Automated Vision-based Recovery of a Rotary Wing Unmanned Aerial Vehicle onto a Moving Platform

Thomas S. Richardson
Department of Aerospace Engineering, University of Bristol, Bristol BS8 1TR, United Kingdom
e-mail: thomas.richardson@bristol.ac.uk
Chris G. Jones and Alexey Likhoded
Department of Aerospace Engineering, University of Bristol, Bristol BS8 1TR, United Kingdom
Ed Sparks and Andrew Jordan
Roke Manor Research, Romsey, Hampshire SO51 0ZN, United Kingdom
Ian Cowling and Simon Willcox
Blue Bear Systems Research, Bedford MK41 6JE, United Kingdom

Received 25 March 2012; accepted 27 May 2013

This paper describes an integrated control demonstration in which a rotary wing unmanned aerial vehicle (UAV) landed on a moving platform using vision-based localization of the target vehicle. Three key areas are covered in this paper: the system components and integration, the control system structure and design, and finally flight test data including the first such successful landing. All the development and demonstration flights were carried out in the field, and key system challenges and lessons learned are highlighted throughout the paper. Results are given for landings carried out on both stationary and translating targets, in wind speeds of up to 15 knots. While the translating target was constrained to linear motion, there are several applications such as recovery to a land-based convoy for which this capability would be applicable. The results show that the approach adopted for this demonstration could form part of a practical solution to the problem of automated UAV recovery in both civilian and military environments.

1. INTRODUCTION

Unmanned Air Systems (UAS) are often used in civil and military situations that are hazardous to human operators (Unmanned Aircraft Systems Roadmap–DoD, 2005). Automated recovery of the vehicles is therefore highly desirable, both onto stationary and moving platforms. In late 2009, the UK Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre (DTC) (Systems Engineering for Autonomous Systems Defence Technology Centre, http://www.seasdtc.com/) decided to pull together several existing strands of work to create an Integrated Control Demonstration (ICD), the purpose of which was to evaluate the practicality of automated recovery using vision-based localization and guidance. Led by the University of Bristol and Roke Manor Research, this team carried out a series of successful flight tests, culminating in fully automatic landings onto both stationary and translating targets. This paper describes the process adopted and the key enabling technologies that were necessary for a successful outcome.

Direct correspondence to: Thomas S. Richardson, e-mail: thomas.richardson@bristol.ac.uk.

The use of vision sensors for guidance has many advantages. Currently, GPS is the most common method used for determining an unmanned aerial vehicle (UAV) position for navigation (Sabatini and Palmeri, 2008), but it is generally not considered accurate enough for precision automated landings without the use of additional sensors. This can be enhanced using differential GPS (DGPS), however this requires the transmission of information between the air vehicle and the landing platform. Furthermore, vision sensors are in most cases passive (they do not emit radiation) and are therefore less likely to reveal the aircraft location to hostile forces. Additionally, since they do not rely on an external signal, they are not vulnerable to jamming or loss of coverage. At close range, vision sensors can be millimeter accurate and are capable of determining not only distance but also the relative orientation between two objects.

There is a wealth of work and different vision-tracking methods available for this type of application. These include both feature-based methods (Mikolajczyk and Schmid, 2005) and direct methods (Iran and Anandan, 2000). Some of the approaches require prior knowledge of the targets, and some extract information from the surroundings in real time (De Wagter and Mulder, 2005, Aug). One of the first applications of a visual system being used for

control onboard a rotary wing UAV was by Amidi (1996), who developed and demonstrated a vision-based helicopter positioning and control system for a Yamaha R50. There have been several examples since, including more recent work in object tracking from a rotary wing UAV by Mondragn et al. (2010), who presented a three-dimensional (3D) pose estimation using planar object tracking for the control of UAVs. This is suitable for complex control tasks such as autonomous landings and accurate low-altitude positioning. The vehicle used by Mondragn is also based on the same airframe as the SR20 discussed in this paper, and results are given for real-time tracking during flight.

Shakernia et al. (2002) present a multiple-view algorithm for the vision-based landing of a UAV. They state that the use of multiple views significantly improves the motion and structure of the estimation. Error limits of 7 cm are given for each axis of translation, and their controller maintained the Yamaha R-50 in the hover above the landing platform in the presence of very high winds. In the work by Saripalli et al. (2002), they demonstrate repeatable landings onto a stationary target using a real-time vision tracker as the primary sensor. An accuracy of 40 cm was quoted for the landings carried out in this work, with the ground effect being identified as a key factor preventing precise control near the ground. Saripalli (Saripalli and Sukhatme, 2003) followed on and extended this work to consider landing on a moving target, again in the field. Video was processed offline, however, and the tracker identified as suitable for the moving landing. The platform speed in this case was in the region of 0.1 m/s, with close tracking of the platform position being clearly demonstrated.

In work by Merz et al. (2004, 2006) they used a modified Yamaha RMAX with an inner loop Yamaha Attitude Control System (YACS) as the testbed for vision-based autonomous landings. The touch down precision they quote is 42 cm and successful landings were achieved with wind speeds of up to 30 km/h. Work has also been carried out with small fixed wing UAVs, for example by Barber et al. (2007, 2009), who present a method for using vision-based feedback to land fixed wing MAVs. It is claimed to be robust to wind and capable of handling both stationary and moving targets. Experimental results are presented that clearly demonstrate the benefit of using vision-based feedback.

Hamel et al. (2010) present a nonlinear controller for hovering flight and landing control on a moving platform. Divergent optical flow is used for control with a quad-rotor UAV as the test platform. In this work, an X4-flyer made by CEA was flown with an embedded camera looking directly down from the body. The manoeuvre considered by Hamel is that of a vertical landing onto a moving platform with unknown dynamics, such as might be shown by the deck of a ship in high seas. Experimental constraints limited the gain that could be used, but successful landing manoeuvres were demonstrated. Oh et al. (2005, 2006) also consider the landing of a helicopter onto a moving ship, but in this case using a tether that is under tension. Two controllers were designed and demonstrated in simulation, and they were shown to be effective for the problem considered. Their future work included implementing the system on an autonomous UAV using a simulated moving deck as the landing platform.

