Weapon Separation Analysis Tools using Matlab^{TM}

Mr Graham Akroyd
Aircraft Stores Compatibility Engineering Agency
Aerospace Operational Support Group
RAAF Base Edinburgh, Australia

Graham.Akroyd@defence.gov.au

ABSTRACT

Computer based tools have been used for many years to aid the separations engineer in the task of evaluating the separation behaviour of weapons from combat aircraft. The regular ‘toolbox’ usually includes a means of measuring the separation trajectory from cine film of the flight tests, a 6 DOF separation simulation code and a tool to display the predicted and measured trajectories. The development of such a toolbox has usually been protracted, manpower intensive and subsequently very costly. At the Aircraft Stores Compatibility Engineering Agency tools have been developed by leveraging the capabilities of Matlab^{TM} and Simulink^{TM}. The developed capabilities enable the simulation of weapon separation autopilots to be undertaken with confidence.

This paper describes the toolbox and capabilities.

1.0 INTRODUCTION

When weapons are released from an aircraft there can be significant aerodynamic and mechanical interaction between them. This combination of complex interactions can result in the weapon colliding with the aircraft, with possible loss of aircrew and aircraft. In other cases separation may be safe but result in disturbances that render the weapon operationally ineffective.

In the past, separation evaluations were done at some risk by flight test alone or at best with wind tunnel scale model jettison tests. Since the seventies, computers have enabled some level of numerical simulation to be used pre-flight to reduce the risks. Weapon telemetry and photogrammetric tools were developed to enable the measurement of flight test trajectories and comparison to the mathematical models to be achieved so as to improve the accuracy of simulations as a test program progresses. Such a process led to reductions in flight test requirements and subsequent integration costs and generally improved separations clearance envelopes.

The software toolbox used by the separations engineer has been expensive and time consuming to develop using traditional programming languages. The author’s own experience and anecdotes from colleagues around the world suggests that around 15 man years effort and 5 years elapsed time is not uncommon.

Paper presented at the RTO AVT Symposium on “Functional and Mechanical Integration of Weapons and Land and Air Vehicles”, held in Williamsburg, VA, USA, 7-9 June 2004, and published in RTO-MP-AVT-108.

Matlab^{TM} is a language for technical computing. It integrates computation, visualisation, and programming in an easy-to-use environment. Simulink^{TM} is a coupled package for modelling and simulating dynamic systems using a visual environment. The improvement in the capabilities of such modelling software in recent years led to an investigation by ASCENG into their potential for use in store separation, with ultimately very satisfying results.

2.0 OVERVIEW OF MATLAB/SIMULINK

Matlab^{TM} is a high-level interpreted computing language with a basic data element that is an array that does not require dimensioning. It is especially suited to technical computing requiring matrix and vector formulations, for which many fast and optimised functions are provided. The interactive nature of the interpreted language means that new routines can be developed very quickly and easily. It includes high level functions for 2D and 3D data visualisation and presentation graphics. There are also facilities for developing graphical user interfaces. The syntax is ‘C’ like, supports data structures, and has similar flexible file input / output capabilities.

Simulink^{TM} is directly linked to Matlab and allows models of dynamic systems to be built graphically as block diagrams from a library of blocks through drag and drop operations. New blocks can be created by the user (S-functions) using ‘C’, Fortran or Matlab^{TM} and added to the library. The diagrams are hierarchical and can be hidden as sub-system blocks within other block diagrams. A defined model can be integrated using a choice of methods and activated from the Matlab^{TM} command line. The package has particular application in control system design and simulation.

For more information, see the Mathworks website (www.mathworks.com).

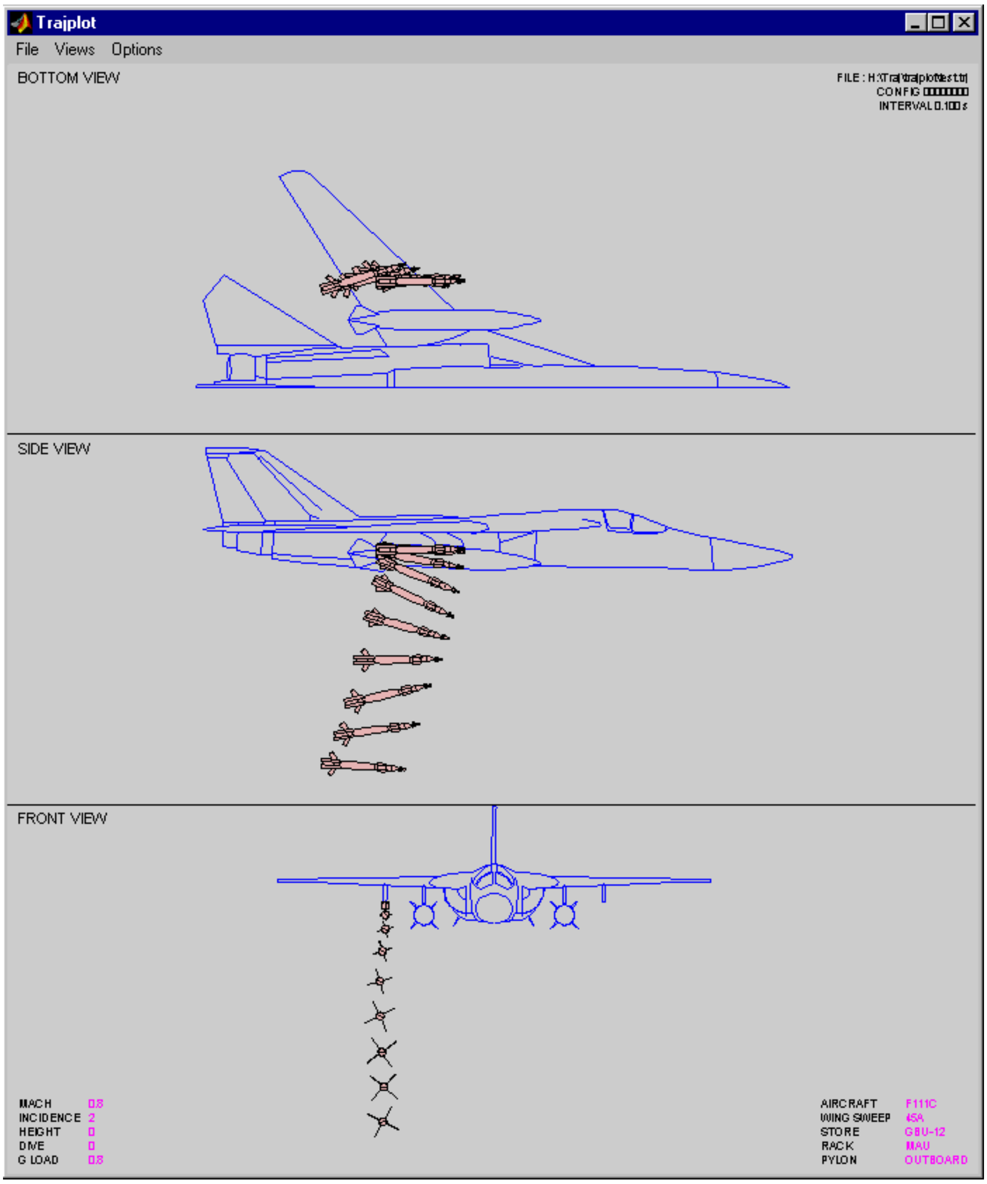
3.0 CAPABILITY ASSSESSMENT

The Australian DSTO has been collaborating with the USAF AFRL in developing MSTARS, a missile 6DOF fly-out and engagement modelling system (reference A). DSTO had developed a version of MSTARS using Matlab^{TM} and Simulink^{TM}. This example provided the impetus to assess them for store separation.

