to use as a smartphone.

There are 10 key reasons a teach pendant will continue to be necessary despite advances in robot programming and control:

1 *Safety*

To program a non-collaborative industrial robot, a three-position switch (also known as a dead man's switch) is required. The teach pendant enables the user, through the use of a key, to change the operating mode of the robot from a teaching mode to unrestricted operation. Even with the rise of collaborative robots, there are still reasons the robot will stop itself through "protective stops" that must be cleared by using the teach pendant. Every system is prone to failures, so an operator will need to take control of the robot and move it to a safe position for an operator to enter the cage and or to move the robot. A teach pendant is still the most efficient way to perform these operations.

2 *Monitoring of alarms and errors when running*

While a program is running, it is important to monitor the robot and the associated components in the workcell. The teach pendant is the window into the robot's operation and where it is at in terms of the overall program control.

3

Testing a new program

While developing a new program for a robot, whether done on the teach pendant itself or via a third-party tool such as an OLRP application, program must be tested at a slow speed under operator supervision. During this testing, the operator is checking for clearance between the robot and the hardware in the workcell (such as machine tools). Other activities done using the teach pendant while testing:

- Running the robot slowly and confirming how it navigates through the work cell
- Walking through lines or program sections one step at a time
- Teaching base frames and tool frames
- Validating and confirming all sections that the robot and its tools are acting upon

4 “Touching up” programs

“Touch ups” are the process of making adjustments to a robot program to fix issues due to drift or due to simulated programming not exactly matching the real world. Variations of less than 1mm are often all the difference between a successful execution of a weld and one that fails.

Common issues that can necessitate “touch ups”:

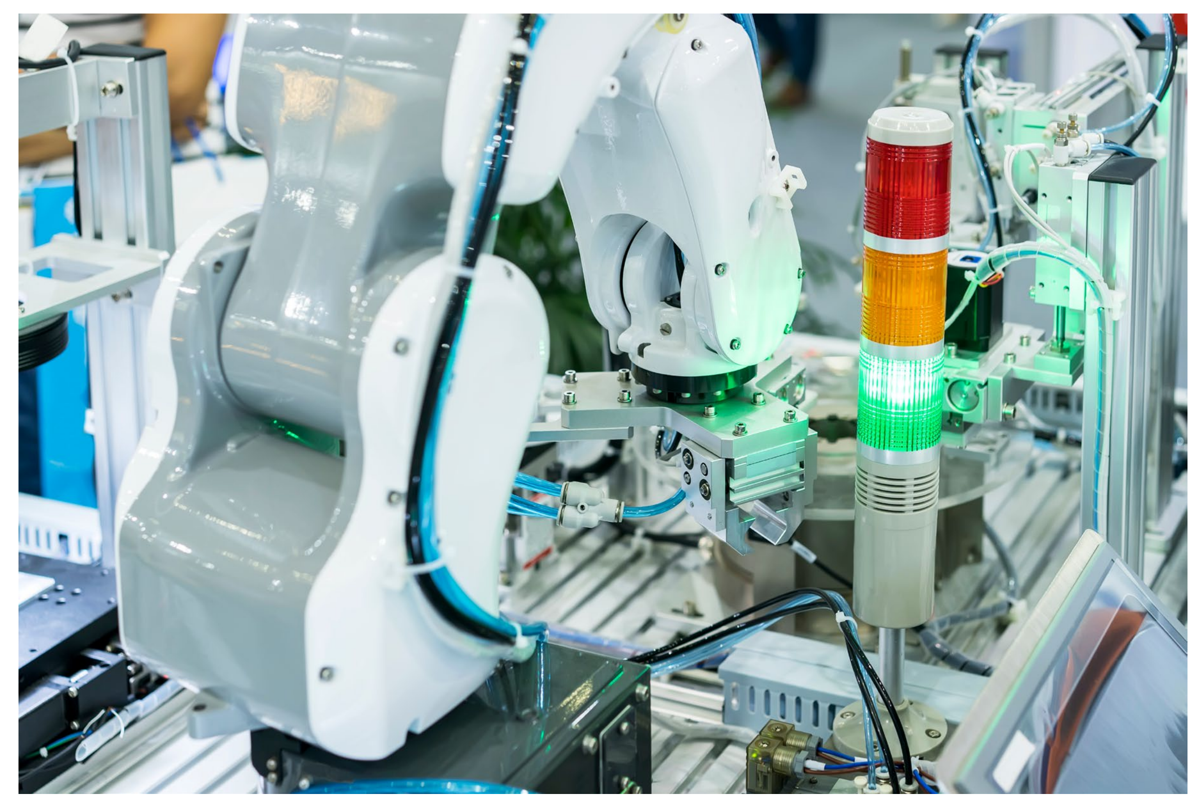
- The tool on the robot didn’t reach the part, or wasn’t aligned correctly
- The robot was not mastered correctly so it’s frame of reference doesn’t match the simulation
- Imperfections in the work cell, such as a slight jog in the 7th axis rail where the simulation expects it to be perfect
- A part positioner has a slight variation to what is in the simulated environment
- Imperfections in the simulation, such as the parts not exactly matching the CAD model or improper measurement of workcell dimensions
- The robot itself drifts over time, that while continuing to move precisely, it’s accuracy in relation to the environment has decreased