Garratt et al. (2009) follow on from the previous work in Oh et al. (2005, 2006) to consider visual tracking and LIDAR positioning for the automated launch and recovery of unmanned rotorcraft from ships at sea. The objective was to consider a relatively low-cost guidance system that determines both the distance and the orientation of the deck. For these tests, the platform used was a Yamaha R-Max with a payload of 30 kg and an endurance of 1 h. A single beacon was used to identify the target, and in conjunction with the laser sensor, it is all that is required to fix the position of the helicopter with respect to the center of the deck. The stated motivation for this is to minimize the likelihood of losing track during the approach and landing at sea, and thus creating a system that is robust to operational requirements.

Marconi et al. (2002) also consider a controller designed for the vertical landing of a VTOL aircraft on an oscillating platform, the motivation of which is for landing on a ship at sea. This is considered to be a nonlinear regulator problem and is shown to be robust to uncertainties. More recently, both Nonami et al. (2010) and Wenzel et al. (2011) have presented methods that, when applied to small UAVs, allow them to carry out automatic takeoff, tracking, and landing onto a moving target. Sven et al. have also demonstrated a vision system suitable for autonomous landings using known real dimensions for relative estimations. At the other end of the scale, Scherer et al. (2012) present the results from eight successful landing tests using a full-scale helicopter as the test platform. They were able to classify landing zones from point cloud maps, with the vehicle automatically selecting the landing site and approach. These are noted as the first demonstration of a full-scale autonomous helicopter selecting and successfully landing on its own landing zones. One of the primary motivations is given as casualty evacuation where the path to a ground goal needs to be found. There are limitations identified, but the tests were shown to be extremely successful. In 2011, an automatic landing of the Boeing Unmanned Little Bird onto a moving target was carried out by Boeing and Thales (Thales, 2011). Sea trials of the demonstrator are planned for the system using a French Navy frigate and the same UAV.

A key paper by Kendoul (2012) provides an excellent and very thorough survey of advances in the guidance, navigation, and control of unmanned rotorcraft systems. It provides a clear and comprehensive overview of the field, and is highly recommended reading for those interested in the field. Kendoul covers platforms, definitions, linear and nonlinear controllers, navigation, sensing, and guidance. Just one of the conclusions drawn is that there is a gap between theoretical work and experimental results, and

that significant engineering work will be needed to close this gap.

The objective of the SEAS DTC demonstration presented in this paper was therefore to consider the recovery phase of a UAV and show that it is possible to land a rotary wing UAV onto a moving platform automatically, using a vision tracker as the means of localization in the final descent phase. This project was designed around a selected UAV platform, the Rotomotion SR20, which itself is based upon a modified radio controlled helicopter. The primary reason for the selection of this UAV was the application programming interface (API) that is available onboard, making the task of issuing commands to the UAV relatively simple. All the test flights for the project were performed outside of the laboratory environment and in the field.

The key contributions of this paper, therefore, lie in the experimental results, with landings carried out onto stationary and moving targets in the field, and with the associated environmental limitations such as wind and lighting conditions. Although the motion of the ground vehicle was constrained, the engineering of the landing system and approach protocol was challenging, and is described within this paper together with the practical lessons learned.

Drawing on the previous work described above, it was therefore expected that the ground effect would be a key issue with regard to landing precision, and that a small-scale rotary wing UAV such as the SR20 would be relatively robust to wind disturbances. Landings onto stationary targets have been carried out by several research groups, however landings onto moving targets in the field are less common. For the purposes of this paper, the translating target was constrained to linear motion, however there are several applications, such as recovery to a land-based convoy, for which this capability would be applicable.

GPS has been used in this work to bring the UAV close enough to the target to obtain a visual track, however in conjunction with other methods, a vision tracker could be used in a GPS denied environment to allow for fully automated touchdown. Although demonstrated on a small rotary wing UAV, these methods would also be directly applicable to both micro and full-scale vehicles, single and multirotor, provided the required computation power and sensors were available. One of the benefits of using a conventional configuration in this work, however, was that roll and pitch disturbances during the landing manoeuvre were minimal, which allowed a fixed camera mount to be used.

In Section 2, the system setup is described, including air and ground components and the flow of information between them. Section 3 describes the different elements of the control system design and the method used to tune them. Section 4 presents the results for the first successful automatic vision-based landing, and finally conclusions and recommendations are given in Section 5.

Table I. SR20 system properties.

Propulsion	Electric (m)	Dry Weight	7.5 kg
Length	1.22	Max Speed	50 km/h
Height	0.56	Endurance	12–24 min
Main rotor diameter	1.75	Max Payload	4.5 kg
Tail rotor diameter	0.25	Telemetry Range	800 m

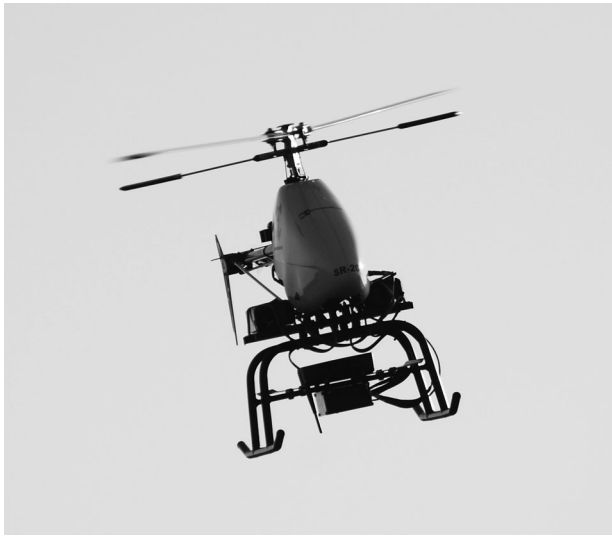

Figure 1. Standard Rotomotion SR20 electric UAV.

