Robotic Relative Motion Reproduction for Air to Air Refuelling Simulation

P. R. Thomas⋆, T. S. Richardson⋆ and J. L. du Bois⋆⋆
⋆*Department of Aerospace Engineering, University of Bristol, University Walk, Bristol, BS8 1TR*
⋆⋆*Department of Mechanical Engineering, University of Bath, Bath, BA2 7AY, UK*

Abstract
Hybrid testing involves the testing of physical subassemblies in a real-time simulation, coupled to numerical simulations of complete systems and their related environmental conditions. It offers the ability to test the performance of components in a realistic operating environment while remaining within the low-cost, repeatable, and relatively safe confines of a laboratory. The University of Bristol and Cobham Mission Equipment have developed such a hybrid test facility for relative motion simulation between two independent bodies. This paper describes the use of the facility in replicating an air-to-air refuelling environment and how key latency issues were addressed.

1. Introduction

Air to air refuelling (AAR) is an important capability to extend the range, payload, and endurance of aircraft. While there has been no interest in adopting AAR for civilian aviation there are a few proponents of its application,[1] promoting reductions in fuel consumption for passenger and freight transport, the reduction of airport loading, the extension of range and payload in existing aircraft, and increased scope for scientific and environmental surveys through improved endurance. For unmanned aerial systems (UAS), where endurance is no longer limited by pilot fatigue, aerial refuelling capabilities offer significant benefits. Refuelling operations have historically been conducted by human pilots who require a high level of training and fast response times, and as such can not (in the near future) be conducted for remotely piloted aircraft over existing data links. The recent proliferation of UAS (including combat systems) has therefore resulted in a demand for automated air-to-air refuelling (AAAR) capabilities and it is this requirement that motivates the work described in this paper. Successful accomplishment of AAAR relies on the development of two key technologies: Firstly, position sensing and tracking, to determine the relative position between aircraft and refuelling equipment; and secondly, control strategies to enable robust and safe operation of the aircraft in steering them to their target.

The development of these two technologies relies on sophisticated testing: the sensor development requires physical tests under realistic conditions, while the control algorithm development leans heavily on realistic sensor data to ensure robust operation. It is advantageous to perform as many of these tests, and recreate as realistic conditions as possible, in a laboratory environment. This permits the testing of sensors and systems critical to the safety of equipment and personnel with reduced risk, and facilitates stage-gate management of large projects to mitigate financial risks. The work described here is concerned with creating both a laboratory model-in-the-loop (MiL) test facility that can satisfy these requirements, and a simulated refuelling environment that is a sufficiently representative for such testing, in order to provide the most comprehensive evaluation possible for these new aerial refuelling technologies prior to flight testing.

Often this type of work will be related to sensing requirements, with no direct contact between the two bodies. A prominent example is the case of satellite docking approach, and it is unsurprising that some of the first large scale relative motion hybrid testing experiments have been focused on this problem.[2–5] The system dynamics in an AAR context have much shorter timescales than satellite manoeuvres and the relative motion is more erratic due to the stochastic nature of atmospheric turbulence and aerodynamic coupling. Robots have been used in other works to simulate aircraft motion in refuelling operations; for example Pollini *et. al.*[6] used a robot to recreate aircraft motion to test vision system algorithms, but the aircraft control loop was not closed.

The facility discussed in this paper uses two industrial robotic arms to manipulate refuelling hardware, driven by a numerical simulation of the AAR scenario. The assembly of such a simulation model is no trivial task.[7,8] The 'hook-up space' is a relatively compact environment with complex interactions between aircraft, refuelling equipment, and aerodynamic effects from the receiving aircraft's bow wave, the wake vortices emanating from the tanker, and air turbulence. The model described in this paper provides a simulated environment for testing and evaluating a number of different control approaches and designs, including autonomous control logic. Then, by incorporating prototype or production sensors into the facility, the measurable quantities can be fed back into the simulation model to close the control loop and provide realistic tests in a controlled environment

The first part of this paper describes the MiL testing facility and discusses the control interfaces to achieve the necessary response performance. The second part discusses the AAAR simulation environment which has been developed to replicate the refuelling environment. Results from running the simulation on the facility are then presented.

