

Control of Industrial Robots for Meat Processing Applications

Zeng Li, Peter Ring, Kym MacRae and Andrew Hinsch

Food Science Australia, CSIRO

Zeng.Li@csiro.au, Peter.Ring@csiro.au, Kym.MacRae@csiro.au,
Andrew.Hinsch@csiro.au

Abstract

Food Science Australia (FSA) has recently developed several sensor-based robotic applications for the meat industry. All these applications require the robot as the primary tool manipulator to have quick responses, during both on-line operation and off-line communication. Unfortunately commercial robots have not been designed with the operational response common among other types of “real time control” equipment. Industrial robots have historically only had guaranteed performance when operated with off-line, controller resident programs. However the dynamic response of industrial robots is a critical factor when determining the feasibility of their use in the meat industry. This paper describes an investigation carried out by FSA to assess the dynamic response and positioning accuracy which a sensor-integrated commercial robot can achieve with remote program control, i.e. the primary control is implemented via a remote computer separate from the robot controller.

1 Introduction

In the past two decades while robots have been widely adapted into many of the labour-intensive industries, like manufacturing, the food industry has had minimal uptake in the use of commercial robots – particularly in meat. Apart from some traditional applications of robots (such as picking and placing), it is only recently that market competition, operational costs and safety requirements have seen some advanced robotic systems being applied among food production lines. Examples of these types of developments include the robotic pig scribing system [KUKA Robotics, 2002] and the De-boning processing robot developed by meat researchers in Canada in 2001.

Food Science Australia is currently investigating several robotic systems for meat production applications, in particular a robotic beef carcass splitter, a robotic dedager, and a robotic sheep fat removal system.

The basic components integrated into a robotic application are: an industrial robot and controller, a material handling unit, and a sensing or detecting system. Meat applications also require a sterilising unit and peripheral plant integration componentry. The motivation to use commercial industrial robots rather than purpose built machines is that industrial robots

possess proven positioning performance, sound safety records, operational reliability and have replacement parts and service technicians readily available. Compared with customer-specific designed manipulators, FSA has found that the development of an application that integrates an industrial robot reduces development cycle time for an application significantly. For both industrial applications and research purposes direct integration of commercial robots is more economical,

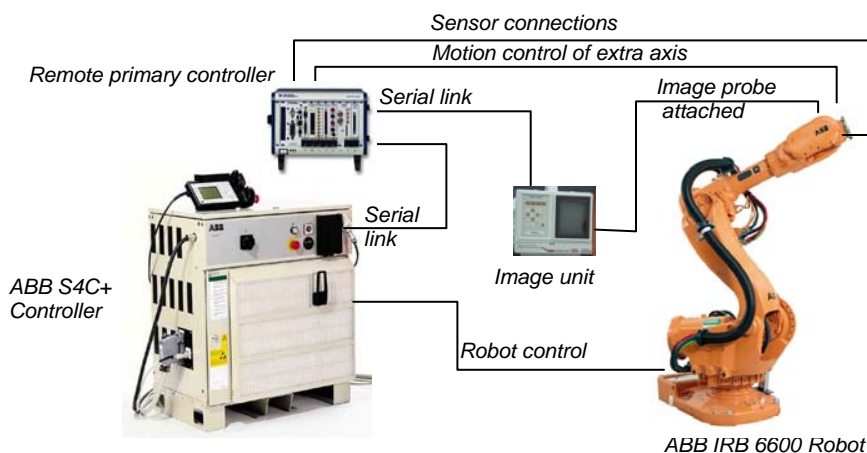


Figure 1 Robotic carcass splitting system

especially as a single robot that can be used in the development of multiple applications, conditional on the system being able to fulfil task requirements.

As the primary tool manipulation component a robot needs to respond quickly under the control of either a controller resident “off-line” programming process or under the direction of a remote primary controller (additional to the robot controller) with access to sensing information, to achieve the quality and efficiency that meat processing requires at production speeds.