2. SYSTEM DESCRIPTION

2.1. Unmanned Air Vehicle

2.1.1. Airframe

The Rotomotion SR20 VTOL UAV, a commercial off-the-shelf system based on a conventional model helicopter, was selected as the airborne platform for the experimental work. It was supplied with an integrated automatic flight control system (AFCS) and associated ground control station (GCS) software, capable of waypoint driven autonomous flight. This approach offered a faster, lower-risk route to flight testing when compared with integrating a bespoke solution. Although the source code for the flight control system was not provided, the required functionality was achieved using the API provided. Details on the SR20 are given in Table I.

Two airborne images of the SR20 are given in Figures 1 and 2. Figure 1 is of the standard SR20 during a test flight, and Figure 2 is of a modified SR20 that is carrying the following: an additional Linux computer, details of which are given in Section 2.1.2; a front-mounted hard-wired Sony video camera for vision tracking; and an additional rear-mounted SD video camera. The front-mounted camera used for the vision tracking has a resolution of 640×480 pixels and the viewing angle of the lens was 60 degrees. A relatively

Figure 2. SEAS DTC modified SR20 with fixed camera and onboard vision tracker.

wide viewing angle was selected in order to make identification of the target easier in the initial phase of the landing. The tradeoff is the height at which a firm visual lock can be achieved; the wider the viewing angle, the closer the camera needs to be to identify the target. A calibration of the camera/lens combination was performed that covered the intrinsic parameters including focal length, pixel aspect ratio, optical center, and image distortions.

2.1.2. Installed Equipment

To minimize transmission delays, the vision processing was carried out onboard the modified SR20. For these purposes, an AgentBox was mounted next to the existing Rotomotion Flight Control computer. This uses standard computing hardware including a solid-state drive and bespoke configured version of the Linux operating system (OS). A single tracking camera, mounted pointing vertically downward, was connected directly to the AgentBox via a USB port and an onboard network switch and wireless bridge provided the necessary communications with the ground-based network.

The AgentBox built by Blue Bear Systems Research was based on a VIA Epia pico-ITX mainboard, model version PX10000G with a 1.0 GHz processor. This was combined with a 1 GB Transcend 533 MHz DDR2 memory SoDIMM. With a VGA output supporting 1920×1440, a USB keyboard was used to program and debug the onboard code in between test flights. During flight, the AgentBox was connected across the common network, wirelessly linked to the ground. High-resolution camera images were stored locally, with a reduced resolution video feed to the operator on the ground. This available computing power within AgentBox was sufficient to run RAPiD at 15 Hz with the outerloop guidance control system requiring negligible processing power compared to the vision tracking.

For the initial tests, up to and including the first autonomous landing on a moving platform, the outer loop guidance control system was run offboard of the SR20. For subsequent work, this was moved onboard to the vehicle, which was then only monitored from the ground station.

2.2. Additional Hardware

A modified car, with a landing platform two meters square, was adopted for the work presented in this paper. In subsequent tests, a smaller unmanned ground vehicle (UGV) has also been used with an associated smaller target platform.

Three ground-based computers were used to control and monitor the aircraft. The first (Linux OS) ran the GCS software supplied by Rotomotion. The GCS provided a map showing the current location of the aircraft together with key aircraft and control system parameters. Data from the AFCS and onboard sensors were fed back to the GCS for monitoring, recording, and post-flight processing.

The second was used to run PuTTY, a freeware telnet/Secure SHell (SSH) client, responsible for the AgentBox uplink, determining camera settings and displaying the camera view to the operator. It also demonstrated whether a lock had been acquired and provided the relative errors between the SR20 and the target vehicle. Visual lock acquisition could also be initiated and terminated on demand.

The third ran the outer guidance control loop software and was interfaced with a standard COTS USB GPS receiver. Guidance commands were sent to the UAV AFCS using the SR20 API. The information that these three computers provided allowed the system to be monitored throughout the approach and landing. In addition, a safety pilot retained overall control throughout the flights and could abort the approach at any point.

2.3. RAPiD Vision Tracking Software

A vision-based control architecture is any control system that makes use of vision-based information directly in the control or guidance loop. Vision data can be used for navigation, vehicle stabilization, vehicle guidance, obstacle avoidance, or target tracking.

Inferring 3D perception of an object in an unknown environment is a complex but solvable problem that can be done with either a single camera and known target, or two cameras at known locations focused on the same object. This project was focused on the last stages of the recovery process, namely locking onto a predefined target and using distance data to land the UAV onto a moving platform. It is possible to do this with a single camera, since by knowing the exact size and shape of the target, the distance between

Figure 3. Collecting RAPiD imagery for tracking.

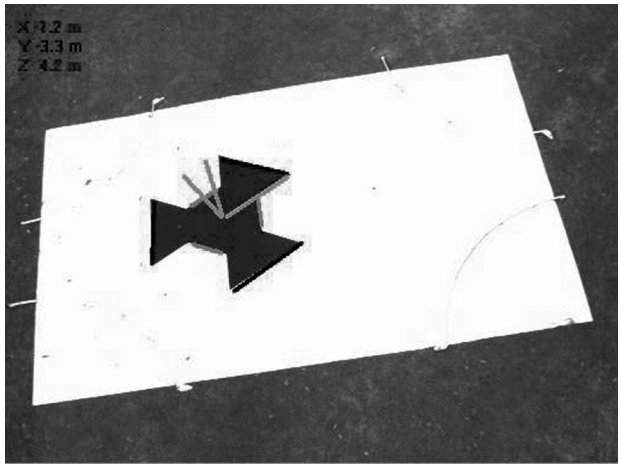

Figure 4. Screenshot from RAPiD demonstrating edge detection and axes alignment.

the camera and the target can be calculated using the size of the target in the image frame.

The visual tracker used in this work was the Roke Manor Research RAPiD (Realtime Attitude and Position Determination) software, which uses a model-based tracking algorithm. Model-based vision tracking uses a predefined description of the object being tracked in order to determine the position and orientation of the object with respect to the camera, and hence relative to the UAV frame of reference. The theory behind RAPiD can be found in a British Machine Vision conference paper by Harris and Stennett (1990) and a complementary paper on the Kalman Filtering of RAPiD pose estimates by Evans (1990).

