

# Towards a Generic Architecture for Autonomous Landing Systems<sup>1</sup>

**James Goodwin, Alan Winfield, Quan Zhu**

Intelligent Autonomous Systems Laboratory,  
University of the West of England, Bristol,  
Coldharbour Lane, Bristol BS16 1QY, UK

[james.goodwin@sea.co.uk](mailto:james.goodwin@sea.co.uk), [{Alan.Winfield|Quan.Zhu}@uwe.ac.uk](mailto:{Alan.Winfield|Quan.Zhu}@uwe.ac.uk)

## Abstract

This paper outlines the design of a 2D analogue for autonomous landing on an asteroid as a hardware demonstrator/development platform, within an overall requirements structure that aims to develop a generic architecture for autonomous landing systems. Although wheeled laboratory robots may appear to be unlikely analogues for asteroid landing, the system is already proving useful as a platform for exploring and testing aspects of autonomy within the mission profile. Real-world sensor noise and inaccuracy as well as real-time issues are testing the system in a way that does not emerge from simulation alone.

Keywords: generic hybrid architecture, autonomous unmanned vehicle, small body (comet/asteroid) landing, real-time control

## 1. Introduction

The phrase ‘Autonomous Landing System’ is used as a broad term to refer to the autonomous control of a vehicle from a 6 degrees of freedom (DOF) unconstrained or flying state to a stable landing state or docking point. The term ‘vehicle’ refers to any aerial, underwater or space robot with up to 6 DOF and little or no influence from a ‘man in the loop’.

We intend to employ a methodology in which algorithms can be implemented and tested on a development platform that provides an analogue of the target platform. Clearly, for such a methodology to be successful, the development platform needs to model, as faithfully as possible, the dynamics of the target and much effort is being expended on accurately mapping the characteristics (sensors,

actuators and dynamics) of development and target vehicles.

We have a particular interest in small planetary body (comet/asteroid) landing, not least because of the special challenges of landing on a small body with chaotic motion and irregular shape. The fact that such bodies cannot typically be mapped from earth more or less demands an autonomous landing system capable of mapping the object, selecting a suitable landing site and then planning and executing the landing. This paper describes the design of a simple 2D laboratory-analogue for small planetary body landing, using wheeled mobile robots, developed with the aim of testing and demonstrating autonomous planetary landing.

This paper proceeds as follows. Section 2 describes the background to this work by first outlining small body landing then, secondly, surveying characteristics of generic robotic architectures. Section 3 then outlines the overall methodology proposed for requirements and design with a view to maximising the genericity of the architecture. Section 4 introduces the 2D laboratory-demonstrator and development platform with reference to the mission phases during landing. Finally in section 5 the paper draws some initial conclusions.

## 2. Background

### 2.1 *Small Body Landing*

The majority of spacecraft are developed using highly conservative approaches. However, despite its conservatism, autonomy and autonomous systems are of increasing interest to the space community as they alleviate the necessity for ground control and reduce

---

<sup>1</sup> This research is supported by the TCD and Systems Engineering & Assessment Ltd.

the problems associated with occlusion and communications delay. The European Space Agency (ESA) is directing its efforts in line with the recently formulated AURORA programme, which is “envisioned as having two goals: searching for traces of extraterrestrial life, such as fossil microbes on Mars, and precursor molecules to life on Earth; and laying the groundwork for future human space exploration”. In this category on-board autonomy has been identified as a key technology (Bornscheigl, 2002).

Small body rendezvous is also becoming an interesting area of exploration. These small bodies provide a difficult environment in which to undertake sample and return exercises, due to their irregular shape, composition and chaotic rotation. The additional difficulty of mapping the objects mean that automated landing systems are highly desirable, and possibly essential.

Comet and asteroid rendezvous has a very limited history. The 1996 NEAR mission was the first spacecraft to actually touchdown on the surface on an asteroid (Yeomans, 1995). Rosetta is planned as the first comet landing and is expected to be launched later in 2003, reaching its destination around 2012. Both of these vehicles use limited on-board processing, requiring their control commands to be up-loaded from an earth-based ground-station. Muses-C is the only mission to an asteroid employing autonomous navigation and landing planned for launch in the near future (NASA, 2002). With such a small exploration history and poor knowledge of the composition of such bodies, it is impossible to analyse the effectiveness of different approaches to small body landing based upon previous experience.