The control system of the robotic carcass splitting system is shown as an example in Figure 1. The application includes an IRB 6600 robot, and standard S4C+ controller, an ultrasound image processing unit and a remote primary control unit responsible for the majority of the system processing. In this application the position of the back bone is detected via analysis of images from an off-the-shelf medical ultrasound unit (results shown in Figure 2). After processing by the primary controller, a traverse coordinate is generated that can be used to control the horizontal positioning (relative to the spine) and corresponding rotary angle of a modified carcass splitting band saw. While the robot arm is moving along the spine from the tailbone to the neck of the carcass, the saw position and rotation about its axis are automatically corrected with the ultrasound feedback data. For this example the robot system requires the vertical velocity of the saw along the spine is required to be approximately 75mm/sec at minimum to achieve production speed cycle times. The application also requires the splitting accuracy (cut through the symmetrical centre of the spine and feather bones) to be $\pm 3\text{mm}$, with a previously developed custom-designed laboratory prototype at FSA achieving within $\pm 1\text{mm}$ ([Beare, 2002]). The image acquisition, communication and other control processes need to be fast enough to match the splitting operation requirements. The capability of an industrial robot system to respond quickly during the splitting operation is a critical issue to the success of this application.

Available robotic dynamic performance (response time, communication and positioning accuracy) will dominate the acceptance of industrial robots in the meat industry in automated applications. Limited information can be obtained from manufacturers and robot suppliers, and very little research has been conducted to analyse the functional properties of industrial robot control. Only one paper presented by [Cederberg, 2002] directly deals with these issues. It indicates that for the industrial robots provided by ABB with S4C+ Controllers a 10Hz response via remote program control was possible. This document concentrates on communication capability rather than

the control response and does not investigate the quality of the resultant performance.

To assess robot dynamic performance, an investigation was carried out with a trial system set-up at FSA. This paper describes the details of the experiment and presents the test results obtained through the investigation. This paper also presents conclusions on the technical resolution of commercial industrial robots and their feasibility to be adopted into current industry applications

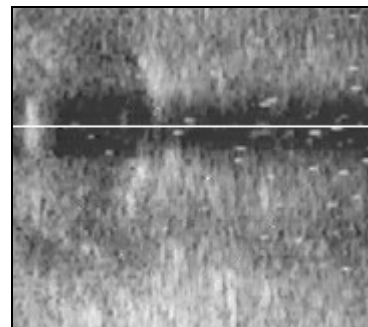


Figure 2 Spine position detected by ultrasound image analysis

2 Experimental system setup

The system established at FSA utilises an ABB IRB 4400 robot (60Kg payload/ 2m reach) for the laboratory experiment. The control of the robot is implemented by a remote computer communicating with the robot controller via ABB’s Robot Application Protocol (RAP) which utilises a Remote Procedure Call (RPC) structure. The computer is connected with the robot S4C+ controller through an Ethernet link.

For this investigation the performance of one coordinate axis of travel (the Y axis) is assessed over a $\pm 50\text{mm}$ movement range. Although 3 dimensional spatial response is important examining 1 direction is indicative of the overall system and provides specific information as to best-case characteristics for real-time correction as would be required for a carcass splitter. Position feedback is provided by a laser sensor attached to the end-effector of the robot, linked to the remote computer. To verify the accuracy and dynamic response, a simple ABB RAPID (controller resident off-line program) routine is loaded on to the S4C+ controller which performs specific move operations. The response to how these operations are activated forms the basis of this paper.

The specifications provided by the manufacturer give the positioning accuracy of a RAPID program, and this is not investigated as part of this trial. This experiment focuses on the performance of the RAPID program compared to remote program control with the position information provided by the sensing system.

3 Implementation of trials

For the purposes of the trial a RAPID program is loaded onto the controller containing all the desired positioning instructions. The movement performance is determined by the destination point, the interpolation method used to reach the destination, the velocity, the zone data and the coordinate system. Normally, the zone data will dominate the positioning accuracy and the smoothness of the robot motion.

This “off line” controller resident RAPID program can be described as a “teach and show” program as the position data for the tool movement is stored in the robot RAPID commands.

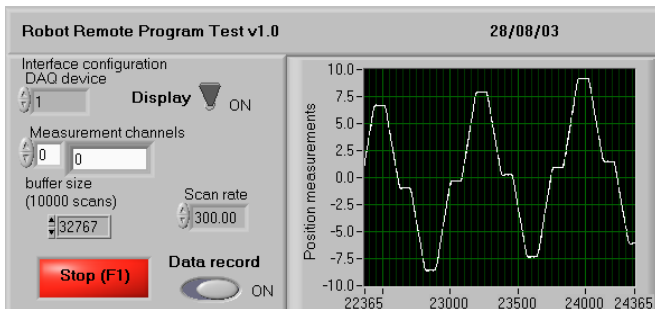


Figure 3 End-effector position measurement

The robot end-effector position is measured with an independent measurement system, consisting of a laser sensor with an accuracy of 30 μm over a +/-100mm range. The system measurement interface is shown in Figure 3.