In essence, RAPiD looks for observable features within the real world that can be compared with those that are predicted using an internal model. The use of a model can reduce algorithm complexity and run-time, and allows distances to be measured when the camera, lens, and image capture system have been correctly calibrated. For the purposes of this project, a RAPiD target model was defined and flights were carried out to determine how well the combination of tracker, processor, and camera functioned on the approach to landing. Figure 3 shows the SR20 during target tracking and Figure 4 is a screenshot that demonstrates the edge detection and axes alignment carried out by RAPiD during flight.

RAPiD operates in two distinct modes, namely acquisition and tracking. In the acquisition mode, the initial calculation of position and attitude of the target relative to the camera is based upon a single image. Using this single frame, edge detection is used to find features that are then matched with features from a known model. This is a relatively expensive process computationally and can involve a large number of combinations being searched.

In contrast, the tracking phase adjusts the existing pose estimate in order to match subsequent images. Feature search is restricted to the immediate vicinity of the predicted feature position. This significantly reduces the computational overhead by reducing the area of the image that is processed. Features that can be tracked in RAPiD include locally dark or light patches, edges (including surface markings, surface boundaries, and profile edges), and bars (two edges in close proximity).

RAPiD is ideal for the tracking of a landing target as it is capable of enabling and disabling control points within the tracking algorithm as they are revealed and obscured, respectively. This enables it to continue to track close to the target area when only part of the image is in view. RAPiD can also ignore control points that are on weak and error-prone edges, and it can also define an edge polarity such that strong edges with the wrong polarity caught in the search region are ignored. For rapid motions within the image plane, it can also predict ahead and reduce the variability of the pose estimates using a Kalman filter.

In Figure 4, the edges of the target area have been highlighted by RAPiD and used to identify a local set of axes aligned with the landing area. From knowledge of these, the relative pose of the onboard camera can be given. Local velocities are also estimated and used within the prediction step to generate the expected pose in a newly acquired image. It is possible to implement a more complex dynamic model within the RAPiD code and update the expected pose based on the aircraft movement, however this was not found to be necessary for the project. One change, however, was made to the original target, and this can be seen in Figure 5.

The visual target, therefore, used to define the landing area for all the subsequent tests is shown in Figures 5 and 6, with the latter showing a single image captured from onboard the SR20 during flight. The change made was to

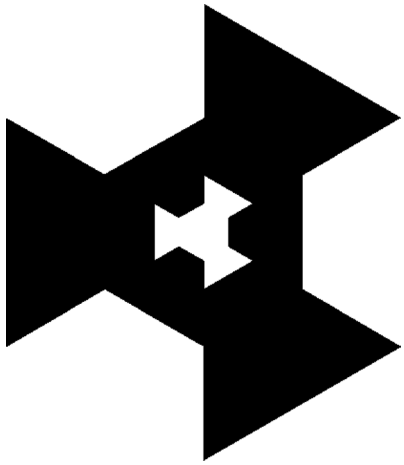

Figure 5. Updated RAPiD target.

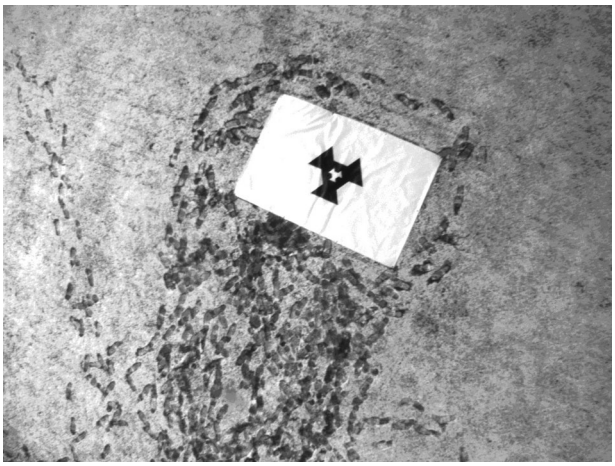

Figure 6. Onboard sample RAPiD target image.

include a white inner pattern to extend visual tracking closer to the ground. RAPiD initially tracks the outer black pattern and transitions from the outer pattern to the inner pattern as the UAV descends close to the ground. Trials were also carried out with additional information in the form of markers within the white inner pattern, however these were not found to be necessary and were not used for the tests presented here. The target was designed to provide a distinctive shape to aid initial acquisition. Contrast changes are detected to find edges of this geometric shape, which is then subsequently compared to control points, predefined within the software.

A Kalman filter was also implemented within the vision tracking software, which works simultaneously as a predictor and filter, using position and speed estimates to predict where the image will be in the following camera frame. RAPiD therefore returns the distances x_c, y_c, and z_c (in the camera's reference frame) and relative velocities between the camera and target, which can then be used to guide the aircraft toward the target area (Evans, 1990).

RAPiD works extremely well within the lab environment, however there are specific challenges to operating with the tracker in the field. With a target diameter of 1 m, a robust and reliable visual lock could be obtained at a distance of 10 m, and RAPiD maintained this to within a distance of 30 cm, camera to target. This enabled the SR20 to move close enough using standard GPS receivers in order to lock on to the visual target before commencing the descent. The primary difficulty, however, encountered in all the tests was related to significant changes in light levels. The camera used could auto adjust, however it was not sensitive or fast enough to cope with significant changes in light levels while allowing RAPiD to maintain a lock. If, for example, the sun were to pass behind a cloud during a flight, the rapid changes in illumination would often cause RAPiD to fail. Changes to the camera and code were tested during flight trials, however no robust solution to this problem was found at the time. For the tests presented here, the lighting conditions were relatively constant.

It was also found to be necessary to filter the output from RAPiD during both the acquisition and tracking phases. During the acquisition phase, pose estimations could vary significantly. Simple acceptance bounds were used in order to discard vision-based estimates that were significantly outside of those expected, and a number of pose estimates were checked sequentially before the data were used in the control algorithms. Once the pose estimate had settled, however, the tracking under constant lighting conditions was found to be smooth and accurate, with increasing accuracy as the distance to the target decreased.

