A software control system for robotic sample transfer

1. Introduction

Our Linear Accelerator endstation has been equipped with an inverted fluorescence microscope for analysis of the irradiated sample and a two-axis robot for fast transfer of the sample between the irradiation point and the microscope. A working system requires flexible software control to co-ordinate the three sub-systems (accelerator, robot, microscope) which normally operate as stand-alone instruments.

The fundamental steps in an irradiation experiment are broken down into:

- Acquire the sample (using the robot)
- Examine the sample pre-irradiation (using the microscope)
- Irradiate the sample (using the accelerator)
- Examine the sample post-irradiation (using the microscope)
- Store the sample (using the robot)

At first glance these steps seem straightforward and easy enough to implement once an inter-process communication method has been established between the separate control software tools which reside on different computers. However, it soon becomes clear that care must be taken to maintain a safe, interlocked and accurate working system. Complications are introduced when dealing with multiple samples, and there are inherent safety (and equipment damage) concerns when trying to co-ordinate a radiation producing device, a mechanical machine with fast and powerful moving parts, and a microscope which contains expensive and delicate optical components. Add to that the fact that the operator is in a different room to the equipment and thus cannot directly see what is going on (only so much can be imaged with CCTV cameras) during the experiment and the need for reliability is obvious.

A decision was made to use 'Named Pipes' as our inter-process communication method. This was because the Windows C commands seemed simple enough to learn and use straight away, and because the method itself is well established (e.g. see http://en.wikipedia.org/wiki/Named_pipe). An overview of the system is shown in Figure 1

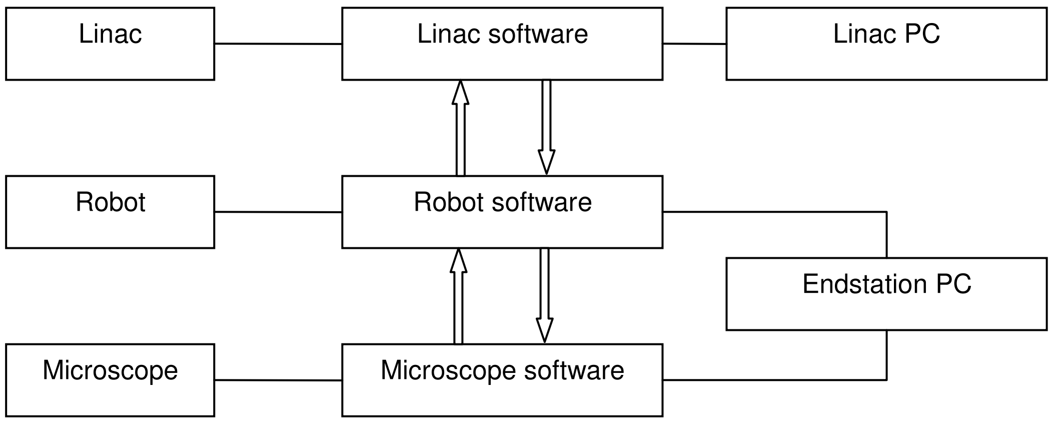


Figure 1: System overview. The three hardware systems (linac, robot and microscope) are controlled by separate software systems residing on two PCs. For simplicity remote view computers and control screens are not presented here

2. The system hardware.

We use two coupled linear stages (Aerotech type PRO225-1200 and type ATS100-200), allowing both horizontal (x axis) and vertical (y axis) motion (Figure 2) required to transfer the cell dish between imaging and irradiation areas. The x-stage allows a travel of 1200 mm, a repeatability of $\pm$ 2.5 µm and allows accelerations of up to 1000 mm/s^2. The y-stage has a travel of 200 mm, sub-micron repeatability and works at accelerations of up to 500 mm/s^2. In order to pick up each sample we use an electromagnet (SM series, Solentec, Ltd.) which magnetises a soft iron disc attached to each sample holder. When the dish is to be released, the disc is momentarily de-magnetised. This system is described separately in the note "A magnetic pick-up and release system for moving lightweight samples".