Towards the end of 1999 the potential of Matlab^{TM} / Simulink^{TM} was investigated by ASCENG. This first of all took the form of porting an existing complex application written in ‘C, and X-windows on a Unix workstation. The application ‘Trajplot’ was used to visualise store separation trajectories using 2D line drawings of aircraft and store from 3 viewpoints – front, side, and bottom. This 3 view was drawn on screen and printed out as a postscript file. The port to Matlab^{TM} on a windows PC proved to be straightforward, with the number of code lines reduced by a factor of five! The resulting Matlab^{TM} version can be seen in figure 1.

Further coding experiments were undertaken with GUI creation and to test the potential to draw 3D objects using the *patch* function and manipulating the view using the mouse. A major advantage with Matlab^{TM} is that it is widely used in the scientific, engineering and academic community world wide and there are many applications and example routines freely available from the Internet. Whilst formulating plans for the ASCENG store separation ‘toolkit’ several component routines were found to already exist – namely those relating to camera models and calibration. Several university groups have been working in the field of machine vision that required these capabilities for CCD video cameras. It required little effort to adapt and conduct verification and validation for some of this work to be applied to cine film cameras.

The assessments and experiments indicated that application of Matlab^{TM} and Simulink^{TM} to store separation was feasible, and may be easier than originally anticipated.



FILE : H:\Traj\trajplot\test.trj
CONFIG [illegible]
INTERVAL 0.100 s

MACH	0.8
INCIDENCE	2
HEIGHT	0
DIVE	0
G LOAD	0.8

AIRCRAFT	F111C
WING SWEEP	45A
STORE	GBU-12
RACK	MAU
PYLON	OUTBOARD

Figure 1: Screenshot of Matlab version of Trajplot showing F-111C with GBU-12

4.0 THE TOOLS

There are three main software tools that have been developed for the ASCENG separations toolbox – Traj3D, WSAS and STEME. Although Trajplot was initially developed as described above, it is no longer used as it is in effect superseded by Traj3D for separation trajectory visualisation. A description of each piece of software is given below.

4.1 Traj3D

Traj3d allows store separation trajectories to be visualised in 3D as an animation or an image time sequence (figure 2). The animation can be played through or stepped forward or backward by a user specified number of frames (that also applies to the image sequence). The 3D view is interactively manipulated using the mouse to set viewpoint and zoom.

The core of the Matlab™ 3D engine is OpenGL and so can take advantage of the now ubiquitous and cheap 3D graphics cards available for the PC due to the 3D games revolution of the late 1990's. The 3D acceleration provided by the cards means that complex geometries can be manipulated with ease.

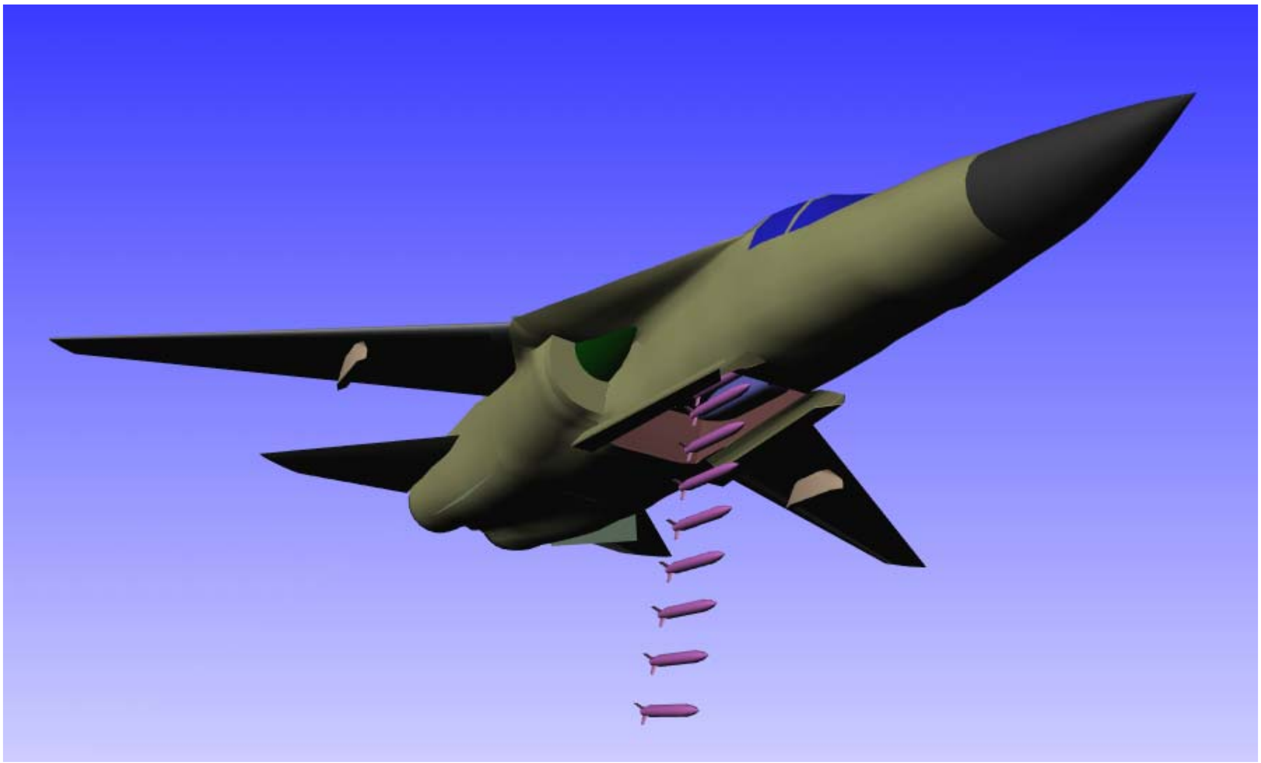
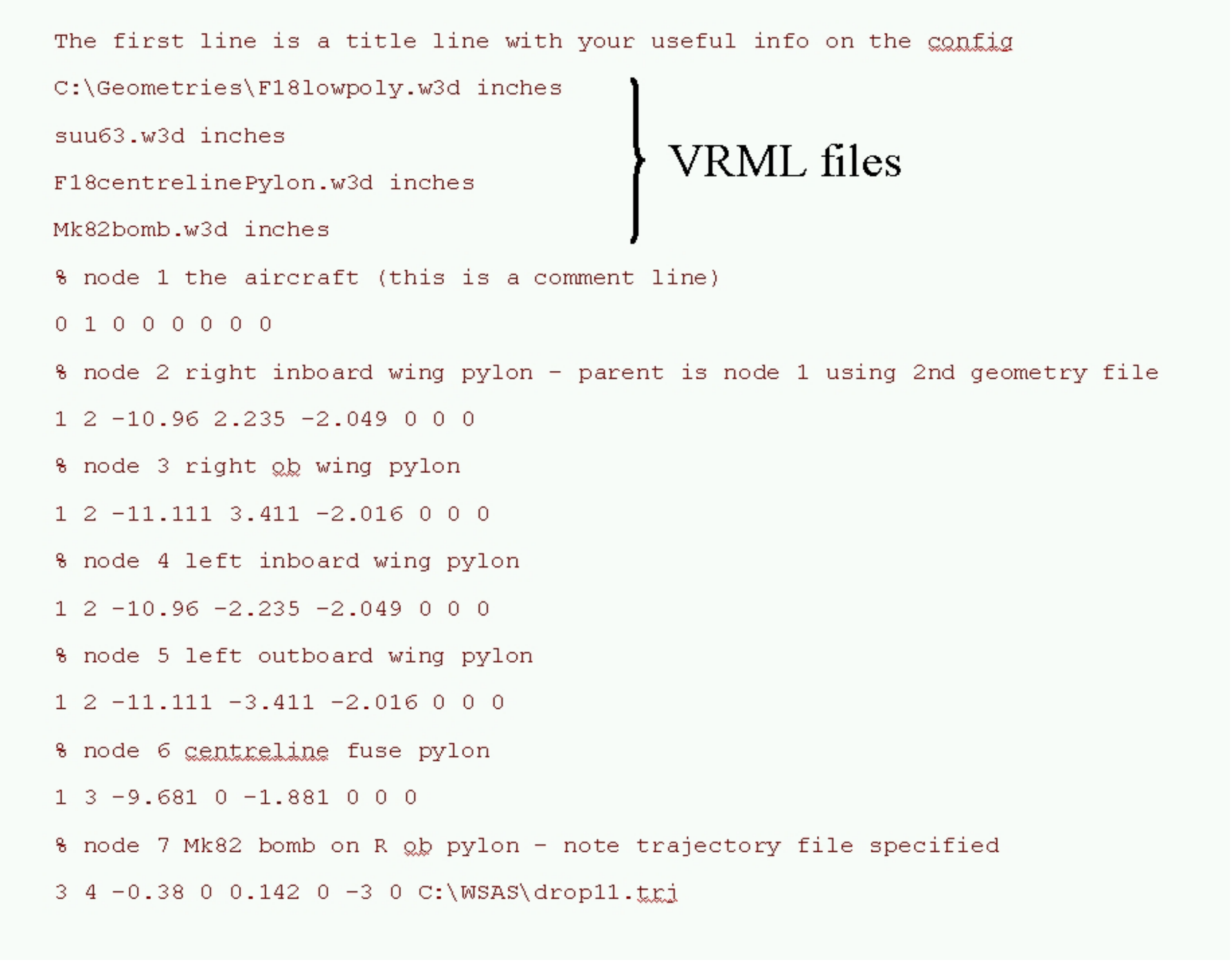