In visual lock mode, the scale factor and known geometry were the sole means used to estimate the distance to the target. A single camera was used for these trials, however monocular or binocular vision could be combined with additional sensors such as a laser range finder or radar to improve the robustness of the pose measurement. No relative significant limit was identified in the approach speed when using RAPiD, and loss of lock was almost invariably due to the target moving out of the image plane or significant changes in the light levels. Results included in Section 3 for a stationary landing demonstrate the process and relative accuracy of the target acquisition.

2.4. System Integration

Figure 7 shows the complete system integration for both the ground vehicle and air vehicle. On the left-hand side, the safety pilot has overall control of the aircraft and can override the system at any time. The SR20 mounted network switch provides communications between the AFCS and Agentbox and interfaces with the wireless bridge to the ground. Position data provided by the Agentbox mounted vision tracker are broadcast globally across the network. The

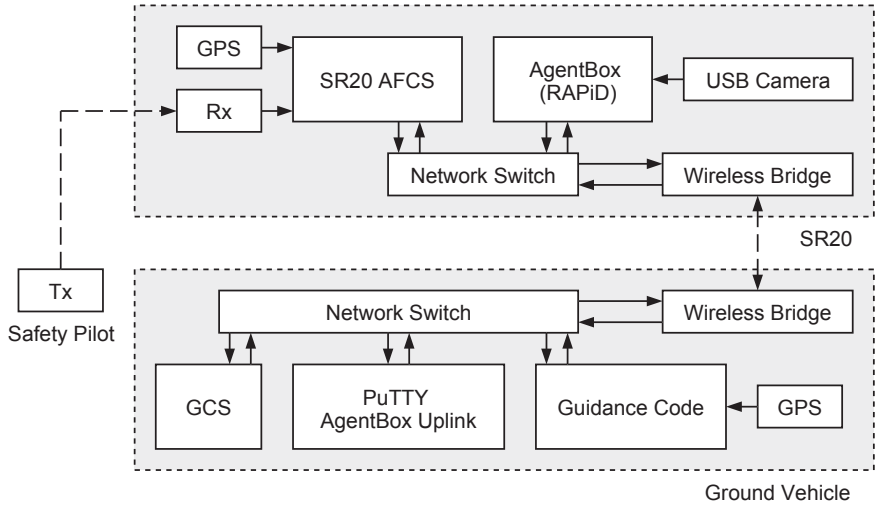

Figure 7. SEAS DTC UAV demonstrator system integration.

GCS can be used to monitor the SR20 AFCS and provides baseline flight data on the aircraft. The computer labeled "Agentbox Uplink" is used to monitor the airborne vision tracker, initiate, terminate, and modify camera parameters.

The third computer is used to run the outer loop guidance code, and in terms of speed, the AFCS is running at 35 Hz, the vision tracker at 15 Hz, and the outer guidance loop at 5 Hz. For the purposes of the demonstration it was decided to use a standard GPS receiver mounted on the ground vehicle to provide a reference with which to bring the aircraft into the locality of the landing platform. The location relative to the ground vehicle was found to be small enough to allow visual acquisition from a height of 10 m. If the system were employed in practice, this initial localization could also be carried out using alternative vision tracking algorithms that could remove any need to pass information or emit any signal from the ground vehicle.

3. CONTROL SYSTEM DESIGN

3.1. Automatic Stationary Landing

For the purposes of control, state information was generated in several axes, as shown in Figure 8. Transformations were set up between earth axes, camera axes, UAV body axes, and

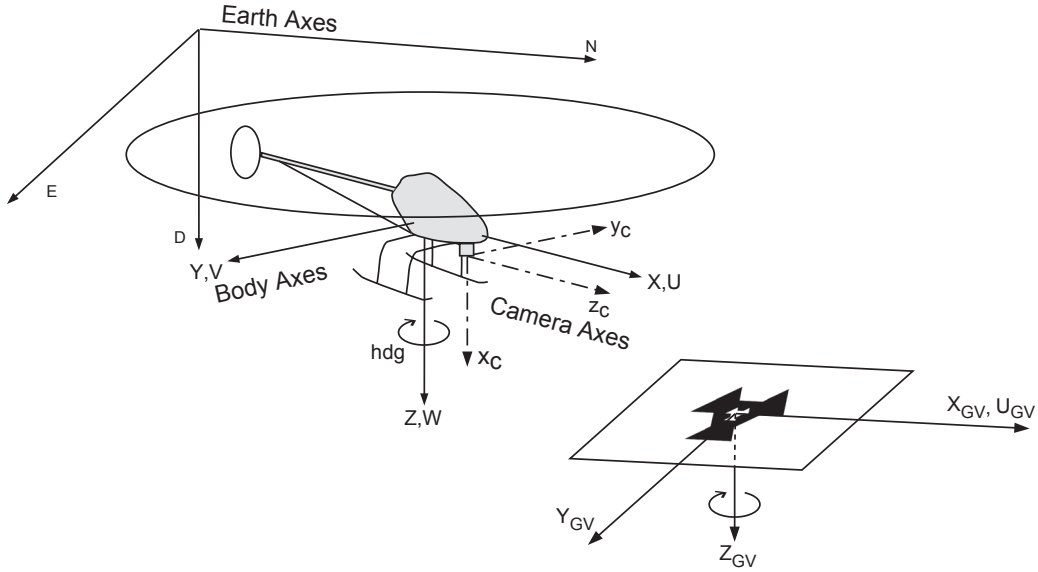

Figure 8. Axes definition: earth, air vehicle, ground vehicle, and camera frame.

