VISION BASED CLOSED-LOOP CONTROL SYSTEM FOR SATELLITE RENDEZVOUS WITH MODEL-IN-THE-LOOP VALIDATION AND TESTING

Jonathan Luke du Bois, Peter Newell, Steve Bullock, Peter Thomas and Tom Richardson
University of Bristol; Department of Aerospace Engineering, University of Bristol, Queens Building, University Walk, Bristol. BS8 1TR; +447803012862; thomas.richardson@bristol.ac.uk

Abstract: *A relative motion robotic facility has been developed at the University of Bristol which allows model in the loop testing of Matlab and Simulink based models. Automated rendezvous is highly desirable for space debris collection and satellite on-orbit servicing missions, both of which could require rendezvous with a tumbling object or satellite. For the purposes of this paper, a simple 6 DoF closed loop satellite model was run in real-time with a 1.3m scale satellite model mounted on one robot, tracked from the second. Successful closed loop tracking and rendezvous are demonstrated for selected configurations, using a single real sensor and real-time vision tracking software. Key details of the RMR robotic facility are also given, in particular with reference to the real-time interface implementation.*

Keywords: *Closed-loop, Control, Satellite, Rendezvous, Relative Motion Robotic Facility.*

1. Introduction

This paper describes the development of a model-in-the-loop (MiL) test platform designed specifically for large scale testing of sensors and control systems associated with the relative motion of two distinct bodies in Euclidean space. Dubbed the Relative Motion Robotic (RMR) facility, it is comprised of two 6 degree of freedom (DoF) robotic manipulators, several motion control computers, and a real-time simulation platform. The utility of the RMR is illustrated with respect to an automated satellite rendezvous and docking (RvD) problem.

Orbital space debris has become a major risk for the International Space Station [1]. Many currently operational satellites are reaching the age where they are running out of fuel and power, but their payload is still fully functional [2]. Instead of decommissioning, these satellites could have their design lifetimes extended by on-orbit servicing. Even where this is not feasible or desirable they might have many expensive components that could be worth salvaging [3]. Automated satellite rendezvous methods could facilitate the capture of large pieces of space debris and the docking of satellites at or beyond the end of their lifetime, allowing for the cleanup of orbits, extension of satellite lifetimes, and recovery of disused equipment.

A variety of RvD tracking techniques are described in the literature, employing a range of test methodologies e.g. [4-11]. Qureshi *et al.* in [4] demonstrate a robotic arm that autonomously captures a free flying satellite with a low-level of object recognition . Kelsey *et al.* [7] have used a model-based pose refinement algorithm and two cameras to determine the relative pose of a target satellite with validation completed on a 3DoF test-bed. Mitchell *et al.* [8] considered four systems for vision-based tracking: AutoTRAC is a camera-based system that uses an LED array that pulses in time with the camera shutter; Natural Feature Image Recognition (NFIR) is a camera-based system that does not require retroreflectors or beacons and instead relies on edge detection in conjunction with a known model of the target; Advanced Video Guidance Sensor (AVGS) is a video-based system that utilises lasers to illuminate retro-reflective targets; and Optech LIDAR is a laser-based system that determines range and speed of a reflective surface.

The development of the ESA Automated Transfer Vehicle (ATV) employed full size, real-time MiL tests over a distance of 250m, using a vision-based laser system to illuminate retroreflectors [9]. A similar example to the robotic setup given in this paper is that of the German Space Operations Centre (GSOC) who are testing a vision tracking system and investigating contact dynamics using two KUKA industrial 6DOF robots for the purpose of on-orbit servicing missions [10,11].

The focus of this paper is on the development of control systems for the RMR facility which are suitable for real-time MiL tests, and the demonstration of capability in the evaluation of a vision tracking system for satellite RvD. In Section 2 the RMR facility is described, and the control options are discussed. In Section 3 the chosen satellite control topology is presented. The vision tracking sensors and software are then described in Section 4; the closed loop controller is outlined in Section 5, and the results are presented in Section 6.

2. RMR: Overview

The RMR is intended for model-in-the-loop (MiL) testing, sometimes referred to as hybrid testing, real-time dynamic substructuring, or grouped in with hardware-in-the-loop tests [12]. In these tests a physical subsystem or set of subsystems is coupled to a numerical simulation of a larger system and environmental conditions, allowing complex realistic test conditions to be recreated in the laboratory. An example configuration, as used in this paper, is shown in Figure 2-1. The closed loop system uses position demands from the numerical simulation to move the robot manipulators and recreate the motion of hardware mounted on the end. Force measurements and signals from sensors mounted on the hardware are fed back into the simulation in order to close the loop.

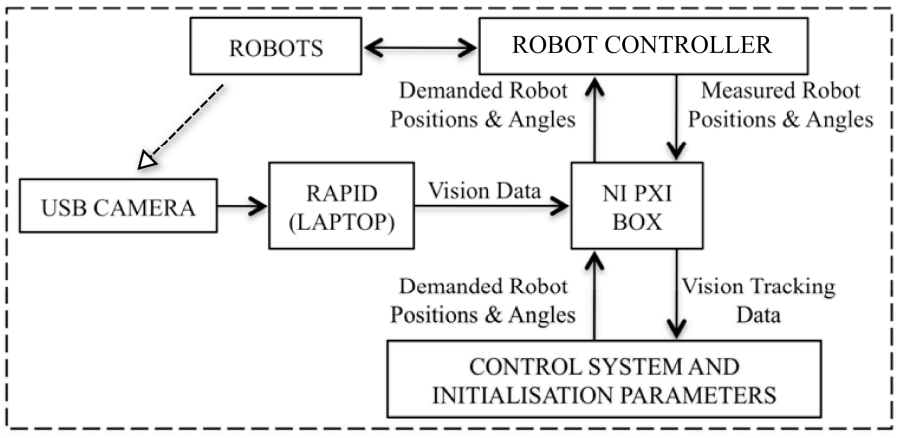


Figure 2-1: System Overview.

2.1. Equipment

Simulation models are compiled from Simulink® [13] and executed in real-time on a National Instruments PXIe-8133RT 1.73 GHz control board mounted in a PXIe-1033 chassis. The PXIe system also runs a supervisory process responsible for orchestrating the test procedure and pre-processing the position demands sent to the robots. This supervisory process also acts as a hub for all the signals associated with the test. The position demands are sent via communications threads on the PXIe to the robot control hardware, described in detail in the next section.

