



Industrial Robots: The new era of industrial automation





Robots and industries

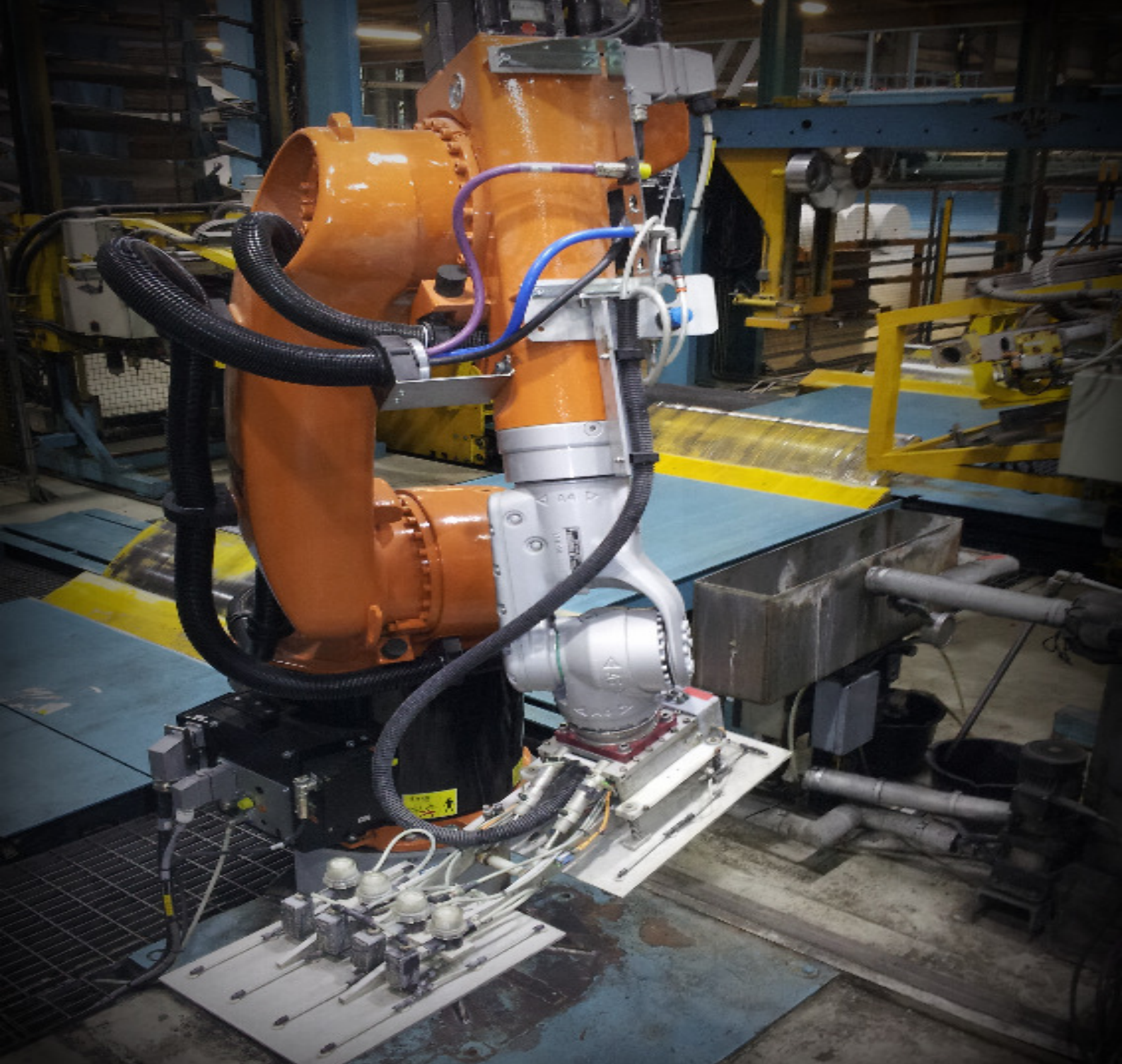
The 1940s witnessed the dawning of a modern day industrial revolution with the introduction of process automation. More recently, it could be argued that another new industrial revolution has begun of even greater significance; the widespread use of industrial robots in the world of manufacturing is now upon us.