## 2.2 *Robotic Architectures*

The primary motivation of this work is to develop a generic autonomous landing system, which could be implemented on a variety of different vehicles. Clearly the choice of architectural design is crucial to ensuring that the system is non-platform specific. Such a design should, however, be capable of implementation on both laboratory-based systems and target vehicles.

Robot control architectures broadly fall into two categories, behaviour-based and deliberative. Behaviour-based systems may provide quick reactions to stimuli, but are generally incapable of planning their actions. In contrast, deliberative architectures may be tailored to produce optimal

results based upon system constraints and a model of their environment. Combining these abilities leads to the development of hybrid architectures. However, it is difficult to implement a hybrid architecture that is entirely independent of its vehicle. Our current work is therefore focused on developing a generic hybrid architecture, which may be implemented on a variety of vehicles. This work necessarily involves designing an interface to link the vehicle to its control architecture. Robin Murphy (Murphy, 2000) and Ronald Arkin (Arkin, 1998) outline some of the characteristics for a robotic architecture. These characteristics have usefully been combined in the following list.

1. **SUPPORT FOR PARELLELISM**  
Reactive and deliberative architectures both require some form of parallel behaviour, where a number of processes must be performed simultaneously or at least within their real-time constraints.
2. **HARDWARE TARGETABILITY**  
How can the architecture be implemented on the robot hardware?
3. **MODULARITY**  
Does it show good software engineering principles? Modularity supports software re-use and abstraction from errors.
4. **NICHE TARGETABILITY**  
How well does it work for the intended application?
5. **PORTABILITY**  
How well would it work for other applications or other robots?
6. **ROBUSTNESS**  
What mechanisms does the architecture provide for fault tolerance?
7. **RUN TIME FLEXIBILITY**  
How can the system be adjusted during operation?
8. **PERFORMANCE EFFECTIVNESS**  
How well does the robot perform its intended task(s)? Measures such as completion time, energy use and minimum distance travelled are all appropriate metrics.

These characteristics will be used to qualify the candidate architectures.

## 3. **Generic Requirements Structure**

The Requirements Structure in Figure 1 below illustrates the breakdown of components for the autonomous landing system. The term ‘development platform’, on the left-hand side of the diagram, refers

to the robot(s) available in the laboratory, for instance the LinuxBot. The development platform will include both physical demonstrator and simulation. The term ‘target platform’, on the right-hand side of the diagram, refers to the real-world vehicle for the autonomous landing system, such as a planetary lander or UUV. The target platform similarly includes simulation. The purpose of Figure 1 is to diagrammatically link target and demonstrator (or development) systems, and their simulators. The box labelled ‘Common Software Components’ at the foot of the diagram thus contains the generic parts of the autonomous landing system. The diagram suggests

that by maximising the number of common software components, and by moving successfully from simulation to demonstrator on the development platform (left-hand side), then we significantly increase the confidence of transferring successfully from simulation to target vehicle (on the right-hand side).

As part of a generic architecture common software components are likely to include, for instance, Obstacle Avoidance, Path Planning, Path Re-Planning, Localisation/Map-Building and Landing Site Selection.