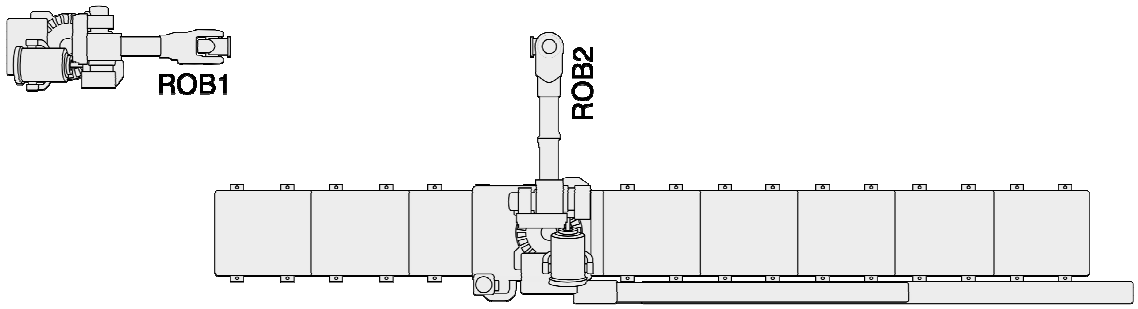


Figure 2-2 Layout of the robotic cell in the RMR facility (plan view)

The robotic cell comprises two ABB IRB6640 6DOF industrial robots [14], designated ROB1 and ROB2, and having the performance characteristics as detailed in Table 1. ROB1 is secured to the ground whilst ROB2 is mounted on a 7.7 m IRBT6004 track to permit translation of the robot base at a rate of 5.2 ft/s (1.6 m/s). The layout is shown in Figure 2-2. The two robots plus track provide a total of thirteen degrees of freedom, and the task of resolving this redundancy into a six degree of freedom relative motion simulation falls to the supervisory process.

Maximum acceleration	~2	g
Maximum relative velocity	~6	m/s
Pose accuracy	0.16	mm
Pose repeatability	0.07	mm
Pose stabilisation time	0.36	s
Track length	7700	mm
Track maximum velocity	1.6	m/s
Track pose repeatability	0.08	mm
reach (ROB1/ROB2)	2.55/3.20	m
Load capacity (ROB1/ROB2)	180/130	kg

Table 1 Performance characteristics for the IRB6640 robots and the IRBT6004 track

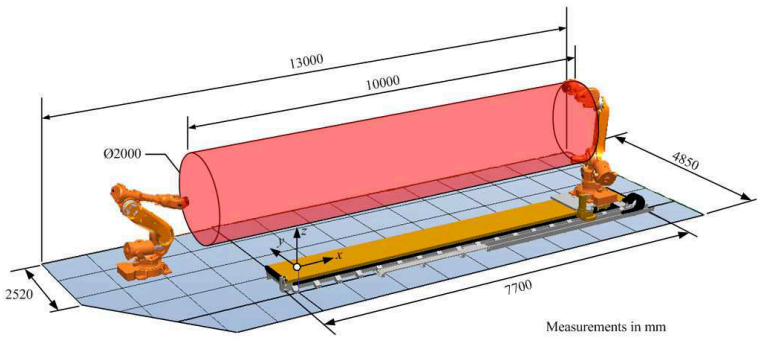


Figure 2-3 Operational envelope for RMR robots

A photograph of the installation can be seen in
Figure **2-4**, with mounted air-to air refuelling hardware as part of a concurrent project [15,16].

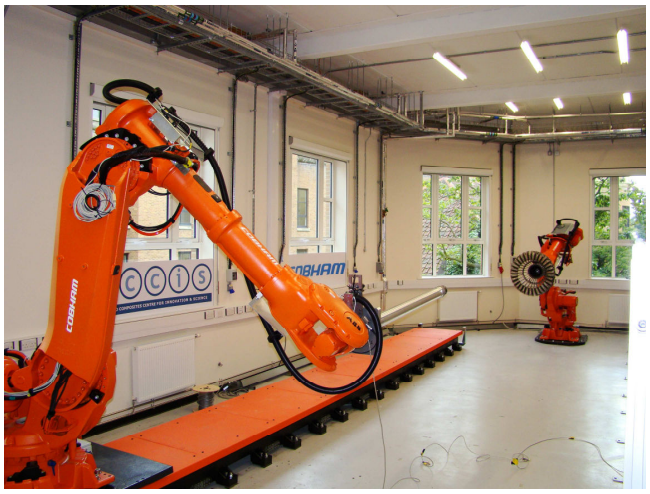


Figure 2-4 Photograph of the RMR robotic cell

The robots are electromechanically driven and each has 6 joints capable of producing 6DOF motion in three Cartesian coordinates and three orientation axes, with accelerations up to approximately 2g and velocities of more than 2m/s. They are supplied with an ABB IRC5 controller which provides high level functionality including coordinate transformations, dynamic kinematic computations which take account of the robot geometry, inertia and payload information, and an axis computer which provides the feedback control loop for the motor drives. Importantly, the proprietary controller also provides several layers of safety controls to protect operators and equipment.

In normal operation, the user would program the robots using a high-level language called RAPID code. The instructions used in the RAPID code provide a powerful tool for quickly generating complex motion paths and creating loops and conditional operation patterns. This

language facilitates a variety of input and output (I/O) methods, including analogue and digital I/O channels as well as Ethernet communication protocols. The data sent and received on these channels can be used in the RAPID code in order to control the operation of the robots.

During robot operation, the RAPID interpreter executes motion instructions which pass the motion commands to a motion planning routine. This planning routine performs kinematic computations and sends the joint motor demands to the axis computer. As well as accepting position demands, the robots are capable of operating in force control mode, using 6DOF force and torque sensors mounted at the end effectors. In this case a nominal motion is specified as well as force demands, and deviations of the measured force from the force demand affect the motion path accordingly.