2. Relative Motion Facility

The relative motion robotics (RMR) facility is comprised of two 6 degree-of-freedom (6DOF) robotic manipulators, one of which is mounted on a linear track. Full scale refuelling hardware is mounted on the robots, and a large range of relative motion can be accommodated to simulate the final 10 m of an approach in an aerial refuelling procedure. The robotic manipulators with the refuelling hardware are shown in Figure 1. The robotic cell comprises two ABB IRB6640 production-line robots, designated R1 and R2, and are supplied with a proprietary IRC5 controller. R1 is secured to the ground whilst R2 is mounted on the 7.7 m IRBT6004 track to permit translation of the robot base at a rate of 1.6 ms^{-1}. The robots are electro-mechanically driven and each has 6 joints capable of producing 6DOF motion in three Cartesian coordinates and three orientational axes, with accelerations up to $2g$ and velocities of more than 2 ms^{-1}. The IRC5 controller provides a high level of functionality, including coordinate transformations, kinematic computations, and motor control loops which take account of the robot geometry, inertia and payload information. Importantly, the proprietary controller also provides several layers of safety controls to protect operators and equipment. A system diagram of the RMR facility is shown in Figure 2.

In normal (factory default) operation the user would program the robots using a high-level language called RAPID. 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. In our bespoke MiL operation, simulation models are 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 of 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. During either normal of MiL 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.

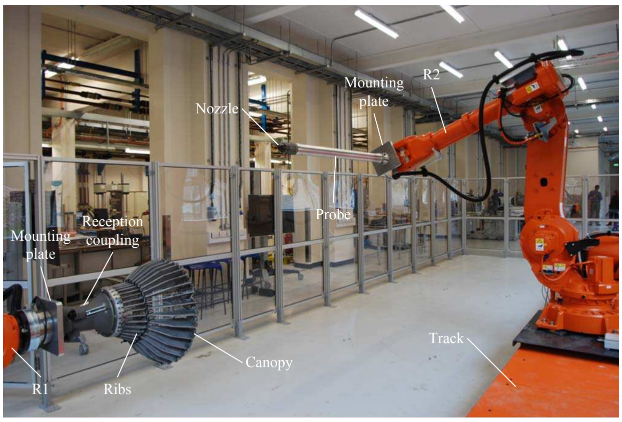


Figure 1: The relative motion robotics facility mounted with refuelling hardware

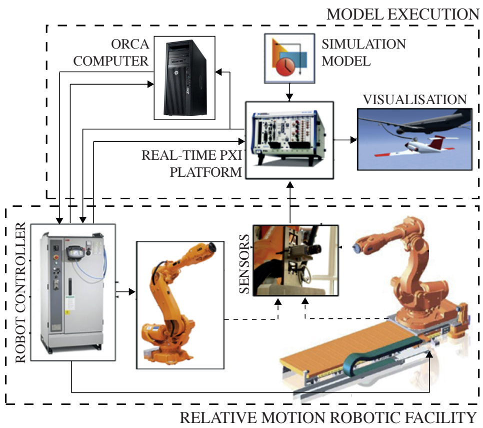


Figure 2: Layout of the facility as a synthetic environment for replicating air to air refuelling

3. Control Interfaces

Three control interfaces for the input of motion demands from the simulation to the robot controller were initially considered:

1. high level (through RAPID)
2. mid level (exploiting force control inputs and using auxiliary feedback)
3. 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. However 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. 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 whilst the safety of the proprietary controller is preserved, as are the kinematic computations and coordinate transformations. The 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. Whilst offering the best real-time control of the robots, adopting this approach requires significant development effort, and there can be a reduction from the intrinsic accuracy of the industrial controller. A further consideration is that this technique undermines some of the more sophisticated elements of the safety controller, and alternative safeguards need to be implemented.