Each new revolutionary epoch is defined by one fundamental benefit – increased productivity - we produce more goods more efficiently and of higher quality than at any other previous time.

Initially, robots were conceived in the imaginations of science fiction writers, and have since established a dominant presence with wide practical application in many spheres of the industrial world and are now employed in the most demanding production environments such as:

- Automotive;
- Electronics;
- Rubber and Plastics
- Food and Beverage;
- Metals

This report aims to summarize how industrial robots are designed, programmed and applied in production settings. In the second part of the report Stephen Carradice, design engineer at Optima Control Solutions, discusses some practical considerations involved in the application of these powerful robots. He explores how the robot controls are integrated with the control system of a host production plant and what the technical challenges are in doing so.



What is an industrial robot?

According to the International Federation of Robotics, industrial robots are defined by the standard ISO 8373.

An industrial robot is “an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications.” Industrial robots are:

- **Reprogrammable:** their motions or auxiliary functions are changed without physical alterations;
- **Multipurpose:** adaptable to a different application with physical alterations;
- **Axis-directed:** axes are used to specify the robot motion in a linear or rotary mode.

Industrial Robots – types and characteristics

The International Organization for Standardization (ISO) has categorised a number of industrial robot types with each category being characterised by specific features. (*Please, refer to table 1*). These typically include:

- **Number of axes of motion** – smaller robots have three axes but for larger applications the number of axes reaches six or more
- **Kinematic structure** – the design method for the robotic arm (for example, revolving joints)
- **Work envelope** – the area of movement of the robot
- **Work range** – the vertical area of extension of the robot
- **Maximum payload** – the maximum weight the robot can carry
- **Maximum speed**
- **Accuracy** – coefficient of errors (for example, in mm.)

Robot definitions per ISO 8373:1994 <i>Source*: Cambridge University press; Robotics for Electronics Manufacturing: Principles and Applications in Cleanroom Automation Karl Mathia</i>	<u>Fixed sequence manipulator</u> : manipulator which performs each step of a given operation according to a predetermined motion operation pattern which cannot be changed without physical alteration.
	<u>Sequenced robot</u> : robot having a control system in which the state of machine movements occurs axis by axis in a desired order, the completion of one movement initiating the next.
	<u>Playback robot</u> : robot that can repeat a task program which is entered through teach programming.
	<u>Adaptive robot</u> : robot having sensory control, adaptive control, or learning control functions.
	<u>Sensory control</u> : control scheme whereby the robot motion or force is adjusted in accordance with outputs of external sensors.
	<u>Adaptive control</u> : control scheme whereby the control system parameters are adjusted from conditions detected during the process.
	<u>Learning control</u> : control scheme whereby the experience obtained during previous cycles is automatically used to change control parameters and/or algorithms.

Table 1. Industrial robot types as defined by ISO 8373



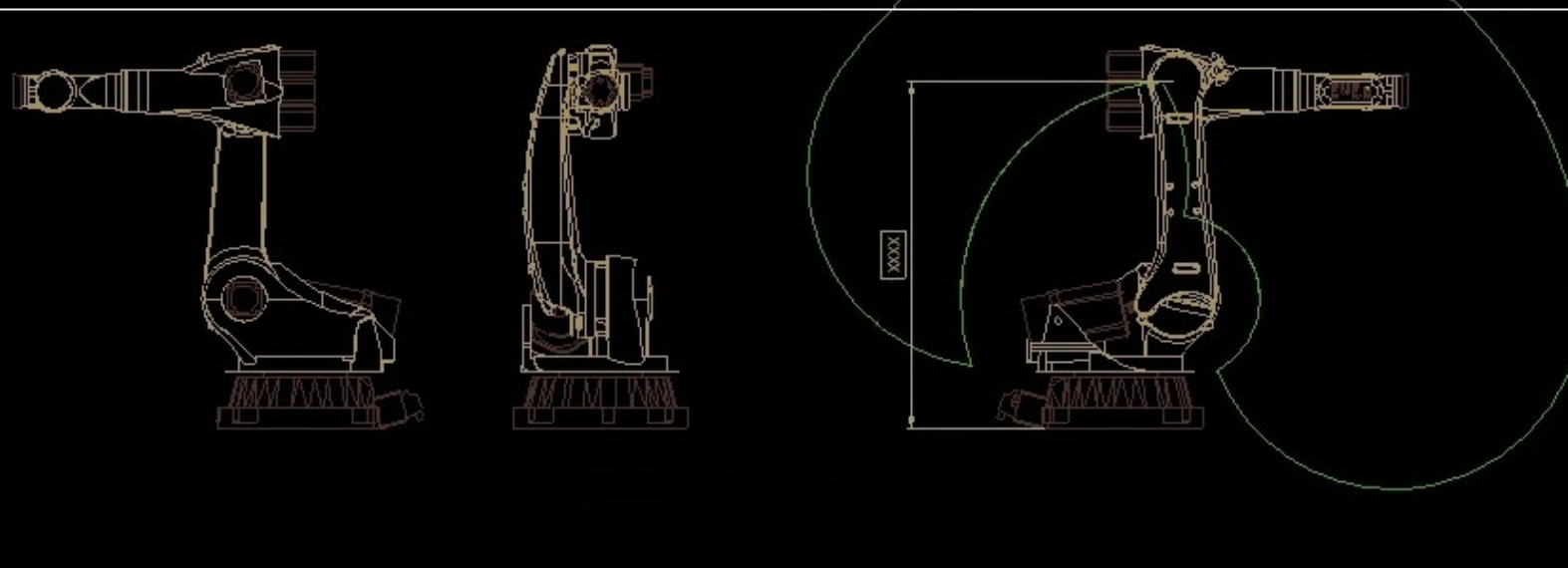
Robots are Great – Motion Carried!