2.2. Control Interfaces

Three control interfaces for the input of motion demands to the robot controller are considered here, and their benefits and drawbacks discussed:

- high level (through RAPID)
- mid level (exploiting force control inputs and using auxiliary feedback)
- low level (direct access to axis computer of robot controller)

The biggest advantage of the first two options is that they retain the robust safety mechanisms of the proprietary controller, and thus permit faster development of auxiliary control without the fear of serious malfunction and injury or damage. The high level option also retains the matured control technology of the proprietary controller, providing the best motion path control for the least development effort. This is provided at the cost of deterministic real-time control; the RAPID code is not intended to receive, parse and compute small motion path segments on the fly in this manner. Methods to provide determinism in this case are available, but these introduce delays in the robot motion with respect to the numerical simulation. Furthermore, the execution cycle of the RAPID scripts has been found to be unreliable at higher position demand rates resulting in a sporadic response.

The mid level control option uses the input signals normally used for force control feedback to affect the motion of the robots. In this manner the RAPID interpreter and much of the motion planning algorithm is bypassed, resulting in a much faster control loop. The safety of the proprietary controller is preserved, as are the kinematic computations and coordinate transformations; thus the auxiliary control can still be applied in the Cartesian coordinate space. The voltage signals provided at the force control interface produce proportional velocities in the robot motion. There is no interface available at this level for positional feedback, so for closed loop position control it is necessary to take advantage of auxiliary sensors. A specific drawback of this method is that the new feedback control must be tuned and will not easily achieve the same performance as the inner loop proprietary control.

The third and final option is to directly access the axis computer of the robot controller. This method effectively bypasses all of the proprietary trajectory planning systems and sends demands directly to the position feedback controller for the robot motors. Whilst offering the best real-time control of the robots, adopting this approach requires a significant development

effort, and there can be a reduction from the intrinsic accuracy of the industrial controller. A further consideration is that this technique will undermine some of the more sophisticated elements of the safety controller, and alternative safeguards must be implemented.

3. RMR: Control and performance

The control paradigm chosen for the RMR combines two of the interfaces discussed above. In this way the benefits of each are largely preserved and the drawbacks predominantly avoided. The high level interface (RAPID) is used to define a nominal trajectory, and the low level interface (ORCA) is used to compensate for delays while remaining within appropriate margins of the known safe trajectory.

3.1. RAPID interface (high-level)

In this section the high-level approach is considered in detail. The biggest barrier to implementing a real time scheme using this approach lies in the non-determinism of the communication protocols and the unpredictable nature of the RAPID code interpretation. The former is imposed by the implementation of TCP/IP Ethernet communications on the robot controller. The unpredictable nature of the RAPID code is due to the fact that it is being interpreted on a processor running a variety of concurrent threads so execution can slow down when the processor is heavily loaded. In normal operation this is not perceptible but when positions are being demanded at rates of 10Hz or more the system is sensitive to small delays in the execution cycle. Methods for mitigating these effects and providing a real time deterministic motion based on the deterministic outputs of the numerical simulation are required.

The flow of information between the simulations and the robots is illustrated in Figure 3-1. The PXIe, the IRC5 controller and the robots are shown as three physical devices, and the distinct processes and threads running on the different devices can be seen. The important elements to note are the buffers on the PXIe and the IRC5 controller which form the gateways between the deterministic execution at either end of the diagram and the non-deterministic message processing in the centre of the diagram. These are necessary to ensure determinism but introduce undesirable delays.

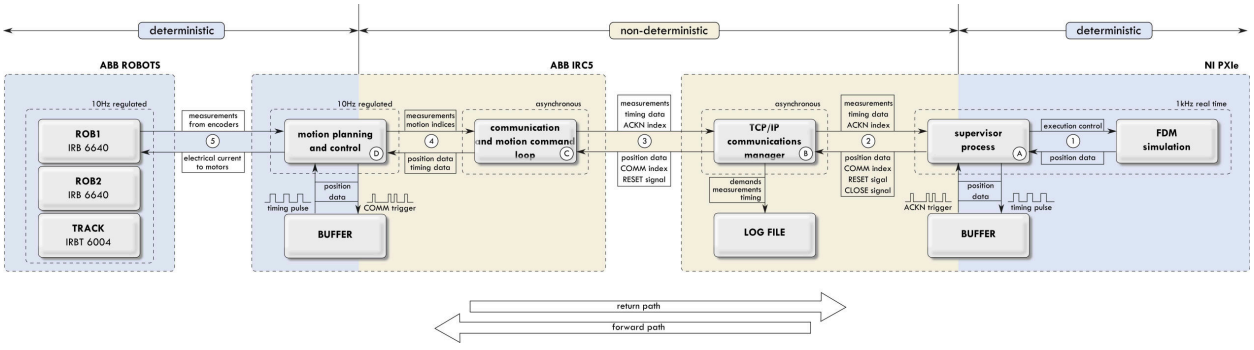


Figure 3-1 Position and control data flow between processes on the PXIe (real time) and IRC5 (proprietary robot) controllers

3.2. ORCA interface (low-level)

In order to facilitate low-level control of the robot hardware, the Open Robot Control Architecture (ORCA) of the University of Lund [17] has been adopted. This control uses a separate ORCA PC which intercepts signals sent between the main computer and the axis computer in the IRC5 controller. It can then augment or override the signals sent to the axis computer and demand joint motor positions and velocities directly. A schematic layout of the axis computer control is shown in Figure 3-2. ABB use position demands in conjunction with velocity and torque feed forward to produce accurate control of the robots. The torque signals are considered commercially sensitive, and are disabled by ABB as part of the licensing agreement for the ORCA interface, leaving the position and velocity feed forward available for use through ORCA. In addition, the controller gains can also be tuned through ORCA, raising the possibility of gain scheduling.