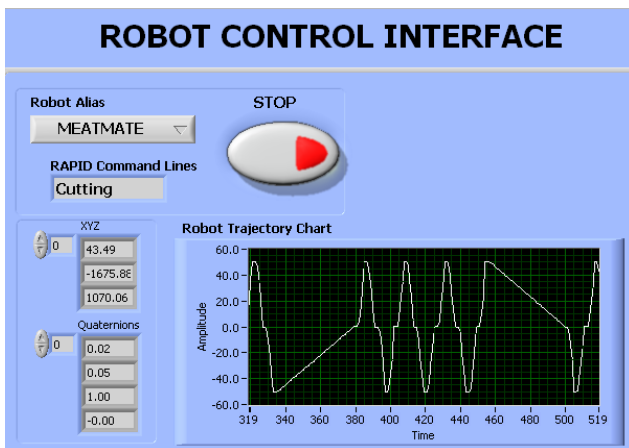


Figure 4 Operation interface on the remote program computer

For the trial comparison a program is also developed on a remote computer that interacts with a base RAPID program that does not contain position information. The remote program (RP) communicates

position data to the base RAPID program and simulates the same motion as the “off line” RAPID program. In the base RAPID program the robot control variables and position data are updated by the RP.

The remote computer is also linked to the laser measurement sensor, to provide positional feedback information to the RP as to when to generate the position coordinates for the robot and update the base RAPID program’s robot position. The interface on the remote computer is shown in Figure 4.

4 Robot performance assessment

The “offline” RAPID program is tested first. Basically, the RAPID program has simple position instruction lines as follows:

```
MoveL Offs(P11,-40,0,-100),v300,
fine,tool0;
```

The above instruction means that the robot is to use a linear interpolation method to move to a new position with coordinate offsets (-40, 0, -100) from position p11. This is to be achieved at a speed of 300 mm/sec. To maximise the quality of the deliverable coordinates, a fine zone data is used which means the robot should achieve a linear accuracy of 0.1mm.

The remote program (RP) control interacts with a RAPID program as shown below. Note that the program executed on the robot now has to receive update position data from the remote computer.

```
PROC cutting()
  MoveL P11,v500,z50,tool0;
  WHILE nZaxis>-1400 DO
    WaitUntil pset<>0;
    MoveL Offs(P11,nXaxis,0,nZaxis),
v300,fine,tool0;
    pnum:=pnum+1;
    pset:=0;
  ENDWHILE
ENDPROC

PROC main()
  string1:="Main_Start";
  pset:=1;
  pnum:=1;
  string1:="Cutting";
  cutting;
  pset:=0;
  string1:="Complete";
  RETURN;
ENDPROC
```

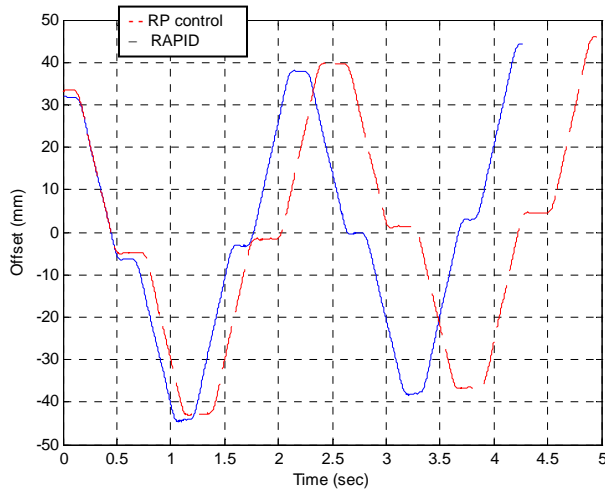


Figure 5 Robot movement completed by RAPID and remote program (RP) control - Linear interpolation and 'fine' zone positioning

With the same movement performance specifications, this time the 'MoveL' instruction has two variables (nXaxis,nZaxis) used for the coordinate position updates from the remote computer.

To make sure the control computer and the robot are performing movement instructions sequentially without missing or skipping single step, a handshaking function is added into the RAPID program on the robot controller. In the program 'pnum' is updated when a movement is completed, while 'pset' is set to ensure the robot can only move when the updated position information is available.