Articulated robots need servo control systems to provide what is in effect the robot's muscles!

So, "What is servo control?" Put simply, if you imagine a small simple DC motor (like the one from a physics lesson at school). If you excite the motor, or connect it to a battery, it will spin. If you connect two batteries (to excite it with a higher voltage), it will spin faster. Now imagine you tell the motor to turn precisely 180 degrees (1/2 revolution) and hold that position no matter how many batteries there are, then that point-to-point movement is typical of a servo control application.

Central to the task of servo control is the concept of closed loop feedback. A closed loop control system relies on a feedback signal for the controlling element to regulate the system. Consider what happens when you are driving your car. As you accelerate towards a certain speed, say 30mph, as the driver (or the intelligent controller of the vehicle) you read the speedometer and regulate the engine power (using the accelerator pedal) to achieve and then maintain the desired 30mph speed. Without the speedometer feedback medium the controller would not be able to accurately control the vehicle.

Position feedback in a servo control system works in a similar fashion. The servo controller having a desired position to which the motor must move to, and using a corresponding position feedback sensor that tells the motor what it's actual position is. In a well tuned control system, the motor will get to exactly the right position and then will turn no more until it is commanded to a new desired position.



Programming and installation of industrial robots

Robots are usually programmed by application engineers using brand specific tools, developed and supplied by the robot manufacturer. These software tools run on laptop or desktop PCs. They often have two types of programming modes, online (or a 'teaching' mode) and offline.

Once the robot programme is successfully written and operational, the programming computer can be disconnected and the robot runs independently. It is an option to either retain permanent connectivity or to re-connect for diagnostic or supervisory use.

Sensor-controlled robots (**table 1**) are widely used because of their capacity to integrate with external equipment such as sensors and cameras to respond to changes in the production environment.

Programming Modes - Online programming

The online programming mode for a robot in effect means 'teaching' it to execute tasks on the actual application (i.e. using manual controls to position the robot to the desired position) then selecting a "Teach" function in order to programme the controller with that target location. Programmers often use a pendant control station so that they can get close to the robot head.

This method records movement and required inputs and saves them in the robot's control system memory. Once operational, movement to preset positions the task is automatically performed when the robot is instructed to by the system.

Offline (or remote) programming

The technique of offline programming does not require the engineer to be physically present, as when 'teaching' the robot. This type of programming more resembles high-level programming and uses similar languages and is often pre-written.

An alternative offline programming method is CAD programming – this is an intelligent programming function that uses interpretation of CAD drawing statistics and simulation techniques (**see picture above**).

There are pros and cons with all methods, but in brief, offline programming is plainly more challenging to develop and commission initially because it is remote programming. However, once successfully completed, the robot programmes can be easily re-issued for subsequent identical systems (OEM type applications).

Online programming offers better commissioning flexibility and accommodates varying application criteria, i.e. mechanical dimension repeatability.