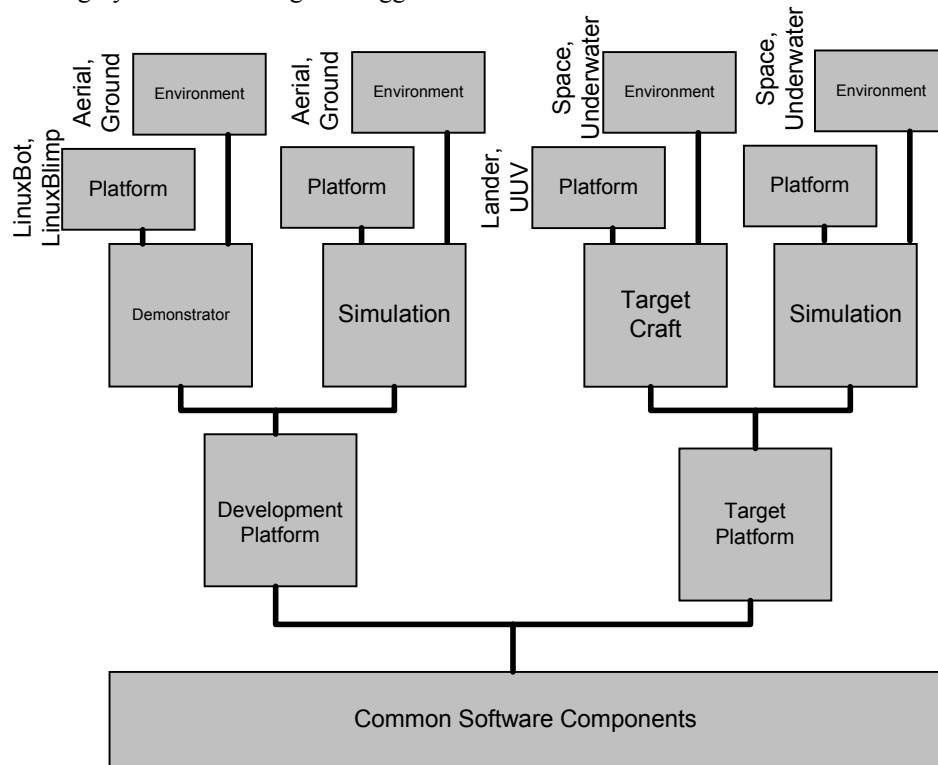


Figure 1 Requirements Structure

#### 4. Demonstrator And Target Platform

Developed at UWE Bristol’s Intelligent Autonomous Systems (IAS) Laboratory for conducting experiments in collective mobile robotics, the LinuxBot is a proven and reliable wireless-networked wheeled mobile robot capable of supporting a wide range of sensors and actuators. The robot has a Linux-based software and communications architecture, which embeds TCP/IP networking tools including FTP, Telnet and Web servers (Winfield and Holland, 2000). Figure 2 shows a photograph of the LinuxBot demonstrator/development platform. The LinuxBot in figure 2, fitted with a range of sensors

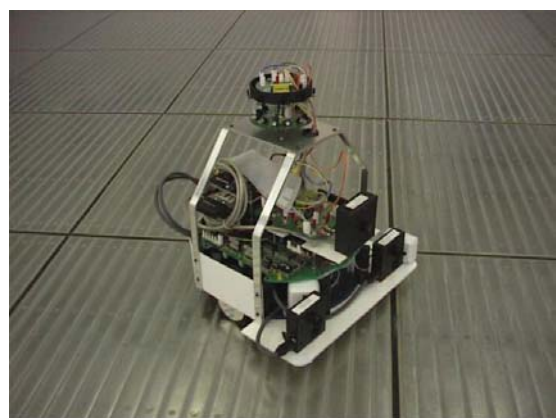


Figure 2 LinuxBot Lander-Analogue

including vision, IR-based collision detection and an IR-based localisation system, is used as the planetary lander analogue. A second LinuxBot is also employed at the heart of the asteroid/comet analogue.

**Figure 3** is a photograph of the LinuxBot fitted with the 'asteroid'. Thus the asteroid analogue may be programmed to imitate the motion of a small body, whose spin axis may not remain static. The shape of

the asteroid analogue is modelled on the near-earth object 216 Kleopatra (NASA, 2003).

The model includes IR landing point beacons and will be further developed to include surface features. This will enable the lander to determine the most suitable landing point based upon the surface features and dynamical constraints.



**Figure 3 Lander-analogue orbiting the KABot (Kleopatra Asteroid-analogue)**

A mission profile for a comet/asteroid landing generally follows the sequence below. Those phases implemented in the demonstrator are highlighted in bold.

1. Ballistic Arrival
2. Orbit Insertion
3. Orbit Optimisation
- 4. Mapping Phase**
5. Gravity Field Determination
- 6. Landing Site Selection**  
Requires path planning, constrained by resource utilisation.
- 7. Orbit Descent Manoeuvre**  
Attempt to enter a geostationary-like orbit, following the surface of the body.