Figure 5 shows the comparison between the movements commanded with RAPID control and those instructed under remote program (RP) control. The position difference is minimal. As a 'fine' zone positioning control has been defined, the robot has very short stop (approximately, 150 ms for the RAPID control and 250 ms with RP control), after a destination position has been reached before the robot moves to the next position. After two cycles there is a time lag of about 800 ms in the RP control response (accumulated from 8 stoppages).

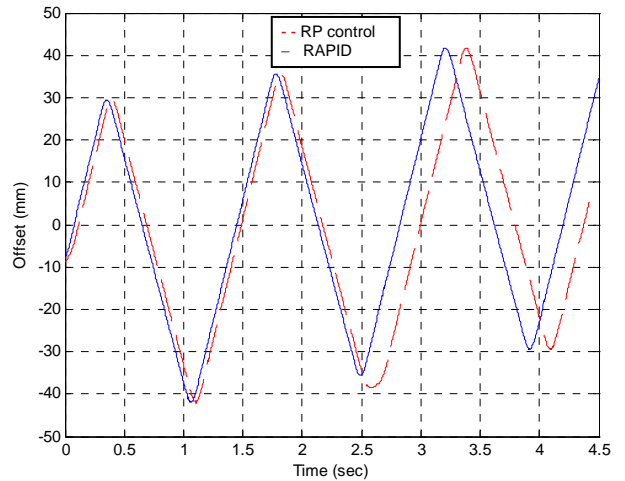


Figure 6 Robot movement completed by RAPID and remote program (RP) control - Linear interpolation and 'z10' zone positioning

In Figure 6 the robot movement instruction is modified to use 'z10' zone data. The destination point of the robot becomes a 'fly-by' point. Once the robot has arrived at location within a 10 mm zone around the destination point, the robot will immediately move to the next point along the path. This type of movement will not have significant time consumed for stop and start motion, but there is a compromise in position accuracy, and a 10 mm distance undershoot can be observed from the ± 40 mm programmed position offset. The time delay after 3 cycles is about 150 ms. This is much shorter than the movement defined with 'fine' zone data shown in Figure 4.

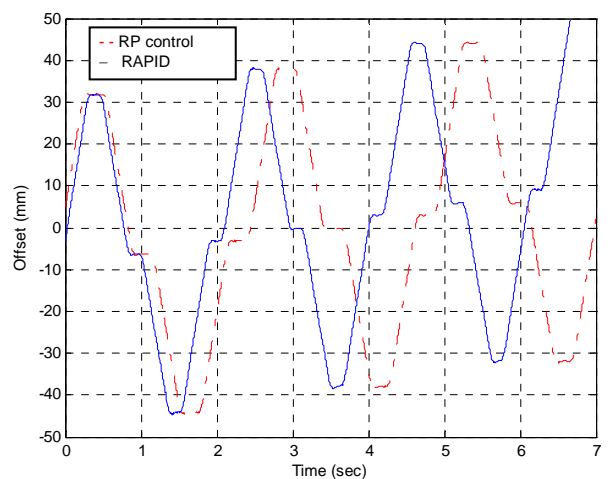


Figure 7 Robot movement completed by RAPID and remote program (RP) control - Joint interpolation and 'fine' zone positioning

Joint interpolation motion is shown in Figure 7 and Figure 8. During joint interpolation the robot is

allowed to move outside a direct linear path between two points as long as the start and end points are in their defined positions. Figure 7 shows an accumulated lag as exhibited in Figure 5 for ‘fine’ zone positioning. Figure 8 shows the time delay is reduced to 50 ms in the first two cycles using ‘z10’ zone data. However, past 2.7 sec, the RP control mode has executed the start and stop movement out of sync with the robot “while loop” cycle that immediately causes significant time delay- by time 3.0 sec there is approximately 300 ms delay between the RAPID and remote program control.