Stephen Carradice

Stephen Carradice has been a design engineer at Optima Control Solutions Ltd since 2006 and has been involved in numerous projects with values in excess of £300,000. He recently, engineered a project in conjunction with KUKA Robots Ltd. The project involved installing and integrating a Kuka robot, specified by them, at a large Flintshire-based paper mill. Here, Steve talks in more detail about the process of integrating the robot's control system with the rest of the production line's controls.

[Steve, tell me more about the project and why you got involved in this project?]

In this project the specification was to integrate a robot at the end of the whole production line to put labels on paper rolls that have been processed upstream. We needed to integrate the robot with the existing control system in order to define the position and movement needed for executing the labelling task.

We engineered the system so that the robot received data programmed from the existing PLC defining the size of the batch coming down the line. The robot then used this data to co-ordinate and adjust its position accordingly.

Generally, when a company purchases a robot like this, it buys a contained unit that has been pre-tested and ready to operate but it has no application specific functionality. The engineering challenge is to integrate the robot within a functioning production line and interface it with various parts of this line. This can be a difficult task that must be done with skill and care; control systems integrators have a critical role to play here. Obviously there are simpler, stand-alone area cases, where integration is not necessary where the robot is programmed in isolation. However, often in established production plants, it is more likely that integration will be required. On this project, my main responsibility was ensuring the robot responded correctly to varying production demands.

[How did you achieve this integration and what was the biggest challenge?]

The biggest challenge was making sure that the robot interfaced well with other existing systems such as PLCs. Because the robot operates on a PC platform and PLCs have their own, different programming language, making them communicate with each other is difficult. If you refer to **Figure 1** below, you will easily spot the difference in the programming languages used by both control systems. I selected a ProfiBus communication system to integrate the main control machine control system with the robot controls.

```

&ACCESS RV01
&REL 24
&COMMENT Pick Labels
&PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe
&PARAM EDITMASK = *
DEF Pick_Labels( )
*****
* CUSTOMER : ---- *
* PROGRAM : PICK LABELS *
* DATE : FEB 2012 *
* BY : OPTIMA *
*****
;FOLD INI
;FOLD BASISTECH INI ----
GLOBAL INTERRUPT
INTERRUPT ON 3
BAS (#INITMOV,0 )
;ENDFOLD (BASISTECH INI)
;FOLD SPOTTECH INI
USERSPOT(#INIT)
;ENDFOLD (SPOTTECH INI)
;FOLD GRIPPERTECH INI
USER_GRP(0,DUMMY,DUMMY,GDEFAULT)
;ENDFOLD (GRIPPERTECH INI)
;FOLD USER INI
;Make your modifications here

;ENDFOLD (USER INI)
;ENDFOLD (INI)

```

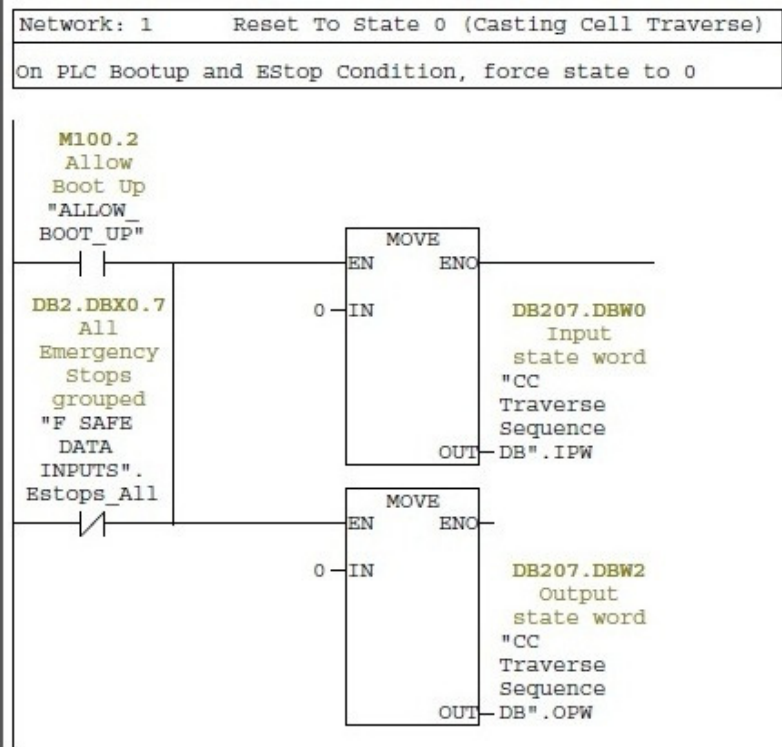


Figure 1. Differences in programming languages: Robot control software (left) and PLC-based ladder logic (right)