Figure 2: Traj3D trajectory image sequence for F-111 with Small Smart Bomb

The geometry format that defines a 3D object used in a scene is that defined by the VRML 97 specification for indexed face sets at reference B. This lists the 3D co-ordinates and then the connections between them to form the polygon facets of an object. The visual properties of the object are also specified, though only colour and transparency are used in Traj3D. A VRML file can be composed of several such objects. Many CAD and CFD packages support the VRML format, though it is also easy to create objects from dimensioned drawings using simple routines in Matlab™ eg axisymmetric bodies defined by a series of radii and their x location. An extensive library of aircraft (fixed wing & rotary) and stores now exists in this format for use at ASCENG.

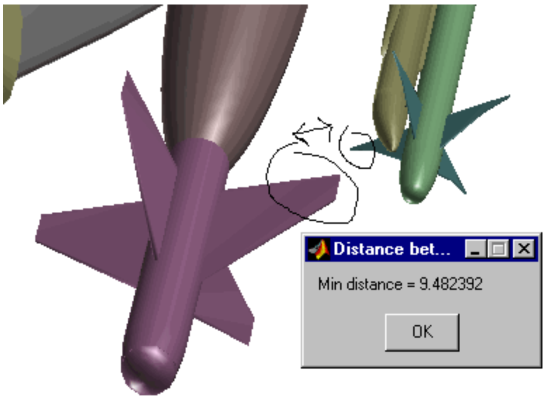
The objects in the VRML files are formed into a 3D scene in Matlab™ using a parent-child hierarchy as created by the user in a simple text file – the assembly file (figure 3). The file consists of a list of the geometry files to use and then how to connect them in the hierarchy. Each item in the hierarchy is termed a node. A node can be offset from it's parent in all 6 freedoms (x,y,z,phi,theta,psi). A trajectory file can be associated with each node. The 6 DOF values in the trajectory file are incremental offsets at each time point and are added to the fixed offsets. The geometry files are identified for each node by a number equivalent to the order that the geometry file is listed in the assembly file. Likewise the order number in which they are listed in the file identifies the nodes.



```
The first line is a title line with your useful info on the config
C:\Geometries\F18lowpoly.w3d inches
suu63.w3d inches
F18centrelinePylon.w3d inches
Mk82bomb.w3d inches
% node 1 the aircraft (this is a comment line)
0 1 0 0 0 0 0 0
% node 2 right inboard wing pylon - parent is node 1 using 2nd geometry file
1 2 -10.96 2.235 -2.049 0 0 0
% node 3 right ob wing pylon
1 2 -11.111 3.411 -2.016 0 0 0
% node 4 left inboard wing pylon
1 2 -10.96 -2.235 -2.049 0 0 0
% node 5 left outboard wing pylon
1 2 -11.111 -3.411 -2.016 0 0 0
% node 6 centreline fuse pylon
1 3 -9.681 0 -1.881 0 0 0
% node 7 Mk82 bomb on R ob pylon - note trajectory file specified
3 4 -0.38 0 0.142 0 -3 0 C:\WSAS\drop11.tri
```

Figure 3: Assembly file example

A complex scene can be easily set up, with multiple moving objects if required. This allows multiple store trajectories to be visualised each with flip out wings if necessary. It also allows Traj3D to be used as a virtual fit check tool to assess the viability of fitting new stores or existing stores in new configurations.



Min distance = 9.482392

Figure 4: Interactive Distance Calculation in Traj3D

To aid in this task there is a distance measure capability (figure 4) where the true 3D distance is calculated between two objects selected by the mouse. The virtual fit check capability has been extensively used, to consider new weapons for ADF procurement and even options for new aircraft (figure 5).