5 *Programming tasks that are more quickly done in the teach pendant than in the OLRP application*

OLRP tools are well suited for applications that have 100s or 1000s of points such as complex paths needed for welding or dispensing. These types of applications justify the upfront investment in fully simulating the workcell in the software and the planning of the application using CAD models. However, many tasks do not require such complex movements. These tasks can often be programmed more quickly via the teach pendant. Machine tending, pick and place, straight line welding or dispensing, are all examples of tasks that are most often done on the teach pendant rather than in simulation. These operations also have to be done quickly, especially in high-mix, low-volume environments to reduce the impact of changeover time.

6 *Prompting the operator for an input*

Especially when starting a new task, the operator may want to pause the program to check the parts or to inspect other areas of the application. These interventions are not programmed in the OLRP software, they require blocks to be programmed and added into the program code from the teach pendant. When the application is running, the operator will also use the teach pendant to interact with the task.



7 Peripheral integration

Work cells consist of many components such as laser scanners, stack lights, and automated work holding devices. These peripherals are difficult to model in most tools in a robot independent manner. The result is that once the program is loaded on the robot, subroutines are written that coordinate the robot movements with other activities in the workcell.

8 Coding of common subroutines

It is not practical to program all aspects of the program in the OLRP. For example, the material handling steps before and after the complex path are often more easily programmed in the teach pendant than in simulation. Many shops even have common subroutines, such as for part drop-off, they have developed that are most easily maintained in the teach pendant.

9 *Adding of control logic*

So now you've created and tested your program, and you are ready to run parts. However, you have 10s of parts to run! In most cases the OLRP applications do not have control logic constructs in their software. As such, the operator still needs to add in the control logic that enables the application to be run unattended, often in coordination with other robots or equipment in the work cell.

10 *Executing 3rd party or built in applications*

For applications such as palletizing, the task is configured, and controlled, from the teach pendant. There is a high degree of interaction between configuring the application and executing it. By being able to configure and instantly execute the task without having to transfer programs from another device, the operator is able to quickly and efficiently implement the task using the teach pendant.

Conclusion

At READY we are making the robot ecosystem stronger by removing the fragmentation that exists so that the focus can be on more innovative ways to control and use robots. Our Task Canvas software running on a touch screen teach pendant removes the barrier of needing an operator skilled in a particular robot brand. By enabling more people to use the robots we are increasing the number of robots that can be deployed by eliminating the bottlenecks from skilled labor that exist today.

Contact

Contact
Ben Gibbs, CEO and Co-Founder
ben@ready-robotics.com

Author
Luke Tuttle, Chief Operating Officer
luke.tuttle@ready-robotics.com\

Thought Leadership Team
Kel Guerin, PhD in Robotics, CTO and Co-Founder
Josh Davis, PhD in Robotics, VP of Robotics
Jake Huckaby, PhD in Robotics, VP of Strategic Partnerships

Production Team
Erik Bjornard, VP of Marketing
Kirk Higgins, Product Marketing Manager

For more resources and thought leading content and case studies on automation:
www.ready-robotics.com/resources

For press inquiries, contact
press@ready-robotics.com

Linkedn
Facebook
Twitter
Instagram
YouTube

READY ROBOTICS

1080 STEELWOOD RD.
COLUMBUS, OH 43212
(833) 732-3967