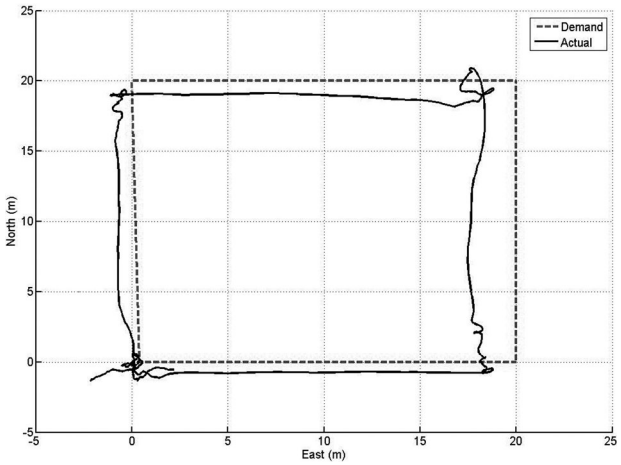

Figure 9. SR20 circuit response.

ground vehicle body axes. Positive directions for each are indicated in Figure 8. Furthermore, an origin was defined for the aircraft at the initial point of take off, with the axes aligned with the Earth axes. Waypoint demands were then passed to the SR20 in terms of this localized set. Locations in the global reference frame are measured in meters north, east, and below the origin. These are referred to as north, east, and down (NED).

Figure 9 shows the standard SR20 response to a series of four waypoint commands, employed in order to fly a square circuit with sides 20 m in length. Starting at the bottom left and initially moving to the top left of the circuit, the aircraft was asked to perform a 90 degree turn at each corner so that it was always facing in the direction of travel. What can be seen from this diagram is that although the response is clearly stable, the response is only accurate to within approximately ±4 m as measured, well above the submeter accuracy required for a moving landing. At the corners of the square, the helicopter also has a tendency to wander. This is due in part to the onboard AFCS reducing the commanded control input when the error between demanded and actual position is small. Furthermore, this is a top level view, and the response was even less accurate over time in terms of altitude. The key point to note is that the SR20 is easy to guide but is difficult to control precisely.

To augment the aircraft, the system was set up as outlined in Section 2.4 and initial tests were carried out in order to achieve an automatic landing on a stationary target. This was done by repeatedly overwriting a required waypoint and transit speed on the AFCS in order to guide the SR20 toward the landing platform based on the vision tracking data passed across the network from AgentBox. Hence the outer guidance loop would update this at 5 Hz based on the current vision-based estimate of the target location.

Figure 10. Stationary landing touchdown.

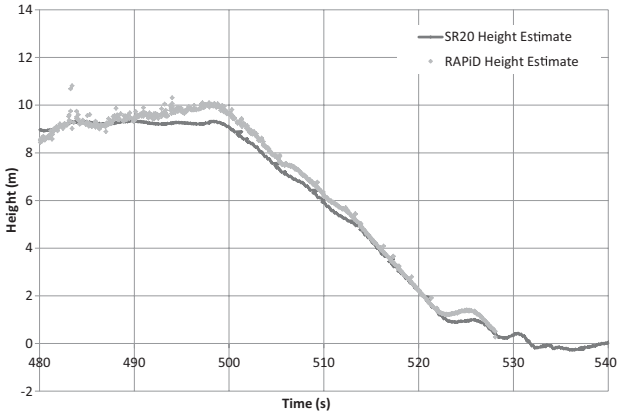

Figure 11. SR20 and RAPiD height estimates during a vertical landing.

The SR20 AFCS generates a response that accelerates toward the maximum required transit speed and then decelerates as it approaches the desired waypoint. The outer loop control law for the stationary landing was therefore set up to provide a waypoint at a set distance below the aircraft, and the demanded maximum transit speed was used to control the rate of descent.

Figure 10 shows the SR20 at the point of touchdown, directly over the visual target. This process was found to be robust, repeatable, and accurate to within ±20 cm. The accuracy could be improved still further if a smaller target were used and the camera location was moved to allow a greater distance between the camera lens and the target on touchdown. In itself, this is a useful capability, as it enables the aircraft to land within a few centimeters of a desired target, based only on a series of known visual cues. Applications could include civil emergency services or mobile military operations.

Figures 11 and 12 provide the height and lateral error estimates for a vertical automatic landing. In Figure 11, the RAPiD height estimate can be seen to follow the SR20

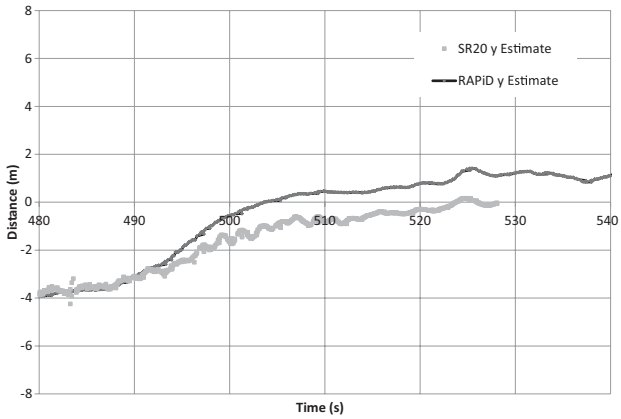

Figure 12. SR20 and RAPiD lateral error estimates during a vertical landing.

estimate relatively closely throughout the descent, only losing lock just prior to the touchdown. There are several interesting points to note, starting with the vision-based estimates at a distance of 10 m from the target. At this distance there is increased scatter of the estimates, which reduces as the SR20 approaches the target. With the camera and lens configuration used, 10 m was identified as a workable limit for robust tracking. Increasing the size of the target, reducing the viewing angle of the lens, or increasing the resolution of the camera would all increase the distance at which a reliable lock could be established and maintained. The SR20 relative position was also zeroed shortly before this flight, which is why the two height estimates track so closely. Over time, the SR20 estimate was found to vary, however the vision tracker was consistent when the visual lock was established.

The second point to note is that there is a slight increase in altitude just before the touchdown that is registered by both the SR20 and RAPiD estimates. This is due to the ground effect and required a prescheduled increase in demanded vertical descent rate, and hence control power, to achieve an accurate landing. The faster that this region was transitioned through, the more accurate the resultant landings. As can be seen from the results in Figure 11, this effect was experienced in the region of 1.0–1.5 m from the ground.