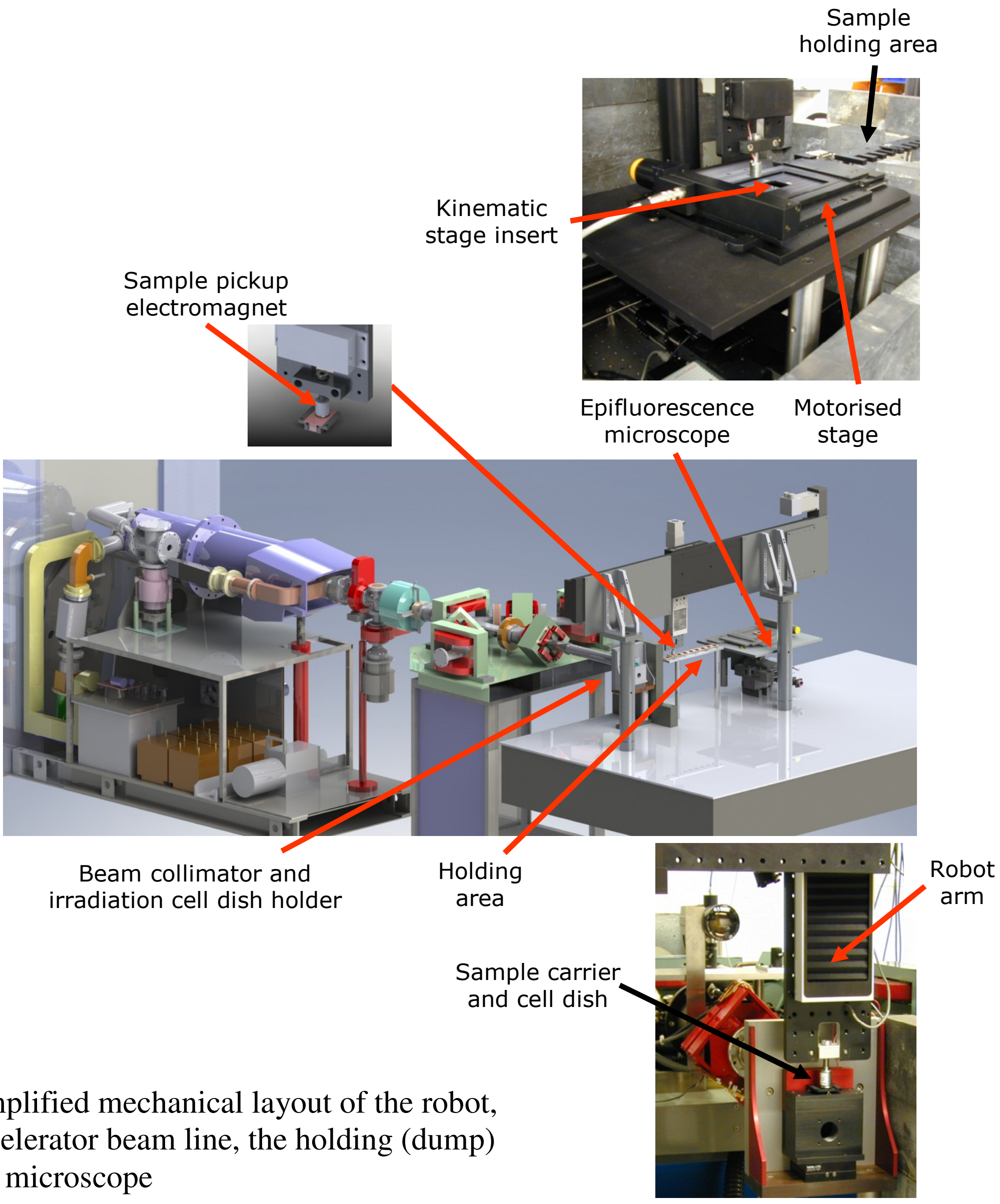

Sample holding area
Kinematic stage insert
Sample pickup electromagnet
Epifluorescence microscope
Motorised stage
Beam collimator and irradiation cell dish holder
Holding area
Robot arm
Sample carrier and cell dish

Figure 2: Simplified mechanical layout of the robot, the linear accelerator beam line, the holding (dump) areas and the microscope

3. Logical sequence in moving a sample

Any sample can be in the following positions:

1. On the microscope XY stage
2. At any one of 9 'rest' or 'dump' positions
3. At the irradiation point

To start an experimental sequence we have a maximum of 9 samples which have to undergo potentially different treatments. The sequence of events outlined in the introduction can be broken down more generally to illustrate the checks needed.