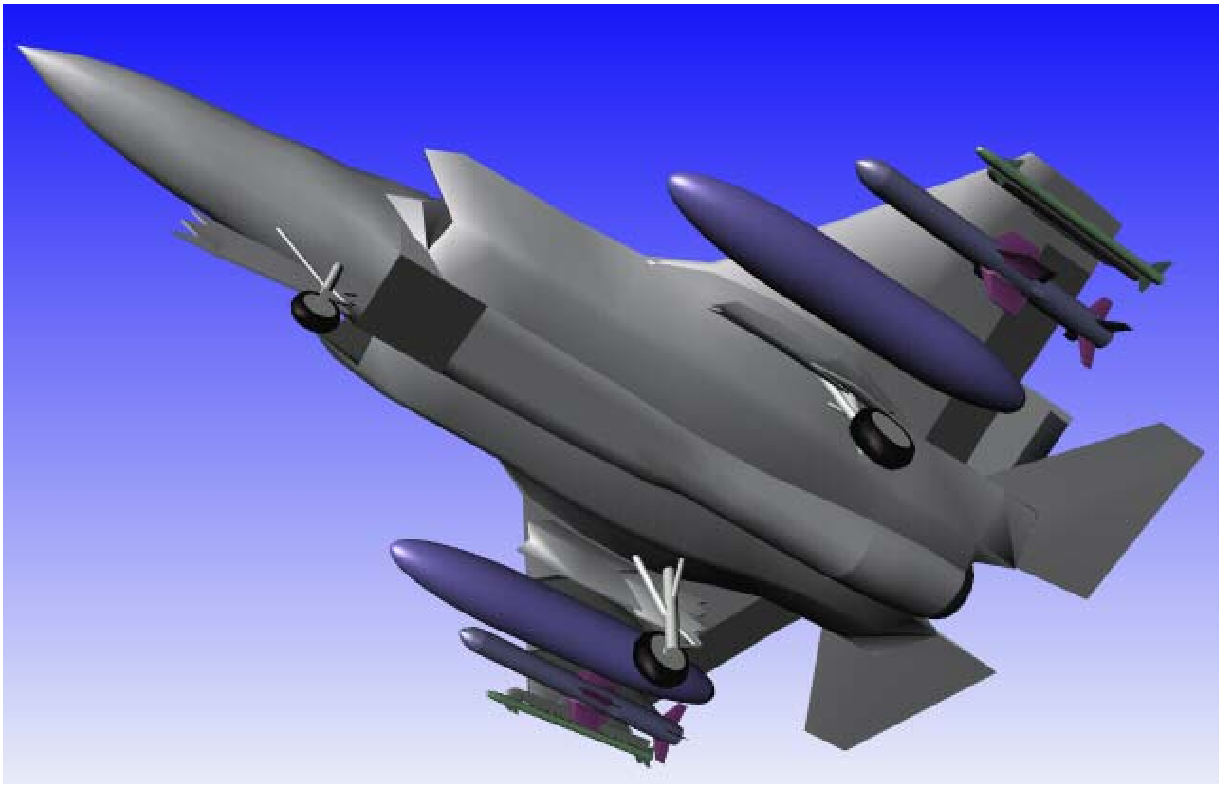


Figure 5: Example Virtual Fit Check in Traj3D

Particular view conditions can be saved and reloaded later and the completed scene saved as an image file, printed or saved as a VRML scene file. The animation can also be saved as an mpeg movie file.

4.2 WSAS

WSAS is a photogrammetric tool for determining the 6 DOF trajectory of a store using digitised cine film or high speed video of the separation event. The software uses an image matching technique, whereby 3D images of aircraft and separating store as viewed from a synthetic camera model are overlaid onto the film images. The film images are corrected for camera and lens distortion. The 3D models are manipulated interactively using the mouse on a frame by frame basis to build up the trajectory. This is a labour intensive process but can be surprisingly quick in practice, with a trajectory from one camera being generating within an hour. To speed up an analysis several people can be used, each analysing a different camera.

An alternative approach using the point tracking technique was considered, but ultimately rejected in favour of image matching, in part due to the synergy of using the same 3D ‘engine’ for trajectory visualisation in Traj3D. Point tracking offers the prospect of automation, but in practice it is semi-automatic at best, with software loosing track of the points and requiring frequent manual intervention. From discussions with operators of this technique the analysis can take considerably longer than the image matching method! Other factors in the decision were;-

- Considerable operational overhead in setting up and surveying the large number of tracking markers required on both aircraft and store.
- Difficulty in using method for small stores eg SRAAMs
- Requires good quality imagery.

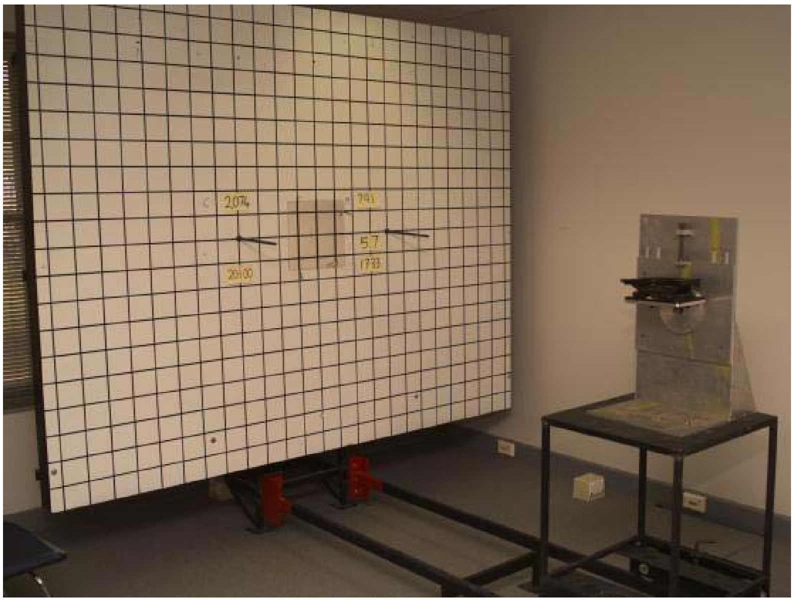
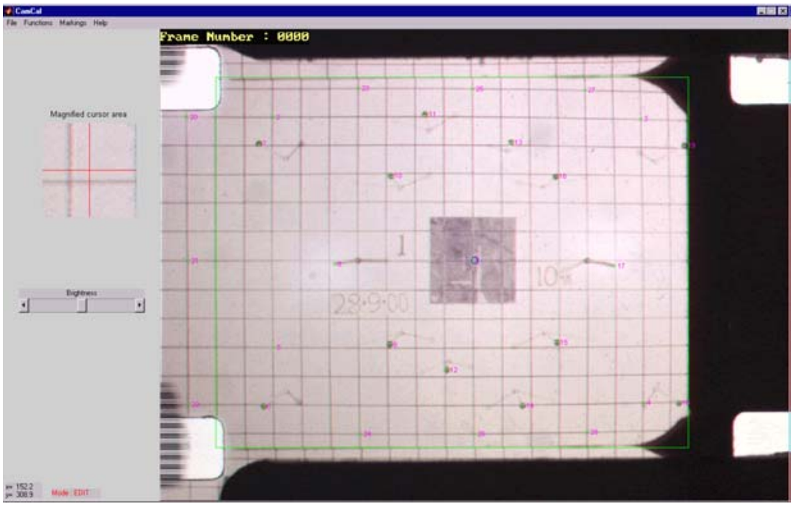


Figure 6: Camera Calibration Board

As stated previously, several ‘ready made’ camera models and calibration systems were already written in Matlab^{TM} by the machine vision community for CCD video. The camera calibration toolbox of Heikkila (reference C, D) was chosen. A GUI was written in MatlabTM to make operation of the calibration routines easier. The cameras and lenses to be used are logged by serial number and then calibrated from a few seconds of film of the calibration board (figure 6). Known points on the board are marked on the image in the calibration tool (figure 7).



Frame Number : 0000

x= 152.2
y= 308.9

Mode : EDIT

Figure 7: Camera Calibration GUI

The calibration method calculates effective focal length, image centre and radial and tangential distortion coefficients from which an inverse distortion can be produced. This information is then used in WSAS to correct the film images (figure 8).