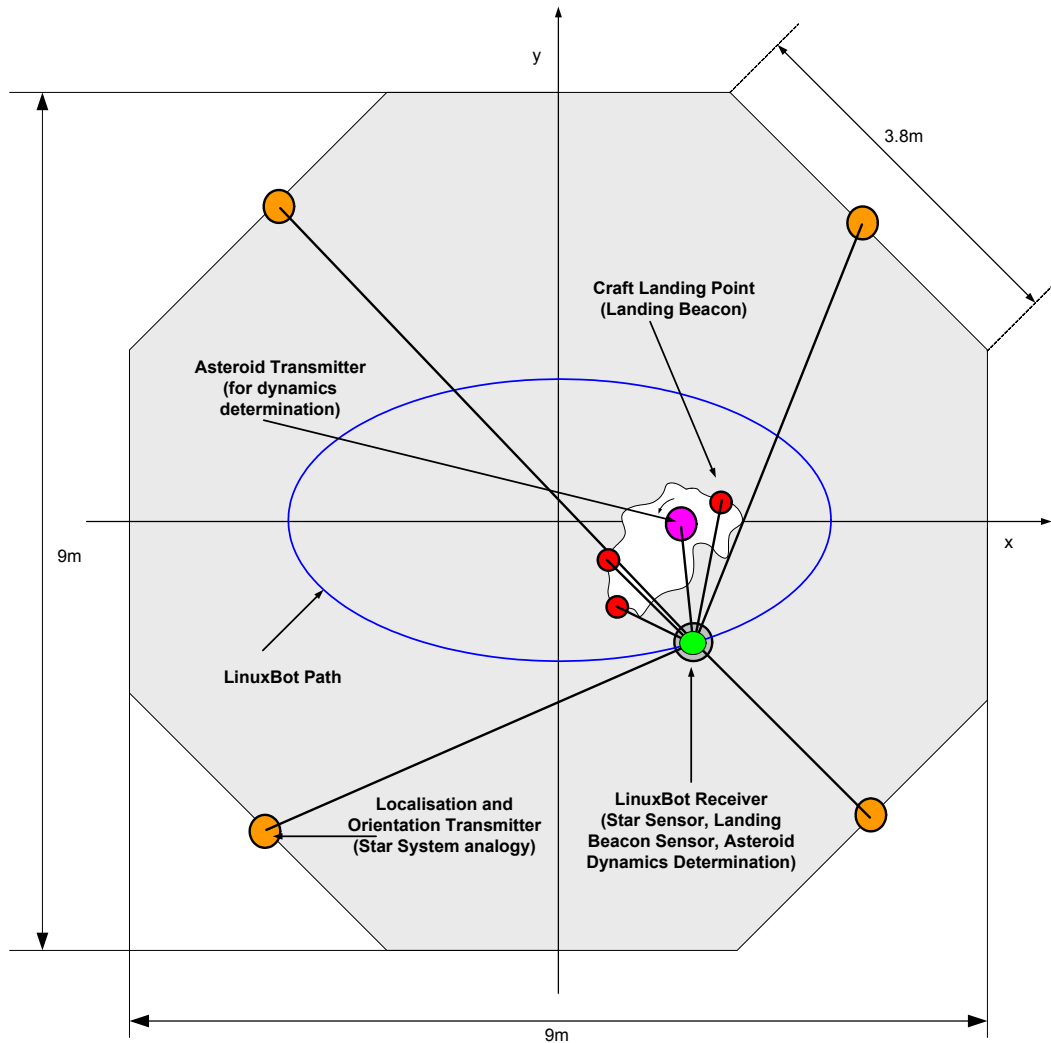
**8. De-orbit Burn**

**9. Active controlled Descent**

Key issues are: Closed Loop; Re-planning/decision making; Health Monitoring/Survivability. Thrusters may be assumed to be either constantly on or off. Pulsing of thrusters is appropriate. Variable thrust is possible but expensive. Variable thrusters were used on the Apollo landings, which also utilised abort procedures.

**10. Terminal landing**

11. Landing
12. Anchoring
13. Surface Operation
14. Take-off from the surface.



**Figure 4 Demonstration Setup**

Figure 4 above shows the laboratory demonstrator/development platform for the small body (asteroid) landing analogue. The LinuxBot lander-analogue is shown describing a Keplerian elliptical orbit around the rotating asteroid-analogue. Four fixed IR- beacons are placed at the edges of the experimental arena, as fixed-star-analogues and enable the LinuxBot lander-analogue to localise itself and make orbital corrections. Whilst orbiting, the lander-analogue will undertake mission phases 4, mapping, and 6, landing-site selection. The IR-

beacons on the asteroid-analogue will be used by the lander-analogue to determine the asteroid's orbital dynamics, and to select a landing site (later it is hoped to use a largely vision-based system instead). Having determined the orbital dynamics and chosen a landing-site the lander-analogue will then plan and execute stages 7-10, matching velocity and then executing a controlled 'descent' to the landing site, iteratively re-planning as it descends to the surface of the asteroid-analogue.