- Pick up the sample:
    - What is the status of the sample?
    - If the sample is ‘resting’, assess the next step.
    - Does it need to be irradiated? If so, is there already a sample at the irradiation position? If so, leave this sample for now, and assess the next.
    - Does it need to be imaged? If so, is there already a sample on the microscope? If so, leave this sample for now, and assess the next.
    - Does it need to be dumped? If so, find a free dump position to set as the drop-off position.
    - If we are moving a sample from the microscope, we must first drive the microscope stage to a sample pick position which aligns the cell dish holder kinematic mounts. Having sent this command, we then need to wait for a response from the microscope to inform that this position has been reached by the XY stage. If a timeout period elapses and no response has been received, an on-screen alert appears, and the next sample is assessed.
    - If we are moving a sample to the microscope stage, we must do the same. Prepare the microscope stage to be in the correct position to accept the sample by sending a command and wait for a response. If this is unsuccessful, alert the operator and assess the next sample.
    - Move the robot to the pick-up position of this sample.
    - Pick up the sample.

- Move the sample:
    - Move the robot to the drop-off position.
    - Release the sample.

- When irradiating:
    - Tell the Linac to initiate irradiation. At present the irradiation parameters are entered manually by the operator but it is a straightforward step to send these from the control program as well.
    - Wait for the Linac to indicate irradiation is complete.
    - On receipt of this confirmation it is usual that we want to look at it on the microscope, so the above ‘acquire’ and ‘move’ steps are processed again.

- When imaging:
    - With the sample in place on the microscope stage the operator can inspect the sample via the microscope software.
    - On completion the operator has the choice to irradiate or dump the sample. If the sample is dumped there is an option to specify how many minutes it should rest for. Dumping a sample is usually the signal for the program to start looking for the next sample in sequence when requested to by the operator. However, it is possible to pre-define sample sequences so that samples are processed automatically in a loop as and when they become available.

All the above rely on four lines of pipe communication:

1. Robot to microscope:	‘move to position to accept / handover this sample’
2. Microscope to robot:	‘in position’
3. Robot to Linac:	‘sample in position, irradiate’
4. Linac to robot:	‘irradiation complete’

Each pipe is tested every few seconds by sending a test message. If the test is unsuccessful the pipe is reset.

4. Sample transfer

The above describes the sequence of communications but what about the motion of the robot itself (in the section 'Move the sample')? Transporting a sample between two positions has, literally, some barriers to overcome, as shown in Figure 3.

A. The first issue is that a sample must be picked up or dropped off vertically for that portion of the move. If angled moves were allowed there would be a risk of hitting something, typically other samples, close by and off to the side. We therefore specify that the start and finish of any pick up or drop off is vertical with a minimum distance of several centimeters.

B. The next decision to be taken is whether there is a barrier between the present position and the target position. If not, then the solution is simply to move horizontally before descending (long dashed line). If there is a barrier however, such as a lead wall (for radiation shielding purposes), the intention will be to attain a height higher than that of the wall before it is reached. The sample can then pass over the wall, and once the barrier is cleared a descent is allowed. A trivial solution would be to immediately raise to the maximum height, then travel laterally until over the target position, then descend (short dashed line). Not only is this potentially slow it subjects the cells to discontinuously changing accelerations. The method employed is to begin moving laterally when arrival at the near face of the wall would then coincide with the vertical part of the move having just finished (if far enough away from the barrier the X move can begin immediately). In this scenario no time is 'wasted' waiting for the X axis to move after the Y axis has already stopped. It also offers a much smoother and continuous motion. Of course, closer to the barrier the X move must be delayed more and more.