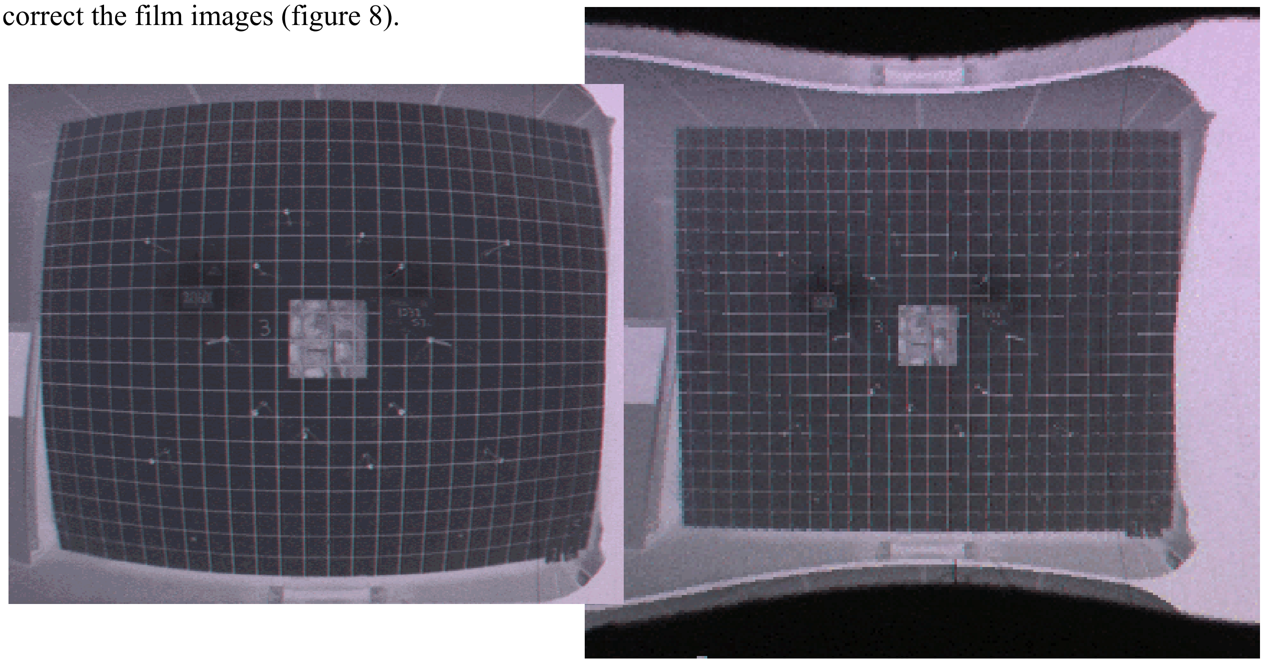
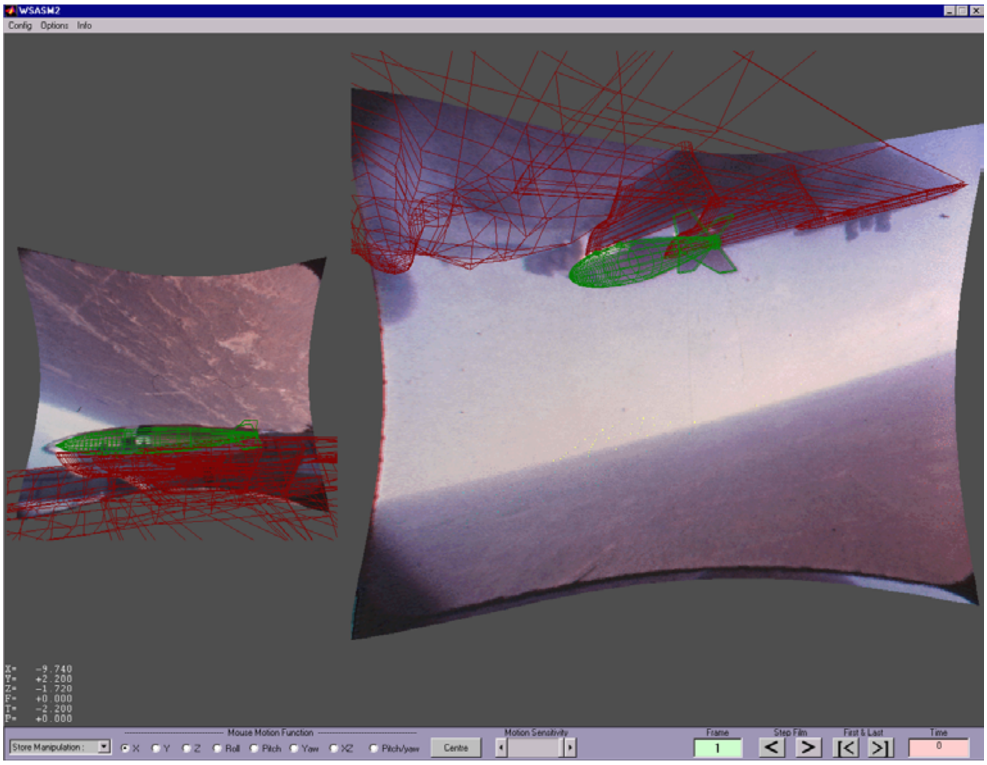


Figure 8: Image correction – original on left

Whilst only one camera can be analysed at a time, a second camera view can be used as a cross check to help the analysis (figure 9).



```
X=    -9.740
Y=    +2.200
Z=    -1.720
F=    +0.000
T=    -2.200
P=    +0.000
```

Frame: 1

Time: 0

Figure 9: WSAS Screenshot showing 2nd camera view

A 6dof graph of the trajectory can be viewed at anytime to check progress and reduce analysis errors, the current film frame point being highlighted. Whilst an analysis can be achieved with one camera, it is improved with several cameras using different views. The trajectory information derived from each camera can then be combined using a 'view angle weighted' least squares technique. A different weighting is applied to each degree of freedom dependent on the camera view of the motion, eg a camera looking side on to a bomb separation will get good information for x, z and pitch angle but be poor for y, and yaw.

It is often useful to still look at the raw data from each camera as this can give an indication of the analysis accuracy. Although ground tests have shown a system accuracy of about 50mm in position and 1 degree rotation, the true accuracy of a flight analysis can be considerably degraded. It varies dependent on size of store, image quality, calibration, operator skill and camera view angles that can be far from ideal.

4.3 STEME

STEME (pronounced 'steam') is an acronym for Store Trajectory Estimation in a Matlab™ Environment. It is a 6 DOF trajectory prediction code for aircraft stores separation simulation, leveraging the capabilities inherent in Matlab™ and Simulink™. The idea was conceived to make use of the latter's ability to graphically model a missile autopilot due to the fact that many modern missiles now use an autopilot to aid separation. Although there are examples of full aircraft simulations and missile free flight simulations built graphically in Simulink™, the store separation problem is considerably more complex, especially the implementation of release device constraints[1]. As a result the decision was taken to create the simulation system as an S-function block that calls code written by the user. There is a choice of languages available for this, but Matlab™ is optimal. The new function block is the 'STEME engine' (figure 10).