AC Servo Drives – servo motor control employed at the robot's axes

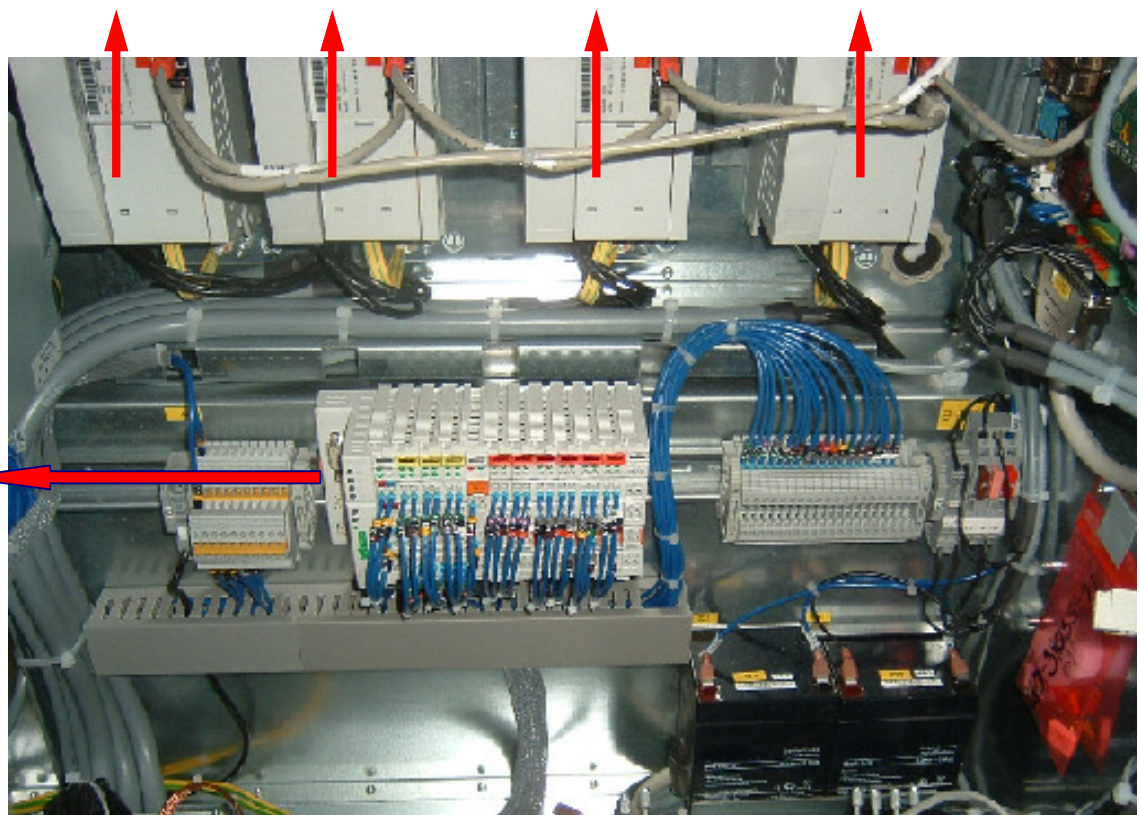


Figure 2. Robot control cubicle



Figure 3. Teach pendant for the robot system (Source: KUKA Robots)

Acquiring process signals from distributed sensors and instrumentation was another requirement of the robot's integration. We specified the various sensors etc, and selected remote I/O unit that was installed at different sections of the production line to meet this demand.

Figure 2 shows part of the robot's self-contained control cubicle and how the different inputs and outputs are integrated.

[How was the robot programmed?]

We used an online, point-to-point teaching technique, Kuka provide a teach pendant that we used to programme the robot. In effect, we showed the robot how to execute a task and it 'remembered' how to do it next time. **Figure 3** shows the main control panel that you use to program the robot.