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

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 thought to be 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 10 Hz or more the system is sensitive to small delays in the execution cycle. Safety mechanisms on the controller are also thought to play a part and the controller will halt the motion if it cannot meet certain stopping distance criteria. 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. 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.

3.2 ORCA interface

In order to facilitate low-level control of the robot hardware, the Open Robot Control Architecture (ORCA) of the University of Lund[9] 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. 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.

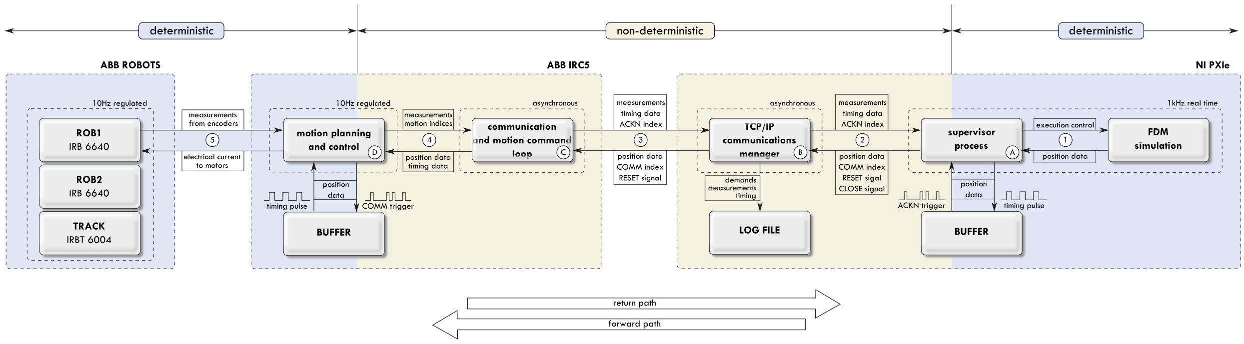


Figure 3: Position and control data flow between processes on the PXIe and IRC5 controllers.

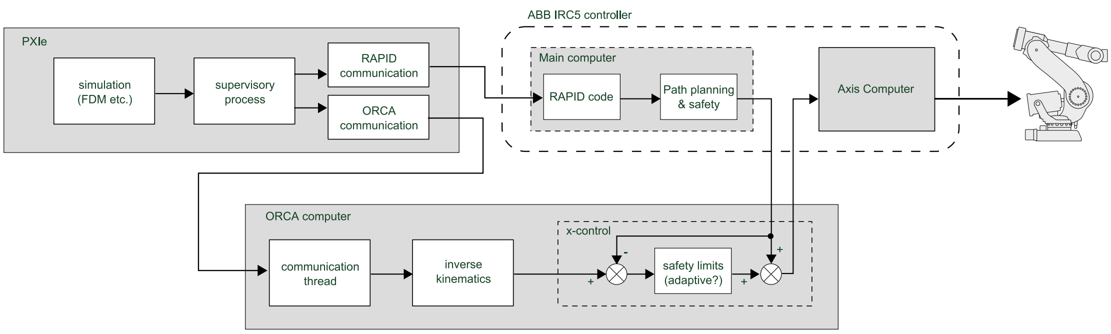


Figure 4: Layout of the combined RAPID-ORCA interface.

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 4. The approach presented can therefore produce fast system response times without forsaking the robust safety of the high-level approach.

3.3 Performance

Figure 5 shows the step response of the robots to a Cartesian position demand. This demand is in the x direction for R2, 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 5 Hz. 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.

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

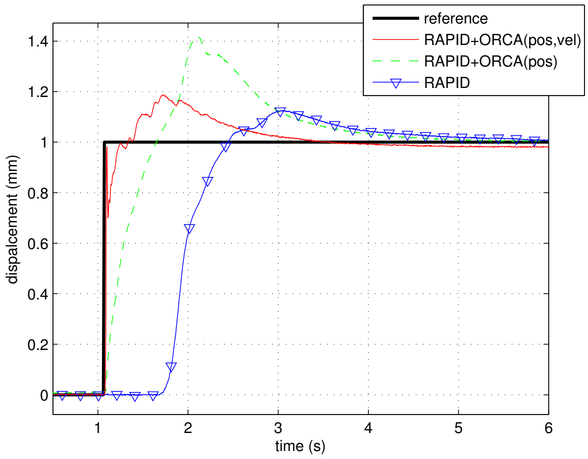


Figure 5: Robot step-response performance.