## 5. Conclusions

This paper has outlined the design of a 2D analogue for autonomous landing on an asteroid as a hardware demonstrator/development platform, within an

overall requirements structure that aims to develop a generic architecture for autonomous landing systems. Although wheeled laboratory robots may appear to be unlikely analogues for asteroid landing, the system is already proving useful as a platform for exploring and testing aspects of autonomy within the mission

profile. Real-world sensor noise and inaccuracy as well as real-time issues are testing the system in a way that does not emerge from simulation alone.

The most difficult and crucial part of this work will involve designing the control system architecture. Unlike many systems, the vehicle is influenced by a variety of stimuli that it must process in order to provide smooth and accurate real-time behaviour using both high-level reasoning and reactive responses. It is difficult to predict the final configuration of such an architecture. However, initial indications are that it is likely to be a form of hybrid architecture, capable of both deliberative and reactive behaviours and operating within real-time constraints. Owing to the necessity for the system to be generic, layers of abstraction must be built-in. We have identified architectures that have a mixture of these capabilities. A number of hybrid architectures for autonomous flight control have been developed and implemented for UAV and UUV operation, including for instance Georgia Tech's Open Control Platform (Kannan et al, 1998), the University of Texas Strategic Tactical Execution Software Control Architecture STESCA (Nelson et al, 1999) and the University of Hawaii Intelligent Task-Oriented Control Architecture (Ridao et al, 2000). However, a generic, hybrid architecture for autonomous landing systems has hitherto not been identified or realised. Such an architecture has the potential to impact upon control systems for a wide variety of vehicles, reducing the repetition of software development and providing control which may be implemented with a minimum of platform specific adaptation.

### Acknowledgements

The work is primarily funded as part of a TCS programme into Autonomous Landing Systems. The TCS programme is a collaboration between the University of the West of England (UWE) and Systems Engineering and Assessment Ltd (SEA). The authors would especially like to thank Ian Woodrow, Chris Chaloner and Ben Olivier for their guidance, technical input and discussions; Chris Bytheway for constructing the asteroid demonstrator; Ian Horsfield and Ian Gillhespy for LinuxBot Support; Robotics advice and discussions of Chris Melhuish, Jan Wessnitzer, Jacob Hurst, Matt Studley, Jason Welsby and Chris Harper.

### References

- Arkin, R.C., (1998), *Behaviour-Based Robotics*, MIT Press.
- Bornschlegl, E., (2002), (On-Board) Autonomy Objective and Technology Support, *ESA. AI Planning and Scheduling for Autonomy in Space Applications (AIPS-ASA)*, July, 2002.
- Kannan S, Hur J, Saroufiem G, Yavrucuk I (1998), Georgia Tech UAV Software Systems, *Technical Note*, UAV Research Facility, Georgia Institute of Technology, Atlanta.
- Murphy, R.R., (2000), *Introduction to AI Robotics*, MIT Press.
- NASA Website (2002), Chronology of Lunar and Planetary Exploration, <http://nssdc.gsfc.nasa.gov/planetary/chronology.html>
- NASA Website (2003), Asteroid Radar Research, <http://echo.jpl.nasa.gov/~ostro/kle/index.html>
- Nelson ML, DeAnda J, Fox R, Meng X (1999), A Software Control Architecture for Autonomous Vehicles, *SPIE Conference on Unmanned Ground Vehicle Technology*, pp 274-181, April 1999.
- Ridao P, Yuh J, Battle J, Sugihara K (2000), On UAV Control Architecture, *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 855-860.
- Winfield, A.F.T., Holland O.E., (2000), The application of wireless local area network technology to the control of mobile robots, *Journal of Microprocessors and Microsystems*, Elsevier, Vol 23/10, pp 597-607.
- Yeomans, DK (1995), Physical and Orbital Characteristics of 433 Eros, *Journal of Astronomical Sciences*, Special Issue on the NEAR Mission to 433 Eros, vol 43, no 4, pp417-426.