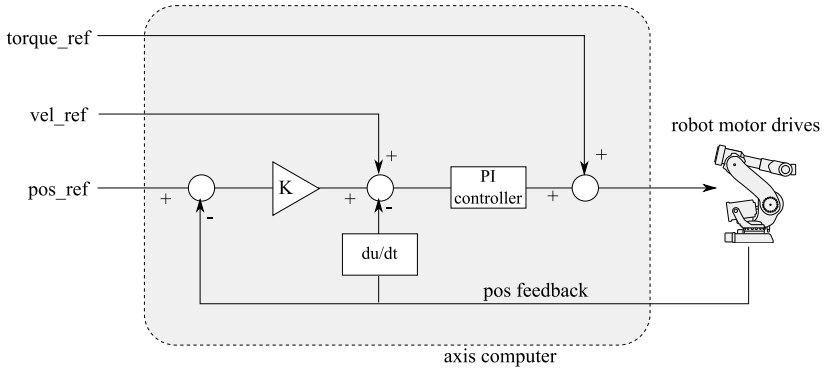


Figure 3-2 Schematic showing the operation of the axis computer

The biggest problem introduced with the use of the ORCA interface is that by directly passing demands to the axis computer, much of the robust safety intrinsic to the industrial control systems is bypassed. To minimise this risk, the approach adopted here is to use the ORCA interface only to *augment* the control of the robots. The high-level RAPID interface remains as the primary input to the robot control, with the ORCA interface used to augment the position and velocity in order to compensate the delay in the high-level control. The extent to which the ORCA interface can modify the signal from the IRC5 main computer is therefore limited, ensuring the robots do not deviate significantly from the safety-assured path determined by the main IRC5 controller. The layout of this system is shown in Figure 3-3. The approach presented can therefore produce fast system response times without forsaking the robust safety of the high-level approach.

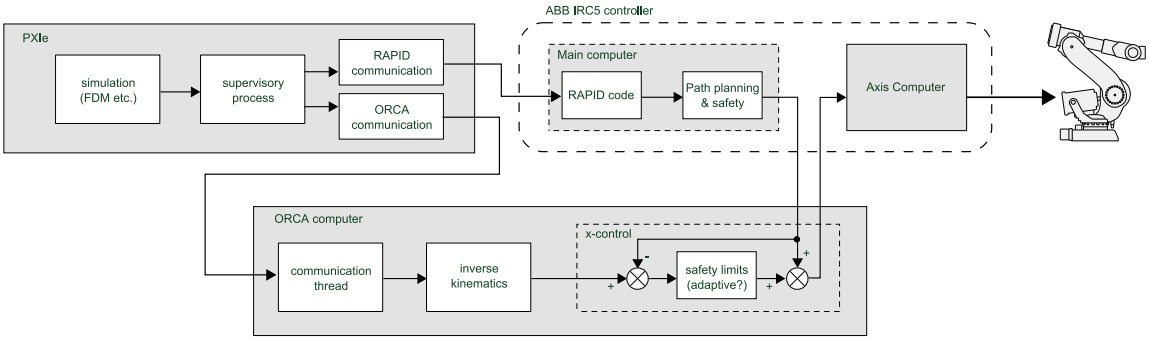
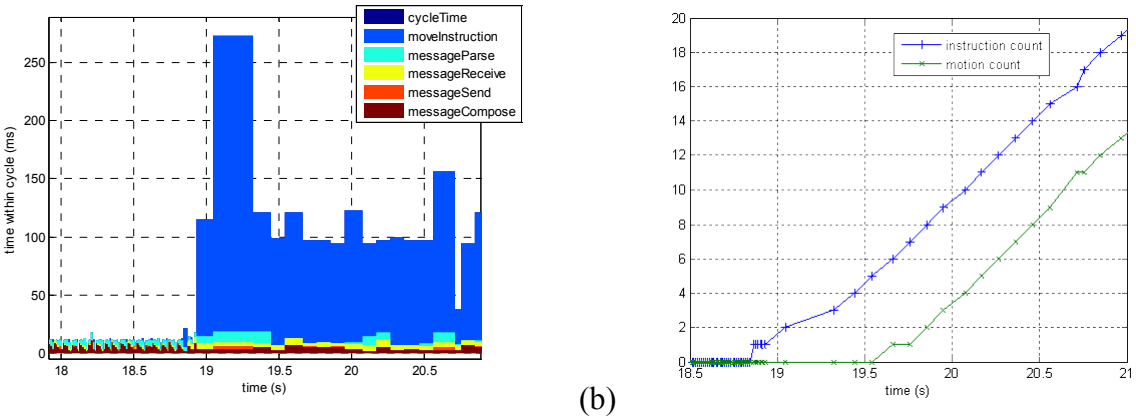


Figure 3-3 Layout of the combined RAPID-ORCA interface configuration

3.3. Performance

The IRC5 exhibits some interesting behaviour when it is first sent position demands through the RAPID interface, as illustrated in Figure 3-4. Figure 3-4(a) shows the timing of the RAPID code sections for each cycle, with the position demands sent at a rate of 10Hz starting just before 19s. The stacked bars indicate the times that the respective tasks have taken on the IRC5 controller for each time step. The total height of each stack is thus proportional to its width, representing the time for a full cycle to complete, and note that the precision of the measurements is 1ms. Figure 3-4(b) shows the number of move instructions executed in the RAPID code compared to the number of motions actually completed by the robots, ascertained using interrupts to trigger counters on the IRC5.



(a) (b)

Figure 3-4 Behaviour of the motion planning as the motion commands are commenced through the RAPID interface: (a) Stacked bar chart showing communication cycle timing; (b) Counters showing the number of motion instructions processed in the RAPID code and the number of motions completed by the robots.

When the position demands first commence, a small 20ms spike can be seen in Figure 3-1(a), caused by the increased time taken for the 'moveInstruction' task. This is the time taken for the motion instruction to be passed to the motion planner and execution of the RAPID interpreter to resume. The spike corresponds to the move instruction for the first position demand. The second move instruction takes around 100ms to process, and the third takes over 250ms, before settling down to a more regular pattern of approximately 100ms per cycle, corresponding to the 10Hz

demand rate. From Figure 3-1(b) it can be seen that the robot motion does not begin until after the move instruction times settle into this regular pattern. This delayed start, combined with the regulating effect of the ~100ms move instruction times, ensures a consistent delay of 0.5s between the position demand being received from the real time simulation and the motion of the robots meeting this demand. It was found that independent of the position demand rate, the robot controller always queued up motion instructions, thus regulating the delay at a minimum of 500ms. This demonstrates the necessity of augmenting the RAPID interface with the lower-level ORCA interface.

Figure 3-5 shows the step response of the robots to a Cartesian position demand. This demand is in the x direction for ROB2, and uses all six robot joints as well as the track motion. Three configurations are used: First, the response using only the RAPID interface, with a motion command rate of 5Hz. Second, the RAPID interface in conjunction with augmented position demands through the ORCA interface. Third, the RAPID interface in conjunction with augmented position *and* velocity demands through the ORCA interface.