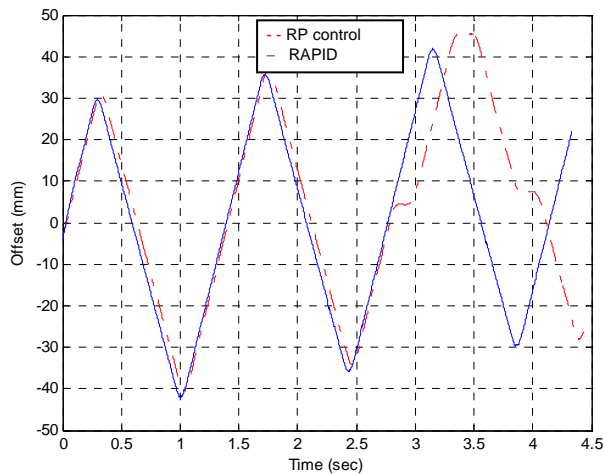


Figure 8 Robot movement completed by RAPID and remote program (RP) control - Joint interpolation and 'z10' zone positioning

5 Control issues of commercial robot control

Based on the test results displayed above, it is shown that commercial industrial robots can achieve good accuracy. However, although the remote program control response has been claimed to reach 10 Hz, and Ethernet communication to be as fast as 100 Mb/sec, the motion response of the industrial robot does not support this specific operation requirement without external data processing.

Interestingly, it has been found that the time consumed for the instruction to be executed comprised of two significant time components-

- The robot takes approx. 0.1 seconds for it to plan the next move. In RAPID “offline” programming the planning process for next move can be carried out in parallel with the execution of the current position instruction.
- The robot delay due to servo lag could end up being as much as 1.0 second depending on the velocity and the actual deceleration performance of the robot, and whether the

robot is close to a robot singularity configuration.

Similar phenomena have been found with KUKA robots. Therefore, adapting an industrial robot directly to an application - if real time or quick response operation is required - is extremely challenging. However, if sensor information can be processed prior to the execution of operation instructions, then the robot can be commanded with off-line programs, which allow pre-movement robot calculation and path planning. If an application allows position information to be remotely acquired and processed before robot movement is required the resultant path is both faster and more accurate. If materials handling can be guaranteed, image information can be acquired and processed before the positioning information is downloaded to the robot. This is applicable in some traditional meat applications. The robots are ultimately operated in a conventional “teach and show” method as they are run with off-line program, with the flexibility to respond to an undefined environment. In this case, the robot’s movement positions are known and they are pre-loaded into the robot controller before movement, although this process may not happen concurrently.

6 Conclusions

Based on the assessment of the performance of an industrial robot, it has been found that if sensor information can be processed prior to the execution of operational instructions, the robot can be commanded with off-line program (RAPID for ABB robots), and a good positioning accuracy can be obtained.

Due to the inherent design of commercial robot programming and control, industrial robots have limited capabilities when real time operation is required. The deterministic 10Hz response frequency claimed by the manufacturers seems unlikely when real time feedback and analysis is required. Depending on program structure there can be a significant time delay in motion response or a lack in tool positioning as may be required for a given operation. Care should be taken prior to design to identify if operational tasks can be fulfilled with off-line programming or that control processing allows enough time for robots to respond. Program optimisation is very important (i.e. minimum number of variables and limited communication packet size desired).

There is a lot of potential to incorporate industrial robots in the meat industry, particularly as the productivity and reduction of cost becomes more essential to the industry. To take advantages of the benefits of robot integration like low cost, quick development and system high durability industrial

robots must be adapted to include features like food-grade housings and improved response time while maintaining currently available accuracy.

Real time control is required in some meat industry processes. Based on our experience, if real time response is needed, system modifications and programming to minimise correction requirements so only one or two joints of robot need fine control has proven successful. This means features available with commercial robots like independent axis or multitasking, can have a big effect on final system design. For example, for IRB robots axis 6 (and 4 on IRB2400/4400) can be configured as an independent axis, which allows to be programmed separately to the rest of the robot.

Development of project solutions for this industry requires application developers and robot manufactures to work closely together to find ways to deliver equipment that meets this challenging environment's requirements.

References

[Cederberg, et al., 2002] Per Cederberg, Magnus Olsson and Gunnar Bolmsjö. Remote control of a standard ABB robot system in real time using the Robot Application Protocol (RAP). In Proceedings of the International Symposium on Robotic, ISR2002. Stockholm, October 2002. IFR, paper No. 113.

[Beare, 2002] Richard Beare, An Ultrasound Imaging System for Control of an Automated Carcass Splitting Proceedings of the 12th Scandinavian Conference on Image Analysis, pages: 375-381,2001

[ABB, 2002], ABB RAPID Reference Manual 3HAC7783-1, For BaseWare OS 4.0.20

[KUKA, 2002], KUKA Robotic Pig Scribing System, Promotion Material, Germany, 2002