The vision lock is lost at approximately 30 cm above the target, which, due to the location of the camera, means that the aircraft itself is approximately 20 cm from touchdown. At this point, the current trajectory and descent rate are maintained for a predetermined length of time to ensure touchdown before reverse collective is applied. Thus the final stage for both the landings on the stationary target as well as the moving target is carried out along a predetermined open-loop path. Current and future developments include moving the camera to provide tracking up to the point of contact, and the inclusion of sensors to indicate a positive contact with the landing platform.

In Figure 12, the estimated lateral error with respect to the target is given for both the SR20 and RAPiD. As with the height estimate, the SR20 was initialized on the center of the target shortly before the flight. What can be seen from the plot is that although the initial estimates are consistent, the RAPiD error is reduced as the aircraft approaches the landing. If the guidance loop operated using the SR20 estimate of position, in this case the aircraft would touch down over 1 m from the target point. The longitudinal estimates of error show a very similar response.

Stationary vision-based landings were found to be very robust and repeatable. The results presented in Figures 11 and 12 are a sample taken from a series of six landings during one flight, all of which were successful. The aircraft was flown over the target manually to an altitude of approximately 10 m, where control was then passed over to the AFCS onboard the SR20. Once a robust vision lock was confirmed, the autoland sequence was initiated with the SR20 descending onto the target. Over 50 automatic stationary landings were carried out before the project moved on to the moving platform.

3.2. Control Design for Translating Target

The first task required for the moving target was to establish a model for the speed response of the SR20 to a reference demand. A series of flight tests were carried out with increasing maximum transit speed, and these were then used to fit a second-order reference model using Matlab. [This model was found to work very well over a range of demanded speeds from 1 to 10 m/s]. The transfer function that relates SR20 ground speed to the reference demand is

$$\frac{U_{\mathrm{UAV}}(s)}{r(s)} = \frac{-0.1551s + 0.3170}{s^2 + 0.5171s + 0.4932}. \quad (1)$$

Experimental SR20 results and a transfer function step response comparison are shown in Figure 13. Using this simple model as the basis, a simulation was set up within Matlab in order to tune and test the outer guidance loop. Rotomotion supplied a simulation environment that has exactly the same API as the actual vehicle. Both the Simulink model and the Rotomotion simulation environment were used extensively for testing the control system design and system integration. For example, it was possible to run the outer-loop control system in conjunction with the vision tracker, but linked to the Rotomotion simulation in order to eliminate risk in the process and reduce the requirement for flight testing. The movement of the simulated SR20 could therefore be tested in response to a movement of the actual target in the image frame of the camera. This was found to be invaluable and reduced the flight time requirements as the complexity of the system was increased.

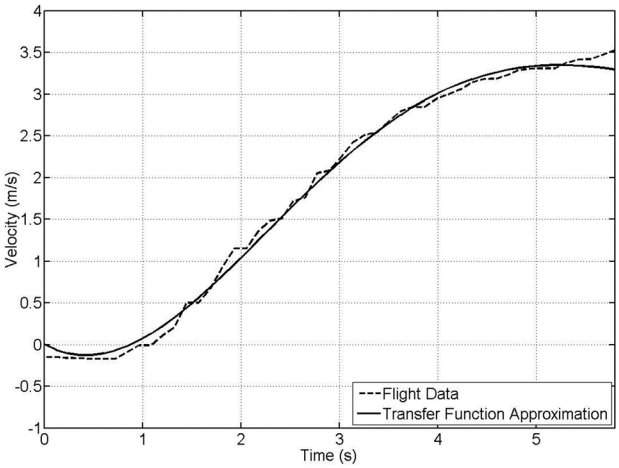

Figure 13. SR20 model reference.

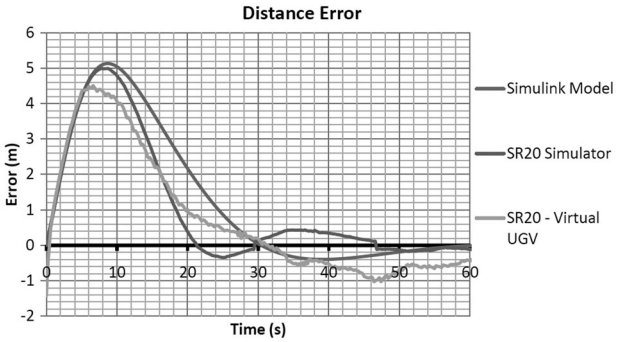

Figure 14. SR20 experiment and model comparison.

To test the speed response of the aircraft to a translating target, a theoretical target was created that moved at a constant speed. Figure 14 shows the error in position of the SR20 relative to the ground vehicle for the Simulink model, the SR20 simulation, and finally the experimental test flight data from the SR20. For the final set of results, the SR20 was established in a hover, then a theoretical target was created, moving away from the nose of the aircraft and the guidance loop generating commands for the SR20 designed to reduce the error in position. All three are stable and closely matched in terms of the speed and maximum error. The control system used for these sets of results is given in Section 3.3.

3.3. Speed Control

In essence, the speed control demand is generated using a PI controller applied to the error in position between the ground vehicle and the air vehicle, and a differential term applied to the difference in horizontal speeds between the two. Therefore, in steady state, both position and speed should be matched. Figure 15 shows the schematic for this structure, for which the gains were tuned using LQR (Bryson, 1994) and the transfer function model found previously. This approach becomes more complex when the relative spatial geometry of the two vehicles is considered. At this point, the speed control is only considered for the two-dimensional case, north and east.

As shown in Figure 15, the input to the SR20 block represents the commands passed to the API. In essence, for all the tests carried out in this project, the waypoint demand was overwritten at the outer control loop frequency. This was generated as a point in three-dimensional space combined with a demanded maximum transit speed. The control system given in Figure 15 is therefore the basis for the transit speed demand passed to the SR20.