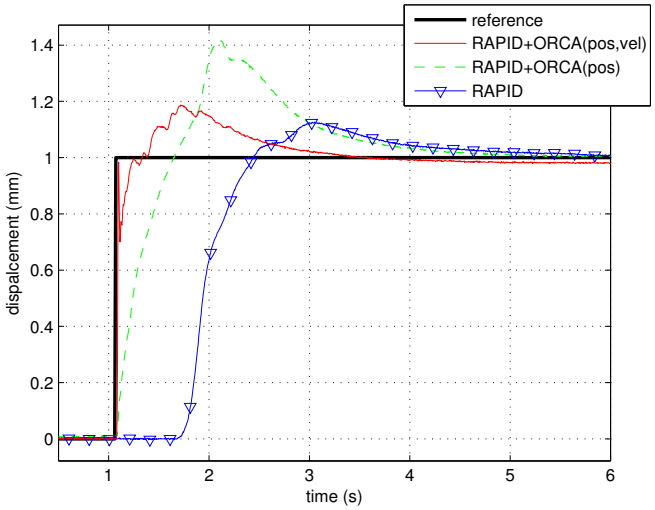


Figure 3-5 Step response for a Cartesian x-direction demand

The latency in the RAPID interface is clear, with more than half a second passing before any motion commences. The motion then exhibits a small overshoot and slowly approaches the reference signal. In contrast, both of the ORCA interface examples show a very fast response to the demand. The augmented position curve appears to be producing a similar overshoot to the first example, but because the velocity demand from the RAPID interface is unmodified, this produces a late acceleration, explaining the large overshoot that follows. This curve then converges on that of the RAPID interface. The final example, using both position and velocity augmentation in the ORCA interface, shows a dramatic improvement in the initial response, where the velocity is close to matching the step of the reference signal. There is then a rebound, presumed to be due to the inertia and elasticity of the robots, and the slower position control loop brings the signal back in line with the reference, again with some overshoot. In practice, a step response in the position demand is an unrealistic criterion for the motion of a physical system, so the rebound seen in the final example is not a concern. It is encouraging that the initial response shows minimal latency, measured to be in the region of 20ms for the initial response and a short rise time, making the complete system suitable for high fidelity non-contact MiL simulations.

4. Vision Tracking

The vision tracking is accomplished using a single camera and a model-based pose refinement algorithm, targeting a reduced-scale model of a satellite. Roke Manor Research provided the Real-time Attitude & Position Determination (RAPiD) vision tracking software [18] for the purpose of this study, which runs on a standalone Windows XP laptop with a direct network connection to the National Instruments PXIe box. A single Sony monochrome CCD USB camera provides the vision data RAPiD calibrated for both the camera and lens. The camera aperture was set to (F ~ 4) and the focus to (f ~ 2m). Vision tracking data was sent over the network via a UDP/IP protocol to the PXIe, where it formed an input signal to the simulation. In preliminary trials the system was tested using a virtual reality simulation in place of the camera feed.

4.1. Software

RAPiD is a model-based vision system using edge detection in conjunction with a known pattern of retroreflectors on the front face of the satellite. A Kalman filter is used to take a pseudo measurement of the difference between the expected position of the satellite and the measured position. This allows for the removal of outliers and ensures the system is robust. Minimising the sum of the perpendicular displacement between the measured edges and the expected edges allows for an optimised tracked satellite position [19]. If the satellite moves closer to the camera and is not fully included in the image, the control points that were active on the obscured section are disabled so that the satellite can still be tracked without being fully visible.

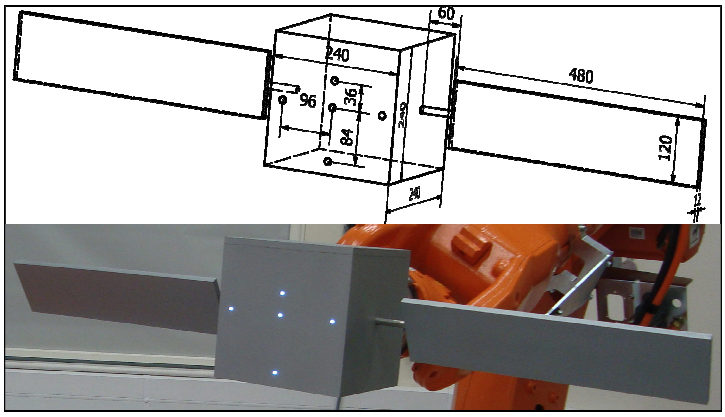


Figure 4-1: Physical Satellite Model

A scaled satellite model, with a span of 1.32m, was made in accordance with the model file for the vision tracking software. Figure 4-1 shows the chosen sizing of the model and an image of the actual model. In the current configuration, the vision tracker requires point lights in a known pattern in order to automatically lock on to the target. In these tests, LED lights have been used to emulate the reflected light from the retroreflectors.

For preliminary trials a satellite virtual environment was set up using the Virtual Reality Modelling Language (VRML) and provides a 6DoF representation of the rendezvous procedure in order to update the output video image.

5. Closed-loop System

A simple satellite model was set up in Simulink and a controller developed for it. These were then compiled to run on the PXIe and allowed the loop to be closed using the real camera sensor data. The schematic for this simulation is seen in Figure 5-1. Three levels of modelling were undertaken to test successively greater levels of performance in the RMR, the vision tracking and the satellite controller. 1DOF, 3DOF and 6DOF were used to allow both relative translation and rotation to be modelled.

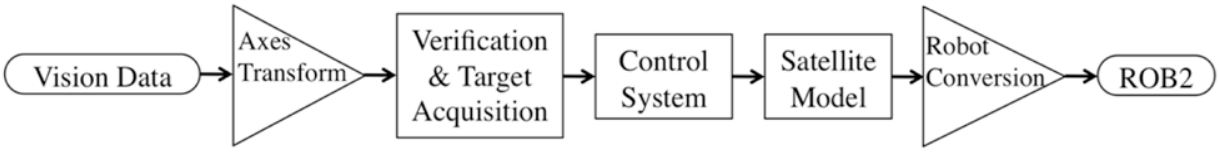


Figure 5-1: Schematic of Simulink Model.