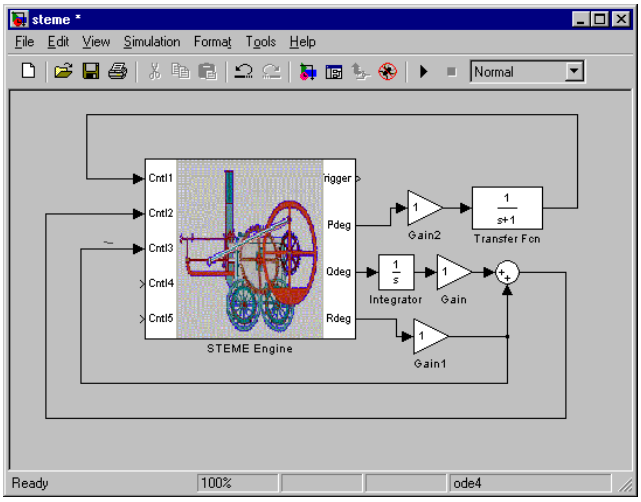


Figure 10: The STEME engine block in Simulink with example autopilot

The Matlab™ code of the block reads a text input file describing the model, and then any look up tables specified in the input file before calculating the store motion using Simulink to integrate the required variables. Store rotations are managed using the quaternion method (reference E) so that a full 360° sphere of rotation is possible. This allows simulation of multi-axis tumbling motions common in the jettison of unstable stores.

The launch aircraft motion is described by a series of idealised manoeuvres, such as straight line, circular pull up or bank turn. This is possible because in the majority of separations the aircraft motion is little affected during release. However it would be easy to include fully coupled aircraft motion in future if

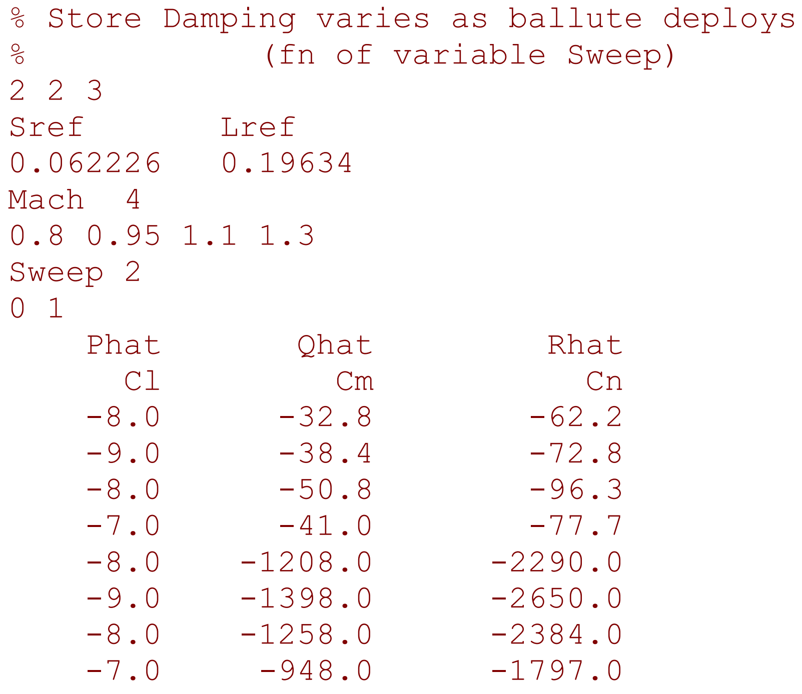
[1] STEME was developed before the SimMechanics toolbox made it easy !!

required. In addition to the equation driven motion, the aircraft motion can also be read from a look-up table generated from a real flight manoeuvre or flight simulator output. As quaternions are used to define rotations, the time points of such a file can be easily interpolated.

Three release device options are available – Ejection Release Unit, Rail launcher and Hook. Each device constrains the separation motion and in the case of the ejection release unit (ERU) also applies ejection thrusts. The constraint also changes as the separation event progresses, for example a missile can only move along a rail while two hangers are engaged, but may pitch and/or yaw about the last hanger when the penultimate has slid out of the rail. In all cases a generic method is used where the store is constrained at up to two locations, with momenta and reaction forces calculated to achieve the required behaviour considering all forces acting and aircraft motion.

The store mass, inertia and centre of gravity location can all be time varying to account for changes due to rocket motor burn or geometry changes such as flip out wings. A lanyard option is available such that a pre-set distance between defined points on aircraft and store can be used to trigger events such as rocket ignition, flip out wings or autopilot timers. Any number of thrusts acting at any point or direction can be applied from look up tables. Ejector thrusts are treated separately, as they have to be switched off at the end of the ram strokes.

The look-up tables used throughout STEME can be up to 7 dimensional and are stored in memory as self contained structures with the text file format reflecting the structure (figure 11).



```
% Store Damping varies as ballute deploys
%            (fn of variable Sweep)
2 2 3
Sref       Lref
0.062226   0.19634
Mach   4
0.8 0.95 1.1 1.3
Sweep 2
0 1
    Phat        Qhat          Rhat
      Cl          Cm            Cn
    -8.0       -32.8         -62.2
    -9.0       -38.4         -72.8
    -8.0       -50.8         -96.3
    -7.0       -41.0         -77.7
    -8.0     -1208.0       -2290.0
    -9.0     -1398.0       -2650.0
    -8.0     -1258.0       -2384.0
    -7.0      -948.0       -1797.0
```

Figure 11: Example Data Structure file

As well as the names of independent variables and dependent variables the structure also allows for condition variables eg reference length, point of action that apply to the data. Post multiplier variables are used to factor the output. The data is usually linearly interpolated, but other options are possible. If a dependent variable does not already exist it is created in the workspace. In this way a general data structure can be used to create new variables in the system for further use as variables in other data structures. New variables can also be created by simple Matlab™ equations at the end of the input file which are then executed at each integration step. The combination gives a powerful and versatile means of creating complex models.

As data structures are self contained they can be set to use local flow conditions for isolated aerodynamics. This allows a store to be defined as individual components eg nose, body, tail that each have an aerodynamic description and use the local flow angles when a flowfield is defined (figure 12). A flowfield is a data structure defining local flow perturbation velocity components and air density at a grid of points around the aircraft – usually generated from CFD.