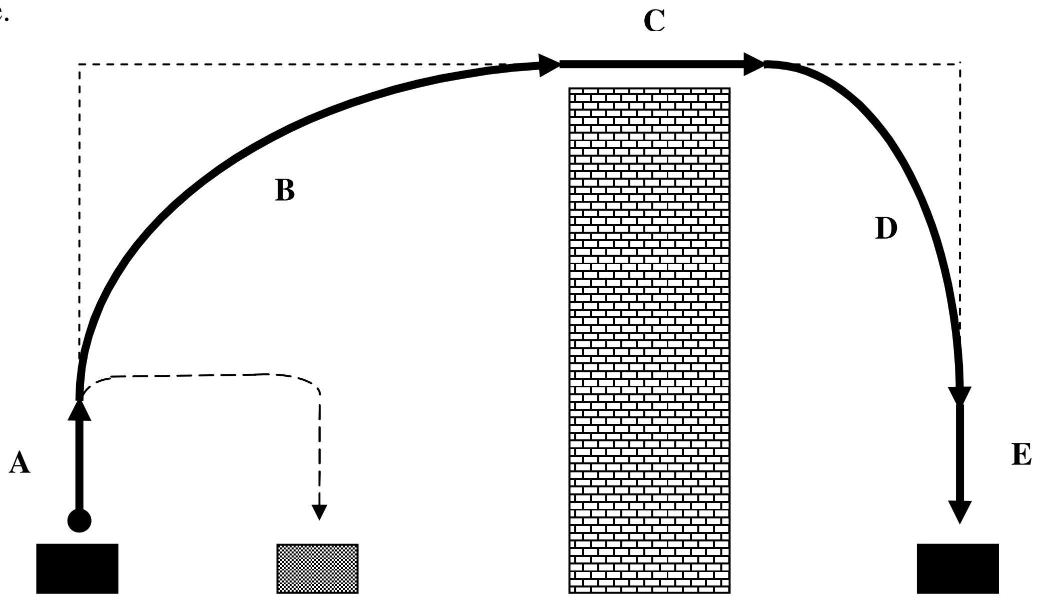


Figure 3: Simplified front view of the robot paths

C. After this, the barrier width is cleared.

D. Then the decision to descend needs to be made. Similarly to before, vertical motion is only recommenced at a time such that will make the X axis stop just when the Y axis is at the minimum clearance height.

E. This leads to the final, vertical only, portion of the move.

The points at which the X and Y axes are engaged are calculated from the knowledge of the co-ordinates of the top of the barrier and each axis speed and acceleration. Care must be taken to allow for the different possible states of both axes when calculating transit times i.e. has the axis had time to reach top speed / will it still be accelerating / will it be decelerating? Each case requires a different formula for determining the distance travelled in a certain time, and this depends on the speed, acceleration, and distance to be covered.

5. Experiment sequence

While the above describes the steps involved in moving a single sample, there is also the issue of how to co-ordinate movement of multiple samples in order to complete an experimental run. The simplest option is for all the moves to be controlled 'manually' by the operator e.g. by pressing (on the robot interface panel) 'get next dish', imaging the dish, pressing 'irradiate', imaging the dish on its automatic return to the microscope, pressing 'dump'.

For multiple samples and well-defined time-points this approach will become laborious and error-prone. What is needed is for the robot control program to have the option for the operator to specify sequences for each sample. Typically this would follow the template of:

- image
- irradiate
- image
- store (dump) for a given length of time
- image
- store (dump) for a (different) given length of time
- etc.

A prototype system has been successfully written and tested. However, the details of the biological experiments to come will dictate how this section develops, and how the software will need to adapt to the users' needs. The software challenges in writing such a control algorithm are in what user-defined sequences are allowed (e.g. 'irradiate' followed by 'irradiate' makes little experimental sense) and in the priority treatment of samples. For example, a sample requiring irradiation followed by imaging then immediate re-irradiation, imaging, and so on will monopolise the robot and leave other samples untouched. Similarly, asking the system to repeatedly image each of 9 samples in, say, 10 second intervals will not be possible, as the moves will take too long.

The solution is either to assume that the operator will plan their experiment intelligently, and allow the control program to make the best of the sequences it is given at run-time, or for the program to run some sort of simulation beforehand and report whether the steps involved are achievable at the correct times.

6. Future developments

As mentioned before, the nature of the biological experiments to come will dictate the shape of the 'sequencer' part of the control program to come and how it develops from its prototype state.