Programming takes a good degree of familiarity but to simply explain, each actuator (numbered) has a different function e.g. No. 1 is used for automatic test. It is mostly used to teach the robot a position. Numbers 2 to 4 show other switches, a familiar function to all is no. 4, the emergency stop button.

The first step in the engineering exercise is to create a program file; Then secondly to begin teaching the robot the various positions needed to execute its task. As I mentioned earlier, the programmer works the robot through its task manually then using the teach command, it 'remembers' the inputs statuses and how to act on these inputs once automated.

The various other keys and switches are used to move (8) and control the robot when it is switched to manual control.

[What can you tell me about the program interface?]

The programming tool runs on a PC platform. Kuka use MS Windows as their PC operating system which makes it very familiar for most engineers. Because a single robot will often perform a number of tasks on the same production process (at different times) the engineer needs to isolate the various task programmes in his software files (this is a good housekeeping method used as standard by software engineers). In the robot world the main program is called 'Cell', within this cell, the engineer creates and organises individual programs for executing various tasks.

In our project for example, we wrote programmes for different tasks such as 'Pick up', 'Spray' and 'Label'.

A particularly beneficial feature (to operators and engineers) is the Program Status bar, it's located at the bottom of the screen in **figure 4**. Because robots work sequentially, if a required signal is not present and prevents the robot from progressing, this bar describes where the robot is in it's sequence and allows the operator or engineer to quickly diagnose the reason for stalling.

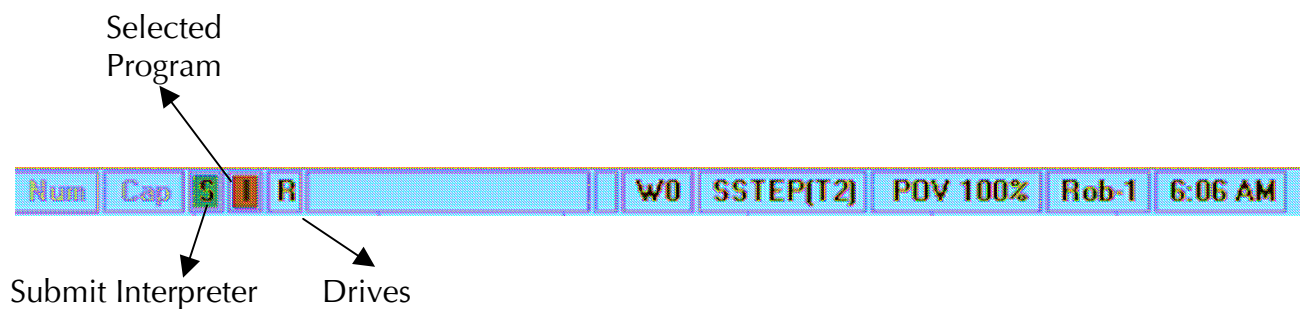


Figure 4. Program Status bar

[Did you use any offline programming?]

Yes, we used our CAD drawings and files initially for test and simulation of the application. By drawing positions we determined pretty well what movements the robot would be required to execute. It was really helpful when writing the final control software later. However, this was not the classical CAD file robot programming method referred to earlier, that is a complex method of remote offline programming where the engineer might never see the robot. It is, as you might imagine, very complex and is rarely used in industrial settings.

[Finally, what do you think are the main benefits of employing industrial robots? What about any disadvantages?]

Robots can flawlessly execute hard tasks such as laser cutting and thermal adhesion. Due to their high speeds, companies achieve better production rates. Improved safety is another advantage – fewer workers get hurt in the manufacturing process. For highly repetitive tasks, robots never get 'tired' or 'bored' – and how could they? One major drawback, however, is that more people are made redundant. The demand for human labour is significantly decreased but that is the only socially negative impact when employing industrial robots.

-- END--

Disclaimer: All data provided in this report is for informational purposes only. Opinions expressed in this report are those of the relevant contributors. Reproduction and citations are allowed after appropriate referencing of the full report. Optima Control Solutions Ltd. reserves the right to prohibit the use of this report by third-party websites and third-party publications.

Images sourced from:

Optima™ gallery – all copyrights reserved.

Design:

Gimp, Picasa

Written by:

Hristina Stefanova

Revised by:

Michael Hill