20 ms for the initial response and a short rise time, making the complete system suitable for high fidelity non-contact MiL simulations.

4. Air to Air Refuelling Simulation

Simulations are written in Simulink and compiled for use on the PXIe platform using National Instruments' Veristand target language compiler. The simulation model (see Figure 6) takes into account:

- Aircraft (tanker and receiver) trajectory demands, navigation logic, and flight control systems.
- Dynamics models for the aircraft, hose, and paradrogue assembly.
- Atmospheric (gust and wake) and bow wave disturbance models

Both the receiver and tanker are rigid-body, six degrees of freedom objects having nonlinear aerodynamic behaviour in the form of lookup data. Reference commands from the guidance and navigation systems are used by each aircraft's flight control system to generate input commands to the aircraft. These in turn, along with the dynamic aircraft states, are used to generate the aerodynamic forces and moments on the aircraft. In addition to the forces and moments from engine thrust and gravity, the total forces and moments are then used to recurrently solve the equations of motion. The guidance and navigation systems for the tanker define a fixed-heading or racetrack manoeuvre for the tanker to follow, whilst the receiver uses a waypoint-based navigation system controlled by a finite state-machine. Both aircraft have lower level control and stability augmentation systems, appropriate to their type.

Additional intermittent forces and moments on the aircraft come from atmospheric turbulence which is characterised by random, homogenous, and isotropic behaviour. It can be modelled well by passing white noise with unity spectral density through a low-pass shaping filter that gives the desired output spectrum. The continuous Dryden-form for the filters are used since they possess rational power spectral densities making their modelling far simpler.[10] The aerodynamic model for the wake vortices is based on the work by Saban, Whidborne, and Cooke.[11] The induced velocity generated by a finite wing is modelled using Weissinger's lifting line theory in combination with Kurylowich's vortex model. The Weissinger lifting line model is the simplest vortex panel method. It works well for swept wings and converges to reasonably good solution in both low and high aspect ratio limits. The Kurylowich model on the other hand is simple and effective in merging the individual vortices generated by a finite wing into a pair of counter-rotating wingtip vortices. A section through the wake flow field at 30 m distance behind the tanker is shown in Figure 7.

The trailing hose and drogue is modelled as a series of lumped-mass, rigid cylindrical links connected by frictionless ball-and-socket joints, after the work by Ro *et. al.*[12,13] Link masses, and gravitational and aerodynamic forces, are lumped at the joints, and drogue mass and forces, modelled with a parametric drag equation, are applied to the end of the final link. The turbulence effects applied to the aircraft are similarly applied to the hose and drogue. The bow wave preceding the receiver, which additionally upsets the drogue's flight, is approximated with the flow generated around a three-dimensional Rankine half-body. The behaviour of a hose drum unit is also modelled to reel in and out the hose based on the tension modelled in the hose linkages.