Secondly, the individual sample moves themselves have been optimized for the current physical layout. This may change in the future. There is the possibility of a second 'dump' area beyond a second lead wall, and for environmental control of the samples it would be desirable to fit an enclosure around the sample areas, with access ports. This second option especially would redefine the possible paths the robot could take and require new motion algorithms which maintain a smooth transit.

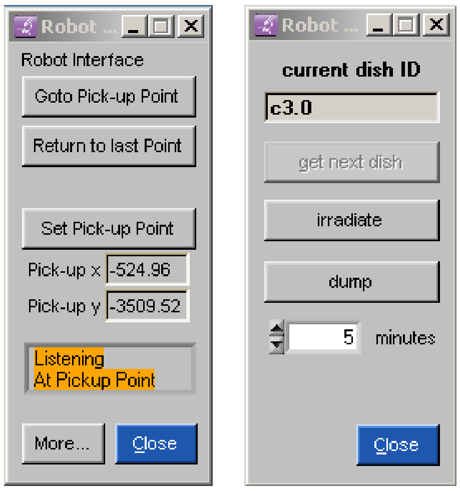
7. Software environment

The robot and linear accelerator software is programmed in 'C' using the National Instruments LabWindows CVI software environment. The linear accelerator program has been written entirely in-house and the robot uses library functions supplied by the system manufacturer, Aerotech, Inc. (http://www.aerotech.com/). The microscope software has been written in-house using Microsoft Visual C++. All programs are C based.

Finally, the fast robot move routine itself is actually written in Aerotech's version of basic, Aerobasic, and resides on the robot controller. The routine is called from the main program when required. The reason for this is that the different sections which comprise the sample transfer move are triggered by robot co-ordinates; these are continuously updated by the controller and the approach adopted minimizes the risk of overshoot. Had the robot co-ordinates been read from the controller into the host, the data transfer delays would have become excessive. The best solution is to allow the controller to take over this part and then inform the main program when the move is complete.

Graphical user interfaces

We describe here some of the Graphical User Interfaces (GUIs) used in this arrangement. The panels shown in Figure 4 set the pick-up co-ordinates of the microscope stage, and allow the user to control the sample sequence.

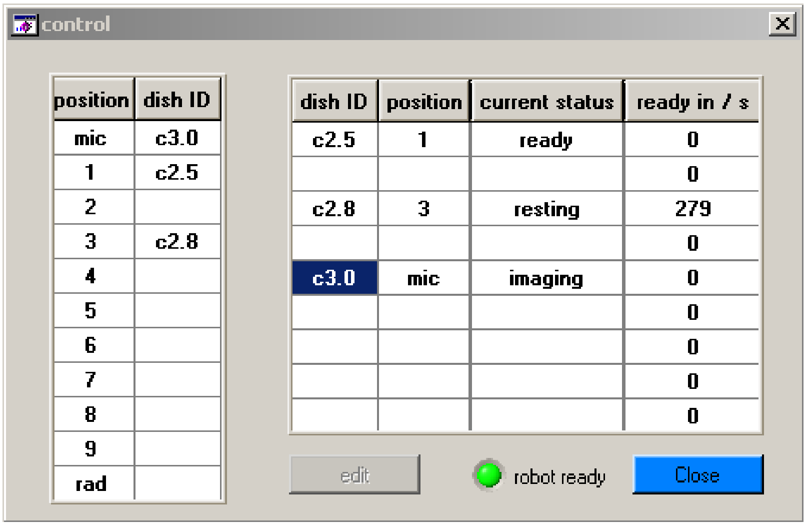


Pick-up x -524.96
Pick-up y -3509.52
Listening
At Pickup Point

current dish ID
c3.0
5 minutes

Figure 4: GUIs used by the microscope software (Left) and the Robot software (Right)

The panel shown in Figure 5 provides information on the status of each dish.