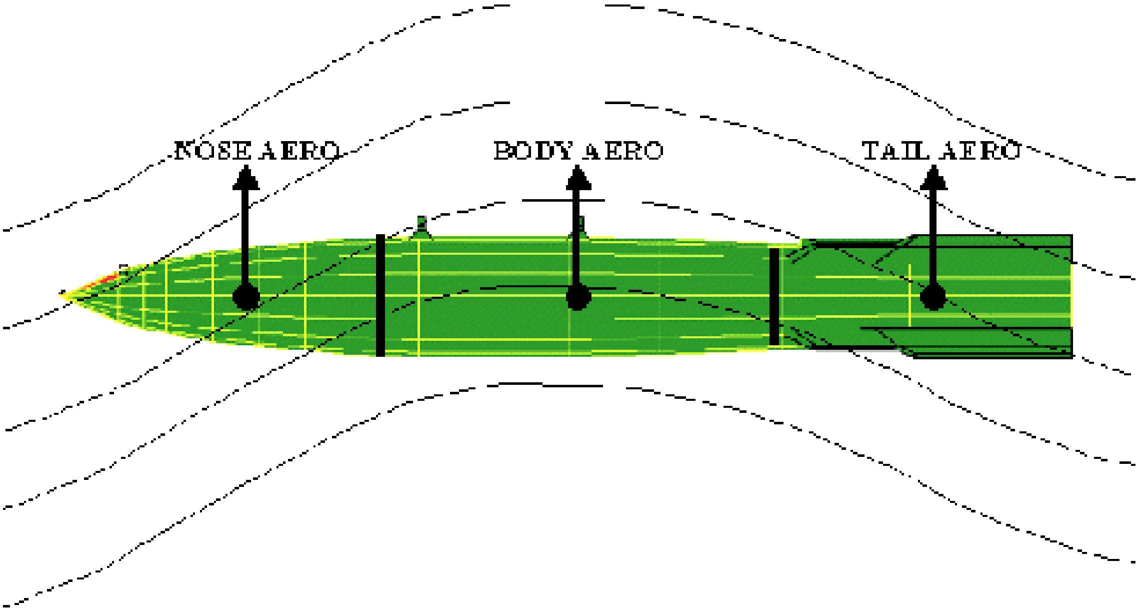


Figure 12: Flowfield and component aerodynamics

The aerodynamic effect of the aircraft on the store is modelled by the inclusion of data structures for carriage loads, grid loads or a combination of them. The loads data can be used directly or more usually an interference load is calculated. This is the difference between the load derived from the wind tunnel and the load calculated from the isolated aerodynamic data structures including flowfield effect if present with the store at equivalent conditions ie same grid load position, orientation, Mach, alpha etc. The interference load is calculated in the initialisation step and then the interference load data interpolated and the result added onto the free air calculations as the trajectory progresses. The interference load is decayed over a distance once outside the grid with a user set decay law – usually linear.

The user can specify any variable in the Matlab™ workspace to be printed to screen as the trajectory calculation progresses or written to any number of user named output files. This allows easy coupling with Traj3D for visualisation eg store trajectory can be saved to one file whilst a wing opening trajectory saved to another for later animation of the separation including wing deployment.

Whilst the input file to the STEME engine block can be defined in any text editor and the model executed from Simulink, a GUI has been written to make operation easier (figure 13). The GUI has a series of pop up panels for data entry, with many options chosen from drop down lists. Many panels have areas that allow data input conditional on options selected eg only ERU items are active in the release device panel if that is the chosen option. The run button executes the model in the background, such that the Simulink model diagram window does not appear. Help is available in the form of a HTML user guide system.

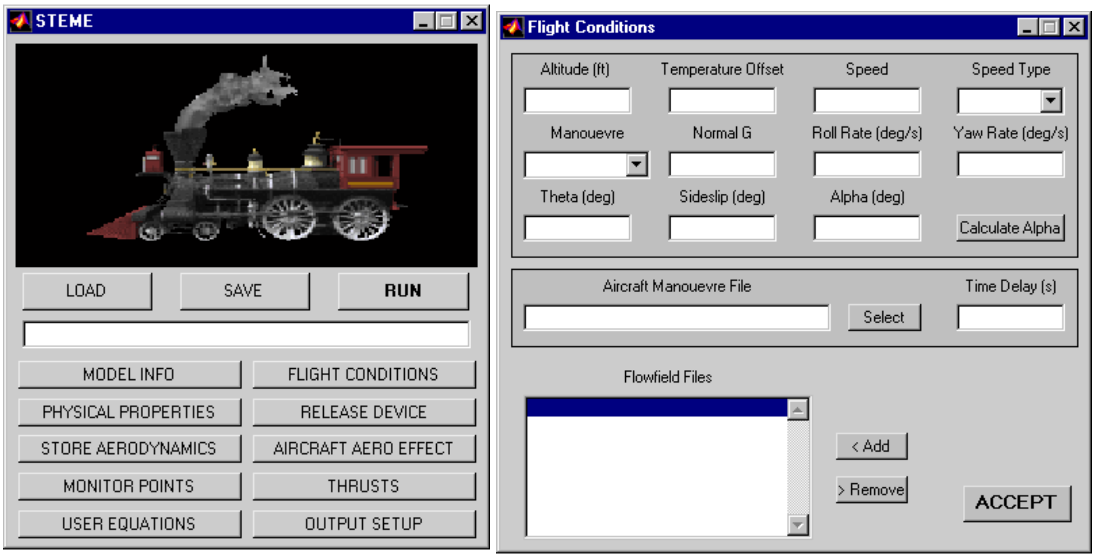


Figure 13: STEME User Interface examples

The STEME engine provides a versatile and comprehensive store separation modelling capability in only 2000 lines of Matlab™ code. It is easy to use, highly configurable without having to reprogram and has a very easy way to model separation autopilots. Programming, verification testing, user guide and documentation of the engine and GUI were completed within three months by one engineer!

5.0 FURTHER DEVELOPMENTS

Several developments of the toolkit have been envisioned. Some are well underway, whilst others are still at the 'pipe dream' stage:

- A batch run / Monte Carlo system.
- Tow cable simulation.
- Parameter ID system to 'match' model to real data.
- Multiple store release capability.
- End to end target engagement and munitions effectiveness simulation.
- Fuel slosh effect on external tank jettison.

The batch run system is almost complete. From a simple text file specifying input variables to change and lists of their values, simulations of all the combinations can be produced. Alternatively, Monte Carlo cases can be produced for the variables. The facility is to be developed to produce cases to generate down to ground simulations for flyout model and safety template generation (reference F), including ricochet using the Barnes-Birkoff method (reference G).

A tow cable capability is also near completion, only requiring good validation data. This is again implemented as a Simulink™ S-function block, allowing tow cable end motions to be specified outside the block (figure 14,15). The cable is modelled as a series of linked springs with aerodynamic cross flow and skin friction drag (any number of segments). Burnham-Hallock (reference H) wing tip vortex effects can be specified, as for low mass towed objects the aircraft wake interaction with the tow cable can be significant.