5.1. Control System Design

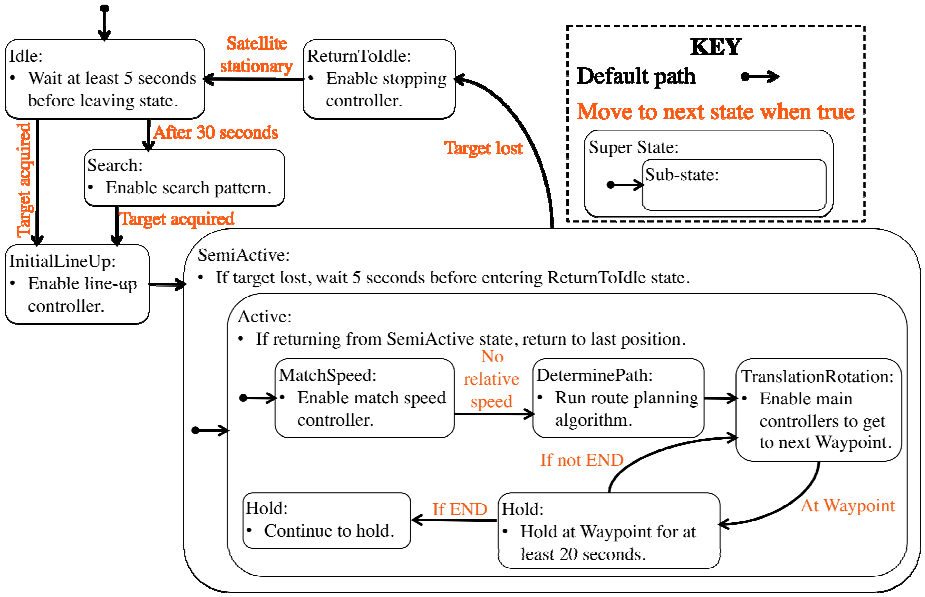


Figure 5-2: Example Stateflow Control Logic Schematic for 3DoF System

A control system was required that can cope with loss of tracking, a rotating target and the delays associated with the system. The control system developed was governed by a series of Stateflow® charts that switched between the various control states and waypoints when necessary, an example of which can be seen in Figure 5-2. A simple low-pass filter was added to the vision-tracking data in order to reduce the amount of noise in the signal with a relatively slow time constant of one second.

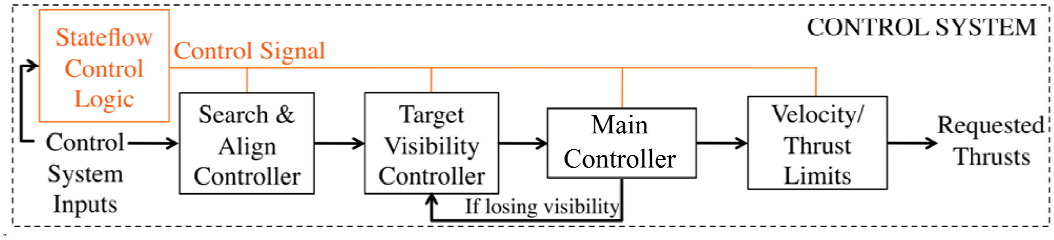


Figure 5-3: Control System Schematics for 3DOF System

6. Results

For the results in this section, testing was conducted in the RMR facility using only the RAPID interface, operating at a frequency of 10Hz. Because the dynamics of the tests being conducted are relatively slow, the added complexity of the ORCA interface for these tests was not required. The setup of the robots, the satellite model and the camera sensor can be seen in Figure 6-1.

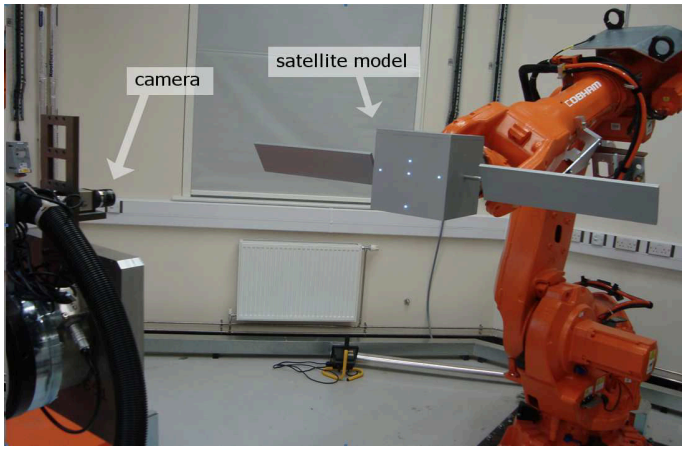


Figure 6-1: RMR Facility Setup.

For each of the following tests, the tracked robot with the camera mount was initialised at a distance of 7 metres from the satellite mounted on the second robot. For the 1DoF tests, the camera was pointed directly at the target, and moved along a single axis during the approach. The 3 DoF tests involve two axis translations and a relative rotation, and finally the 6 DoF tests allow the satellite to move along all three axes, and rotate about each, relative to the camera. Initial tests held the target satellite fixed whereas the most complex tests involved 6 DoF camera movement, together with the target satellite rotating.

Figure 6-2 shows 1DoF closed-loop MiL test results where the target satellite is at the origin and the chase satellite starts 7m away. The chase satellite must reach each of the three defined waypoints in order to move on to the subsequent waypoint. An enlargement of the final 10 seconds of the simulation is also shown and a steady-state 3.5cm error can be seen between the actual position and the position the vision tracker outputs. The calculated mean error between the vision tracker and the actual position was 4.8cm, representing a range error of 2.3%.