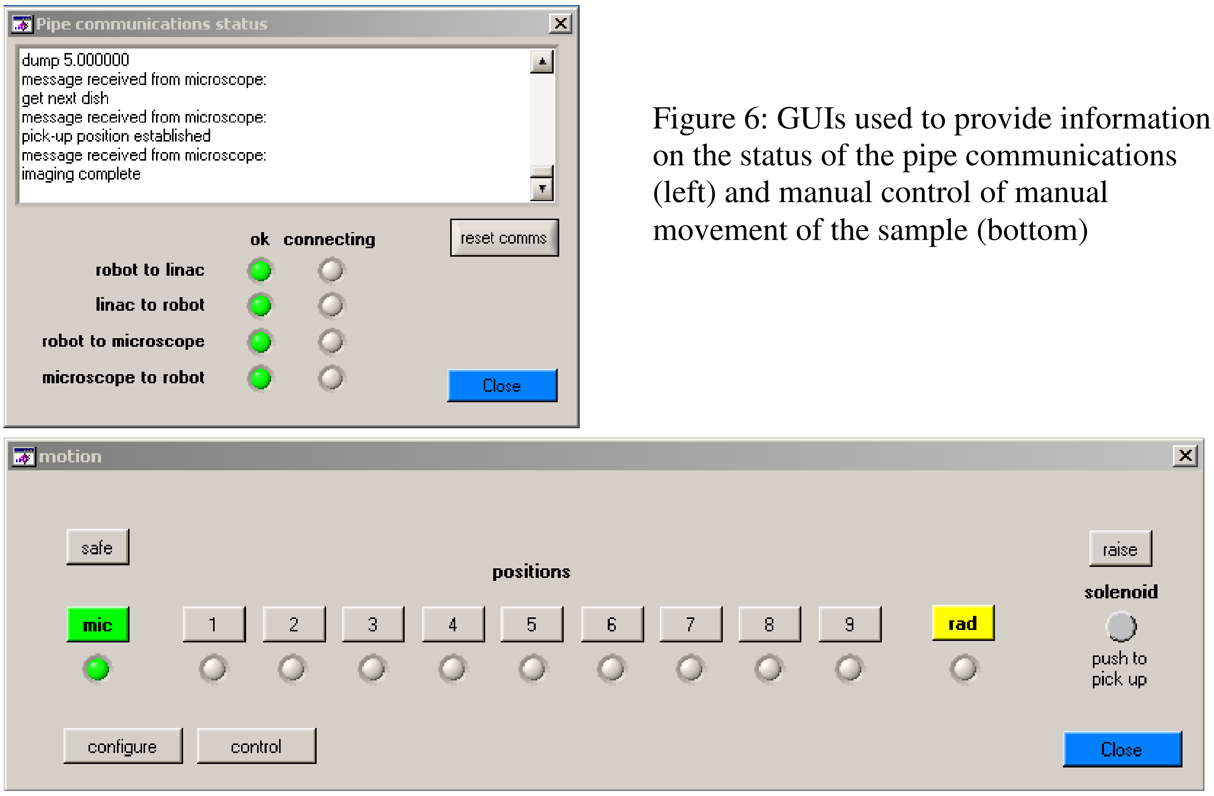


position	dish ID
mic	c3.0
1	c2.5
2	
3	c2.8
4	
5	
6	
7	
8	
9	
rad	

dish ID	position	current status	ready in / s
c2.5	1	ready	0
			0
c2.8	3	resting	279
			0
c3.0	mic	imaging	0
			0
			0
			0
			0

Figure 5: Dish status panel

Figure 6 shows the panels used to provide information on the status of the pipe communications and to allow for more manual control of the robot position and working parameters.