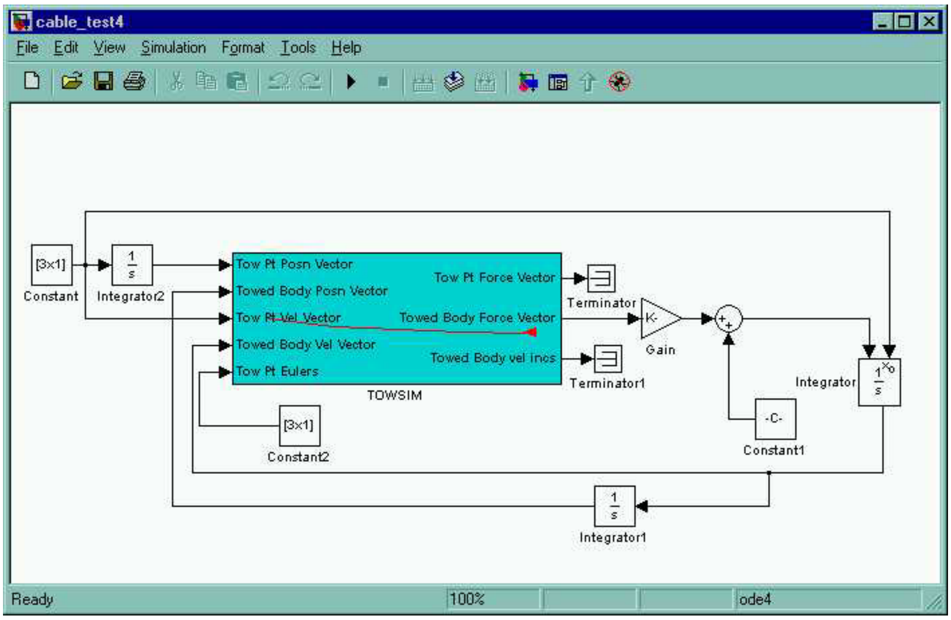


Figure 14: Tow cable simulation block in Simulink

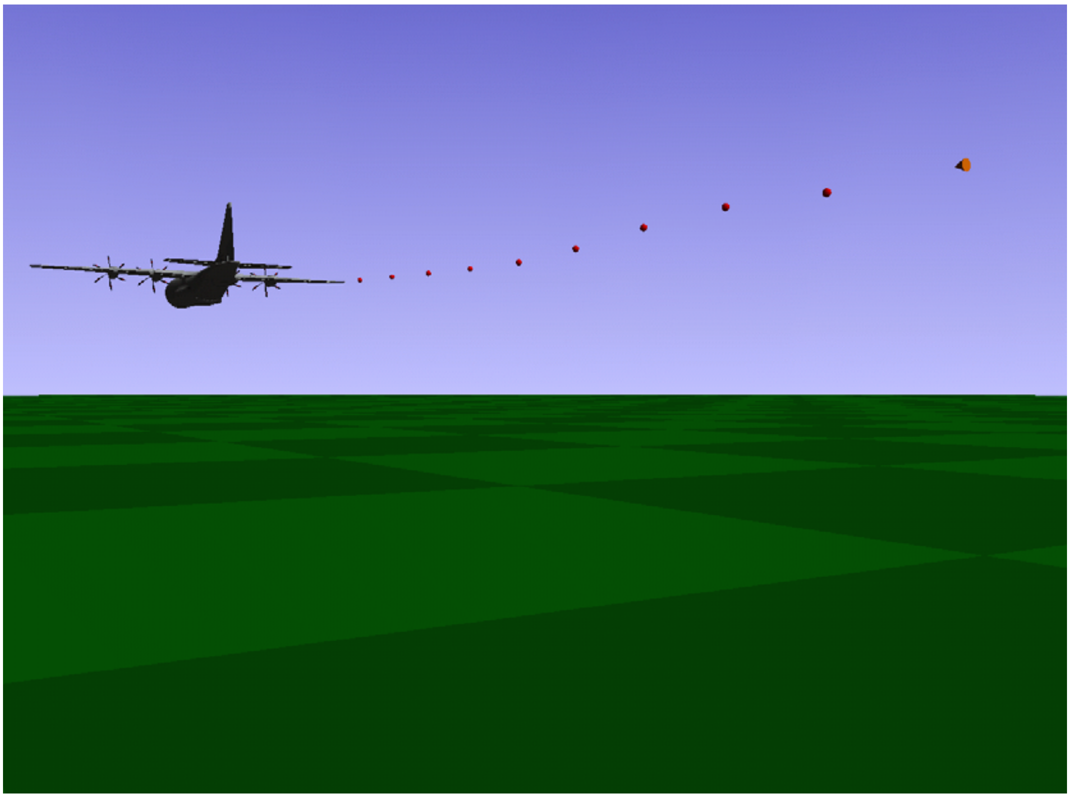


Figure 15: 3D Visualisation of Tow cable simulation

The parameter ID capability would use the batch run technology and the existing optimisation toolbox for Simulink^{TM}. Multi-store simulation should only require several instances of the STEME engine in the Simulink^{TM} diagram with links to work out interactions. Simulation of the engagement scenario requires the full guidance autopilot description, target motion and seeker model. These could be included from the Simulink^{TM} implementation of MSTARS, which currently ignores release disturbance effects, which can have a significant impact on overall effectiveness!

5.0 CONCLUSION

The commercial modelling and simulation package of Matlab^{TM} / Simulink^{TM} has matured into a highly capable development environment for engineering software. Complex and professional looking software can be easily and quickly developed to meet engineering simulation requirements. The resulting software is significantly more compact in terms of lines of code and so easier to understand and maintain than traditional software languages. This also aids in verification and validation.

Matlab^{TM} / Simulink^{TM} has proved to be well suited for use in store separation, with a full capability toolkit being developed in record time. Traj3D gives excellent 3D visualisation of separation trajectories and has proved invaluable for assessing new weapons fit options. WSAS is a low overhead, fast and easy to use tool for deriving the trajectory of a separating store from cine film or high speed video. STEME has been developed as a versatile and powerful 6DOF separation modelling tool, ideal for modern separation autopilot simulations.

6.0 ACKNOWLEDGEMENT

The author would like to thank the US Navy Office of Naval Research (ONRIFO) for their generous support through their visitor support programme.

7.0 REFERENCES

[A] 'Munition Simulation Tools And RecourceS', www.mn.afrl.af.mil/public/mng/mstars.htm.

[B] Web3D organisation, (1997), 'Virtual Reality Modelling Language Specification', ISO/IEC 14772-1.

[C] Janne Heikkila, (2000), 'Camera Calibration Toolbox for Matlab'. University of Oulu. www.ee.oulu.fi/~jth/calibr/

[D] Janne Heikkila & Olli Silven (1997) 'A Four-step Camera Calibration Procedure with Implicit Image Correction'. University of Oulu, Finland.

[E] Joseph M. Cook, Michael J. Zyda, David R. Pratt, Robert B. McGhee, (1994) 'NPSNET: Flight Simulation Dynamic Modelling using Quaternions', US Navy Postgraduate School, Monterey, California.

[F] Dr Ming Chao, Graham Akroyd, (Oct 2003) 'Verification and Validation Process of the Australian Safe Escape Methodology', 6th Australian Explosive Ordnance Symposium, Canberra.

[G] W. W. Barnes, (1988) 'An approach to the prediction of the ricochet of projectiles from land surfaces', Australian Department of Defence – Central Studies Branch.

[H] J. N. Hallock, (1991), 'Aircraft Wake Vortices: An assessment of the current situation' DOT-FAA-RD-90-29.

DISCUSSION EDITING

Paper No. 17: Weapon Separation analysis tools using matlab

Authors: Graham Akroyd

Speaker: s.a.

Discussor: Nafiz Alemdaroglu

Question: 1. The interaction between the flow field and the trajectory calculation is not well understood. Matlab can be used to formulate the 6DOF trajectory. However you need the flow field data?.

2. How do you get the flow field? Where do you get it from.?

3. Interaction with flow solver and matlab is through a look up table?.

Speaker's Reply: The flow field is calculated using CFD and included as a lookup table.

However it is not essential to have a flow field as the main way of applying the aerodynamic effect of the aircraft is through wind tunnel measured grid loads. The use of a flow field and multiple reference point free stream aerodynamics for the store is to improve the interpolation/ extrapolation from the grid conditions and so reduce the amount of wind tunnel data needed by using a much coarser grid.

Discussor: Fredrik Ljungberg

Question: Is a down-the-rail model included in your launch model?

Speaker's Reply: Yes, all release systems are modelled, i.e. rail, eject and eject/hook.