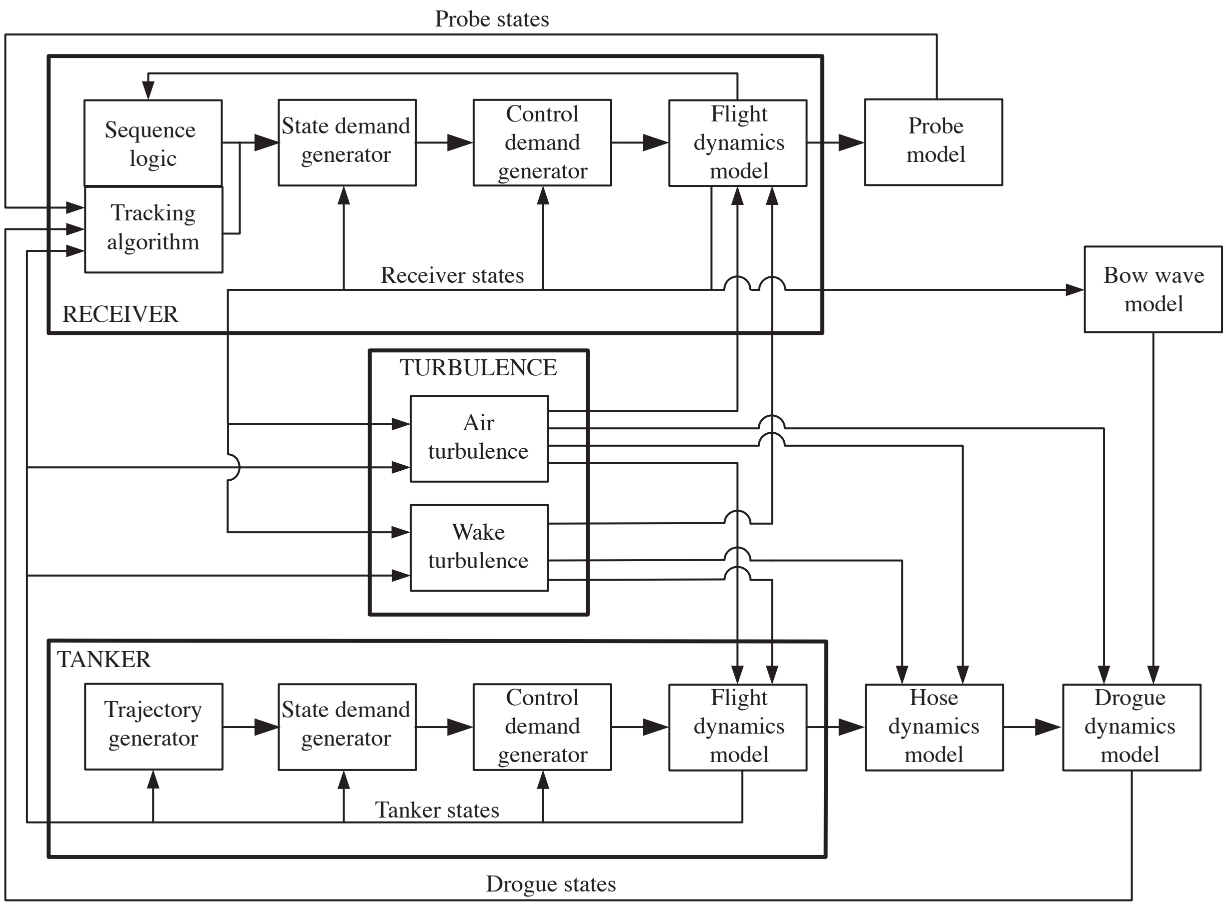


Figure 6: System diagram of the air to air refuelling simulation model

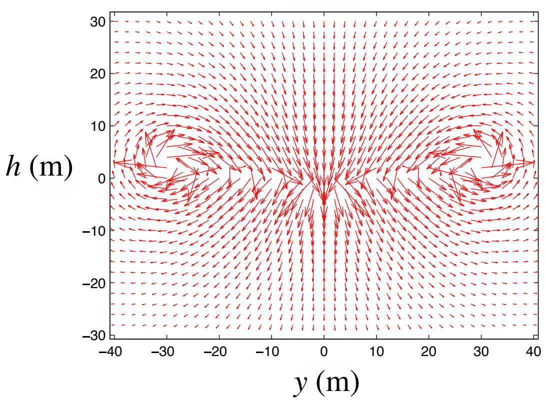


Figure 7: Typical wake vortex field section

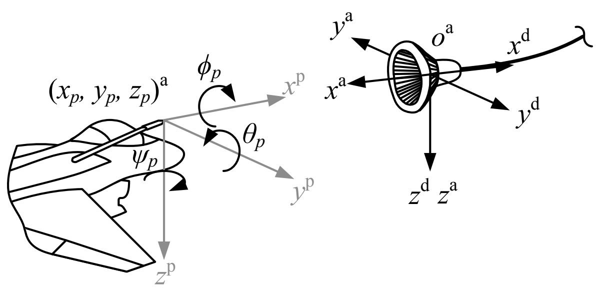


Figure 8: Probe (p), drogue (d), and approach (a) axes definitions

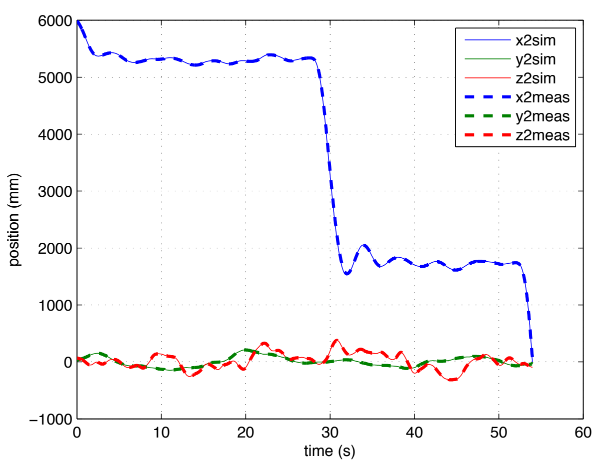
The output of the simulation for its use in the RMR is the position and orientation information for the probe and drogue which is replicated by the manipulators. To that end we define a set of axes systems as in Figure 8 which identifies the refuelling probe (p) and paradrogue (d) objects. The task in probe-drogue AAR is to approach and couple the probe with the drogue. Consequently the probe must track and close the range between it and the drogue, this is described in terms of the approach frame (a) which is coincident with the datum – the location of the drogue when trailed from the tanker, in still air. The probe and drogue positions are then described with the coordinates (x_p, y_p, z_p) and (x_d, y_d, z_d), relative to the origin o^a.

5. Real-time MiL Simulation

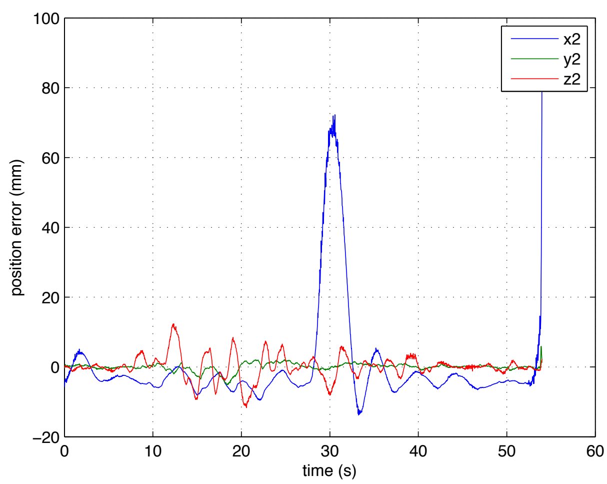
For the MiL tests the probe and drogue motion are made relative to the drogue's datum point, and are reproduced independently by R2 and R1 respectively. One degree of redundancy remains: the track motion. This is resolved by separating the motion of the probe into high and low-frequency components; the robot axes are used to perform the high-frequency motion and the track moves the robot base to provide the low-frequency, quasi-static response and give the probe its full longitudinal operational range. Figure 9a shows the absolute position data, including the 3DOF translational position output from the simulation overlaid with the measured response of the robots. At this scale the lines appear coincident. The simulation shows a position hold approximately 5 m aft of the drogue, followed by an approach to the pre-contact position, which is again held approximately 2 m aft of the drogue, and finishing with an aggressive engagement. The simulated refuelling procedure is conducted in light turbulence. Figure 9b shows the position error from these plots. In making this comparison the time base of the measured signal was adjusted to account for the delay, with the intention of presenting position errors rather than temporal errors. It is interesting to note that any effects from corner path artefacts are indiscernible in the presence of other disturbances. The large peak at approximately 30 s corresponds with the rapid approach of the receiver aircraft to the pre-contact position. A similar sharp rise is seen at the end of the plot where the final engagement is made. Although these errors are presented here as positional errors, they are found to be better described as temporal discrepancies; the differences seen are the result of a lag between the demanded motion and the measured robot position when moving at high speeds. What is not apparent in this figure, but can be determined from close examination of Figure 9a, is that while the robot motion lags the demand at some points, it leads the demand at others. This may point to a variable frequency response that could be characterised to the ends of further improving the performance using a feedforward control approach.