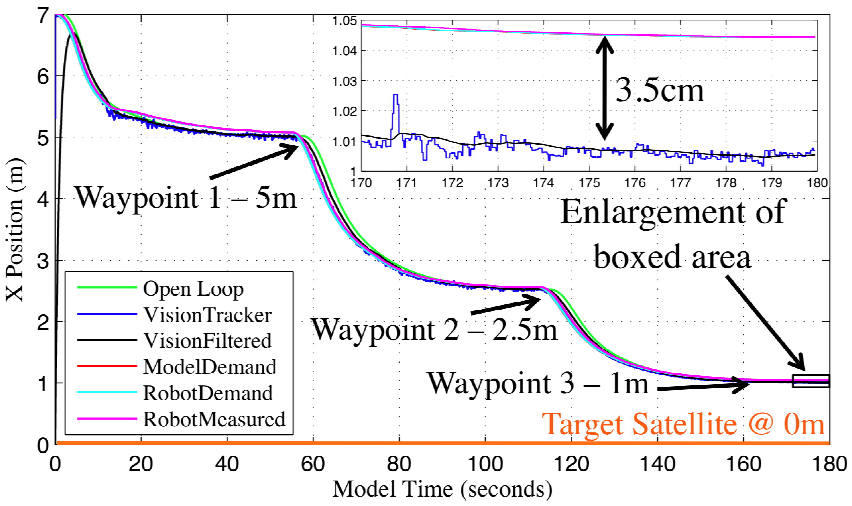


Figure 6-2: 1DoF Closed-Loop MiL Results

Figure 6-3 shows the 3DoF closed-loop MiL test results where the target satellite is at (0, -0.6) with a 20° yaw and the chase satellite starts at 7 metres behind it facing directly forwards. The chase satellite must reach the 3 waypoints, which have an X, Y and ψ component, in the correct order for a successful rendezvous. The mean errors between RAPiD and the known position were 3cm (1.42% range), 10.1cm (2.91%) and 2.47° for X, Y and yaw respectively. In particular, the rotation tracker data is seen to become increasingly noisy when getting close to the target satellite and sees a significant increase when the back edge of the satellite is lost from view, as labelled in Figure 6-3. The yaw data is seen to vary by ±0.5° before this point and ±1.5° after.

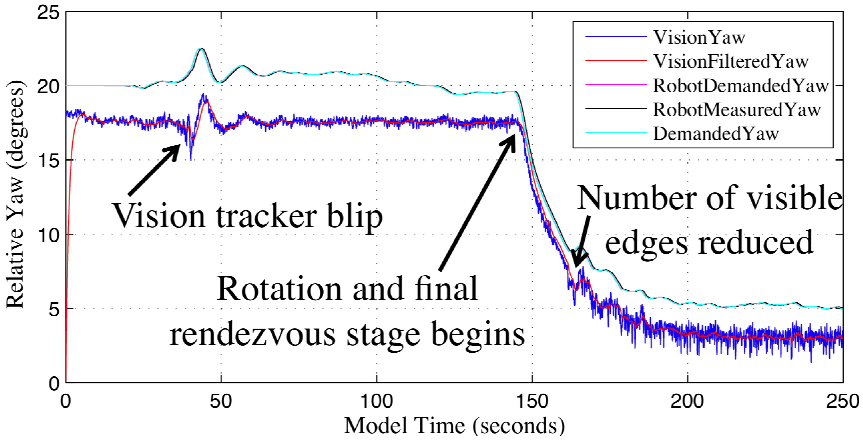


Figure 6-3: 3DoF Closed-Loop MiL Test Results – Relative Yaw

Figure 6-4 to shows the 3DoF rotating closed-loop MIL test results where the target satellite is at (0, 0.3) with an initial yaw of -30°, rotating at 9 degrees/min, and the chase satellite starts at (7, 0) facing directly forwards. The chase satellite must reach the 3 waypoints in the correct order and must stay facing the target satellite as it rotates for a successful rendezvous. The mean errors

between RAPiD and the known position were 3.6cm, 9.2cm and 2.77° for X, Y and yaw respectively. At the final waypoint the chase satellite continued to track the target and rotated with a constant offset, as seen in Figure 6-4, which shows the resultant yaw response.

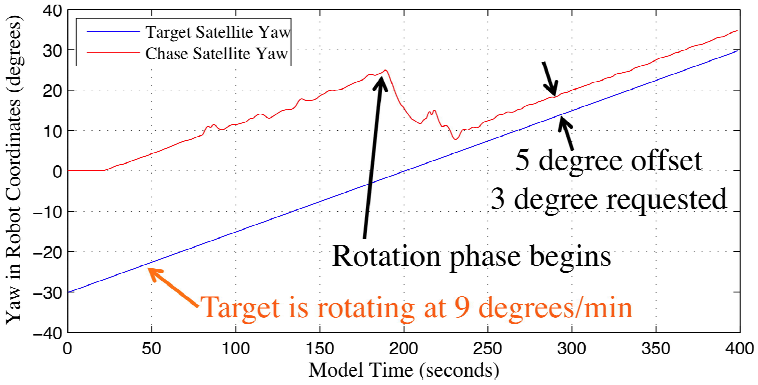


Figure 6-4: Rotating 3DoF Closed-Loop MiL Test

Figure 6-5 gives the X and Y separation in camera coordinates and demonstrates that whilst the target satellite was rotating, the chase satellite maintained a distance of 1.5m as requested.

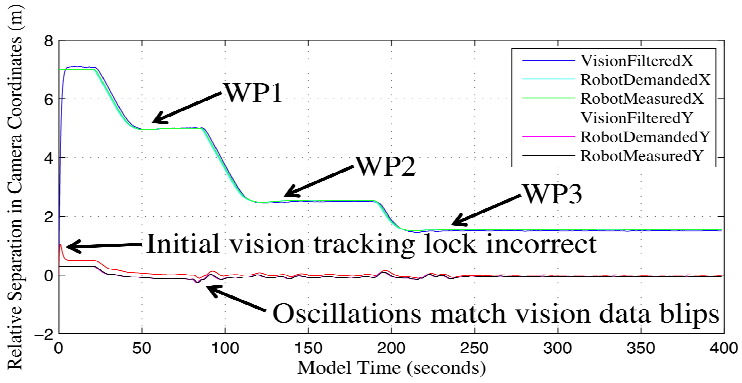


Figure 6-5: Rotating 3DoF Closed-Loop MiL Test Results

Figure 6-6 shows the rotating 6DoF closed-loop MiL test results where the target satellite is initially at (0, -0.4, -0.3) with Euler angles of (-40, 0, -20) for roll, pitch and yaw respectively and is rolling at 24 degrees per minute with the chase satellite initializing at (7, 0, 0) facing directly forward. The chase satellite must reach the waypoints as before but must continue to track the target satellite as it rotates. The mean errors for the 6DoF MiL rotating test were 2.6cm, 10.3cm, 11.9cm, 0.8°, 2.8° and 2.3° in X, Y, Z, roll, pitch and yaw respectively. The vision tracking for the roll axis performs particularly well for this test when compared to the pitch and yaw axes. The roll errors were ±0.5° or less for the whole test, whilst the pitch errors varied between -9° and +5° and the yaw had the 2° steady state error.