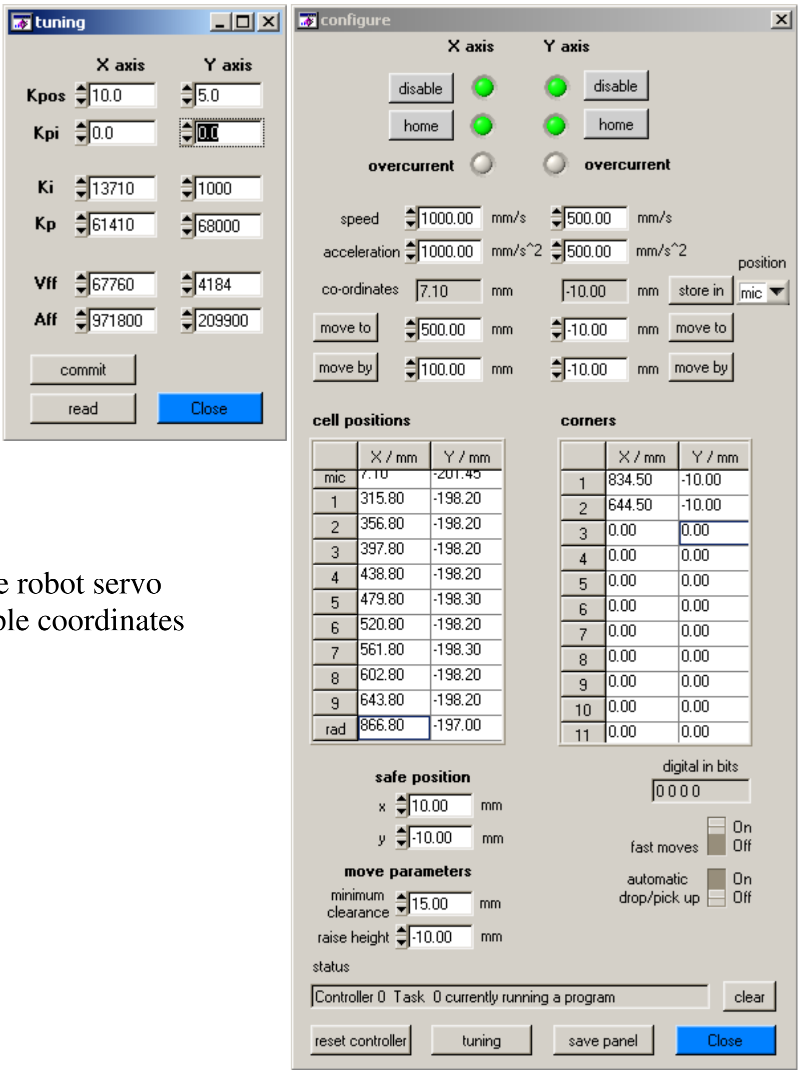
Pipe communications status

```
dump 5.000000
message received from microscope:
get next dish
message received from microscope:
pick-up position established
message received from microscope:
imaging complete
```

	ok	connecting
robot to linac	🟢	⚪
linac to robot	🟢	⚪
robot to microscope	🟢	⚪
microscope to robot	🟢	⚪

Figure 6: GUIs used to provide information on the status of the pipe communications (left) and manual control of manual movement of the sample (bottom)

Figure 7 shows the panels used to provide tuning for the robot axis servos and to define the sample coordinates and provide readout and low level control of the servos.



tuning

	X axis	Y axis
Kpos	10.0	5.0
Kpi	0.0	0.0
Ki	13710	1000
Kp	61410	68000
Vff	67760	4184
Aff	971800	209900

configure

	X axis		Y axis	
speed	1000.00	mm/s	500.00	mm/s
acceleration	1000.00	mm/s^2	500.00	mm/s^2
co-ordinates	7.10	mm	-10.00	mm
move to	500.00	mm	-10.00	mm
move by	100.00	mm	-10.00	mm

position: mic

cell positions

	X / mm	Y / mm
mic	7.10	-201.45
1	315.80	-198.20
2	356.80	-198.20
3	397.80	-198.20
4	438.80	-198.20
5	479.80	-198.30
6	520.80	-198.20
7	561.80	-198.30
8	602.80	-198.20
9	643.80	-198.20
rad	866.80	-197.00

corners

	X / mm	Y / mm
1	834.50	-10.00
2	644.50	-10.00
3	0.00	0.00
4	0.00	0.00
5	0.00	0.00
6	0.00	0.00
7	0.00	0.00
8	0.00	0.00
9	0.00	0.00
10	0.00	0.00
11	0.00	0.00

safe position
x 10.00 mm
y -10.00 mm

digital in bits 0 0 0 0

fast moves: Off
automatic drop/pick up: On

move parameters
minimum clearance 15.00 mm
raise height -10.00 mm

status: Controller 0 Task 0 currently running a program

Figure 7: GUIs used to tune the robot servo drives (left) and to define sample coordinates and other low level functions

This note was prepared by S. Gilchrist and B. Vojnovic in August 2011. PR Barber contributed to software and D. Martins provided some of the diagrams. We thank IDC Tullis and J Prentice for software models of the hardware and for assistance with its installation.

We acknowledge the financial support of Cancer Research UK, the MRC and EPSRC.