6. Conclusions

An overview of the Relative Motion Robotic (RMR) facility at the University of Bristol, developed in collaboration with Cobham Mission Equipment, has been given and the important considerations in implementation and performance have been discussed. Timing is critical in structural hardware in the loop simulations, and factors affecting performance in this regard have been described. Steps taken to improve the performance and to push the limits of the equipment being used were discussed. The outcome of this work has been a facility to implement high-bandwidth, closed-loop, hybrid simulations in a robust, stable, and realistic manner. A high fidelity multi-entity simulation model has been developed and sample results from a simulated air to air refuelling exercise on the facility have been presented. These results demonstrate the suitability of the RMR for conducting advanced tests of aerial refuelling hardware and sensors, for the purpose of developing automated aerial refuelling capabilities. This will lead to tests of novel sensing and control technologies for autonomous air-to-air refuelling. Beyond this the focus will be shifted towards including force feedback capabilities and developing methods for simulating discontinuous contact events.



(a) Absolute positions



(b) Relative position error

Figure 9: Positional data from the real time flight simulation

7. Acknowledgments

The authors wish to express their gratitude 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] Nangia, R. K., Operations and aircraft design towards greener civil aviation using air-to-air refuelling, *The Aeronautical Journal*, 110(1113):705–721, 2006.

[2] Newhook, P. and Eng, P., A robotic simulator for satellite operations, in *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Montreal, Canada, June 2001.

[3] Kaiser, C., Rank, P., Krenn, R. and Landzettel, K., Simulation of the docking phase for the SMART-OLEV satellite servicing mission, in *Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i- SAIRAS)*, Hollywood, California, USA, 2008.

[4] Boge, T., Wimmer, T., Ma, O. and Zebenay, M., EPOS–A robotics-based hardware-in-the-loop simulator for simulating satellite RvD operations, in *Proceedings of the 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Sapporo, Japan, August–September 2010.

[5] Suatoni, M., Mollinedo, L., Barrena, V., Colmenarejo, P. and Voirin, T., Use of COTS robotics for on-ground validation of space GNC systems: Platform dynamic test bench, in *Proceedings of the 11th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Turin, Italy, September 2012.

[6] Pollini, L., Mati, R. and Innocenti, M., Experimental evaluation of vision algorithms for formation flight and aerial refueling, in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Providence, Rhode Island, August 2004.

[7] Dogan, A., Sato, S. and Blake, W., Flight control and simulation for aerial refueling, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, California, USA, August 2005.

[8] Wang, J., Patel, V. V., Cao, C., Hovakimyan, N. and Lavretsky, E., Novel L1 adaptive control methodology for aerial refueling with guaranteed transient performance, *Journal of Guidance, Control, and Dynamics*, 31(1):182–193, 2008.

[9] 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, in *Proceedings of the ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications*, Anchorage, Alaska, May 2010.

[10] U.S. Military Specification, Flying qualitites of piloted aircraft, MIL-F-8785C, November 1980.

[11] Saban, D., Whidborne, J. and Cooke, A., Simulation of wake vortex effects for UAVs in close formation flight, *The Aeronautical Journal*, 113(1149):727–738, 2009.

[12] Ro, K., Basaran, E. and Kamman, J. W., Aerodynamic characteristics of paradrogue assembly in an aerial refueling system, *Journal of Aircraft*, 44(3):963–970, 2007.

[13] Ro, K. and Kamman, J. W., Modeling and simulation of hose-paradrogue aerial refueling systems, *Journal of Guidance, Control, and Dynamics*, 33(1):53–62, 2010.