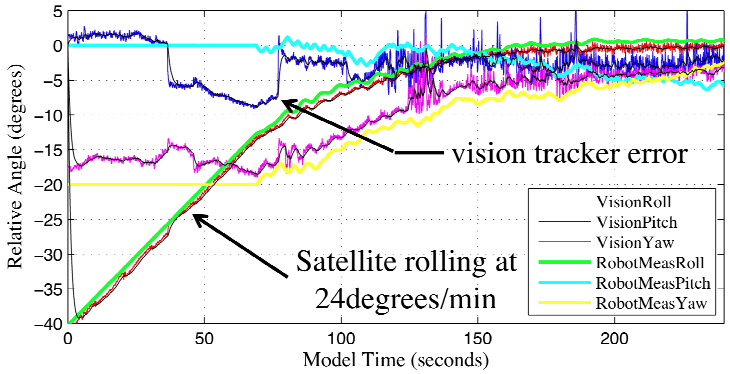


Figure 6-6: Rotating 6DoF Closed-Loop MIL Test Results

7. Conclusions

A two robot facility has been set up at the University of Bristol in the UK for the study of MiL relative motion robotics. This is capable of responding with a latency in the region of 20ms whilst running real-time complex 6DoF Matlab models.

The results in this paper demonstrate stable closed loop MiL tests up to and including 6DoF tracking and rendezvous, with initial translation and rotation offsets together with steady state non zero rotation speeds. A single monochrome camera was used to generate the real-time vision track data and errors which were shown to be sufficient for robust and repeatable tracking and rendezvous.

Acknowledgements

Many thanks to Ed Sparks and Roke Manor Research for providing the RAPiD vision tracking software and to Anders Robertsson, Anders Blomdell and Klas Nilsson at the University of Lund, Sweden for the considerable time they have spent helping with the setup of the low-level ORCA interface with the ABB IRC5 controller.

The work described here has been largely funded by Cobham Mission Equipment as part of the ASTRAEA Programme, which seeks to enable the routine use of UAS (Unmanned Aircraft Systems) in all classes of airspace without the need for restrictive or specialised conditions of operation. The ASTRAEA programme is co-funded by AOS, BAE Systems, Cobham, EADS Cassidian, QinetiQ, Rolls-Royce, Thales, the Technology Strategy Board, the Welsh Assembly Government and Scottish Enterprise.

References

[1] NASA, Orbital Debris Quarterly News, 2000-2012, http://orbitaldebris.jsc.nasa.gov/.

[2] Operational Satellite Statistics, UCS Satellite Database, 2012, http://www.ucsusa.org/.

[3] Defense Advanced Research Projects Agency, The Pheonix Project, 2011, http://www.darpa.mil/our_work/tto/programs/phoenix.aspx.

[4] F. Qureshi, D. Terzopoulos and P. Jasiobedzki, Cognitive Vision for Autonomous Satellite Rendezvous and Docking, *Proceeding of the 9th IAPR Conference on Machine Vision Applications*, 2005.
[5] Two Chinese Satellites Rendezvous in Orbit, The New Scientist, Volume 207, Issue 2776, 2010.
[6] T.E. Rumford, Demonstration of Autonomous Rendezvous Technology DART Project Summary, *Proceedings of the SPIE Conference 5088*, 2003.
[7] J. Kelsey, J. Byrne, M. Cosgrove, S. Seereeram and R. Mehra, Vision-Based Relative Pose Estimation for Autonomous Rendezvous And Docking, *Aerospace IEEE Conference*, 2006.
[8] J. Mitchell, S. Cryan, D. Strack, L. Brewster, M. Williamson, R. Howard and A. Johnston, Automated Rendezvous and Docking Sensor Testing at the Flight Robotics Laboratory, *Aerospace Conference 2007 IEEE*, 2007.
[9] D. Pinard, S. Reynaud, P. Delpy, and S. Strandmoe, Accurate and Autonomous Navigation for the ATV, *Aerospace Science and Technology*, **11**, 2007, Pages 490-498.
[10] F. Sellmaier, T. Boge, J. Spurmann, S. Gully, T. Rupp and F. Huber, On-Orbit Servicing Missions: Challenges and Solutions for Spacecraft Operations, *SpaceOps Conference*, 2010.
[11] T. Boge, T. Wimmer, O. Ma and T. Tzschichholz, EPOS – Using Robotics for RvD Simulation of On-Orbit Servicing Missions, *AIAA Modelling and Simulation Conference*, 2010.
[12] JL du Bois, B Titurus, NAJ Lieven, 'Transfer Dynamics Cancellation in Real-time Dynamic Substructuring', ISMA, Leuven, Belgium, Sept 2010
[13] MATLAB®/Simulink ® R2012a (software), The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098, UNITED STATES, 2012.
[14] ABB Robotics Ltd., Auriga House, Precedent Drive, Rooksley, Milton Keynes, Buckinghamshire, MK13 8PQ, United Kingdom.
[15] JL du Bois, P Thomas, T Richardson, 'Control Methodologies for Relative Motion Reproduction in a Robotic Hardware in the Loop Simulation of Aerial Refuelling ', AIAA Guidance, Navigation, and Control Conference, August 2012.
[16] J du Bois, P Thomas, T Richardson, 'Development of a Relative Motion Facility for Simulations of Autonomous Air to Air Refuelling', IEEE Aerospace Conference, Big Sky, MT, USA, March 2012.
[17] Blomdell, A., Dressler, I., Nilsson, K., Robertsson, A., and Dressler, I., "Flexible application development and high-performance motion control based on external sensing and reconfiguration of ABB industrial robot controllers," Proc. ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications, the 2010 IEEE Int. Conf. on Robotics and Automation (ICRA2010), 2010.
[18] RAPiD Roke Manor Research, Romsey, Hampshire, SO51 0ZN, UK
[19] A. Blake and A. Yuille, Active Vision, MIT Press, Cambridge, 1993.
[20] U.S. Department of Defense, Missile Flight Simulation, MIL-HDBK-1211, 1995.