The sign of the error, however, also changes depending on the relative position of the vehicles. Figure 16 shows how this was generated. The ground vehicle path is known, either from GPS data or when using vision tracking, by calculating the movement of the image in the reference frame combined with the motion of the air vehicle. A theoretical line is then drawn perpendicular to the directional of travel and which passes through the center of the landing target. If the air vehicle lies behind this line, the error is taken as positive; conversely, if it lies in front of this line, it is taken as negative.

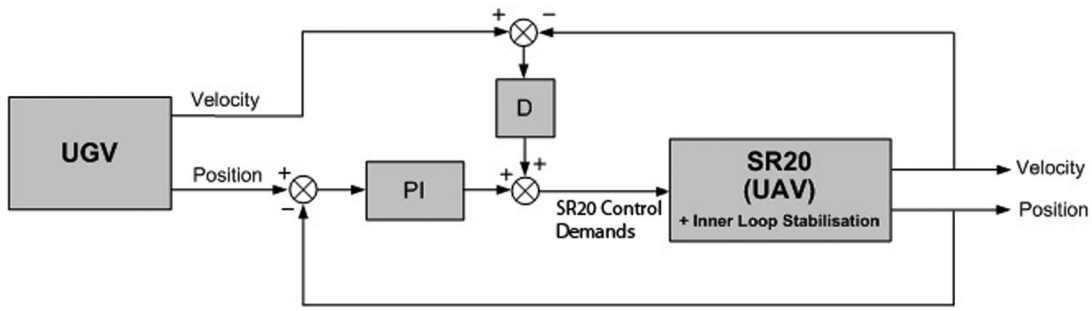

Figure 15. Speed control system structure.

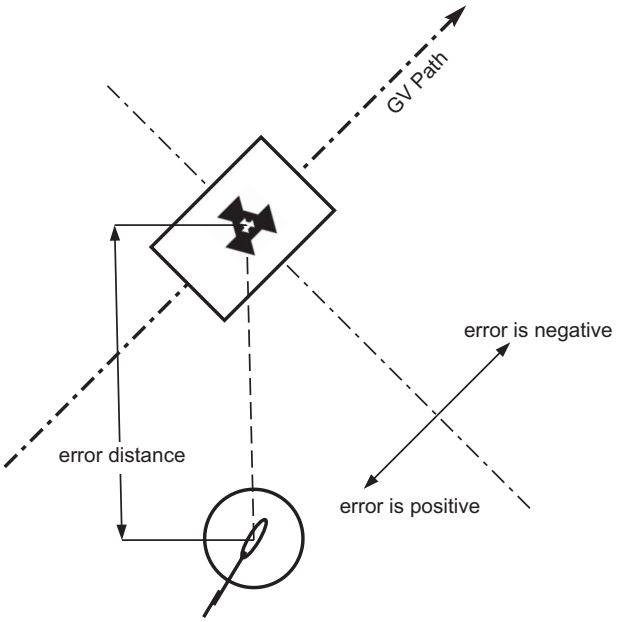

Figure 16. Link between SR20 position and the sign of the error used in the speed controller.

Filters were also applied to all the GPS signals of the form

$$y_t = 0.6 * x_{t-1} + 0.4 * x_t, \tag{2}$$

where the values were found experimentally in the field balancing noise with speed of response. It was also found to be necessary to limit the data used in order to remove erroneous vision and GPS readings. These limits were chosen as shown in Table II. Limits were also placed on the integral speed demand term, the waypoint position relative to the air vehicle, and the overall speed demands passed to the SR20 API. These were all selected in order to ensure that the aircraft was flying at all times within a well-defined and known flight envelope.

During the approach to the ground vehicle and prior to the descent phase, the vertical component of the waypoint demand was kept constant. Once the required criteria had been met to initiate the final descent, the target waypoint was moved to a constant position below the current SR20 height. The position of this point relative to the aircraft allowed the glide slope to be defined.

These limits were all necessary to be able to approach the landing within a clearly defined and predictable envelope. The ground vehicle speed was kept constant, and the track constrained to linear motion in order to simplify the problem. Changes in ground vehicle speed could be accommodated by changing the limits on the aircraft response and automating the definition of the glide slope. Changes in the dynamic response of the aircraft, perhaps with a different platform, would require the speed controller to be retuned, however the basic control structure would not need to be altered. More complex ground vehicle motion, however, it would require a more complex controller, and unless variations in motion were smooth and relatively slow, it would also require a more responsive and agile air vehicle.

3.4. Lateral Control

The position of the desired waypoint needed to be calculated relative to the air vehicle, and this was selected based on the position and relative velocity of the target, as illustrated in Figure 17. The controller selected one of the three following options:

1. Target is moving and aircraft speed demand is positive.
   A vector is projected from the UAV through a point 10 m ahead of the ground vehicle. The position demand is then chosen at a point 15 m from the UAV along this vector.
2. Target is moving and aircraft speed demand is negative.
   A vector is projected from the UAV through a point 10 m behind the ground vehicle. The position demand is then chosen at a point 15 m from the UAV along this vector.
3. Target is stationary (speed less than 0.5 m/s).
   A vector is projected from the UAV directly through the target. The position demand is chosen at a point 5 m from the UAV along this vector.

The choice of the point 10 m ahead of the ground vehicle was calculated based on the desired tracking response and ground vehicle speed, and could be automated if required. A distance of 15 m from the SR20 was chosen to ensure that the SR20 would remain at the desired maximum speed. If the approach speed were increased, then the length of this vector would also need to be scaled.

A target with a speed less than 0.5 m/s was treated as a static target because of the noise inherent in the sensors, particularly the GPS. The error in a GPS measurement varies with time, meaning that velocities can be implied,

Table II. Data limits.

	Maximum X error (m)	Maximum Y error (m)	Maximum Z error (m)
Initial Distance	100	100	20
Between Distance Readings	4 (20 m/s at 5 Hz)	4 (20 m/s at 5 Hz)	1 (5 m/s at 5 Hz)
Between Speed Readings	0.5 (2.5 m/s at 5 Hz)	0.5 (2.5 m/s at 5 Hz)	0.5 (2.5 m/s at 5 Hz)

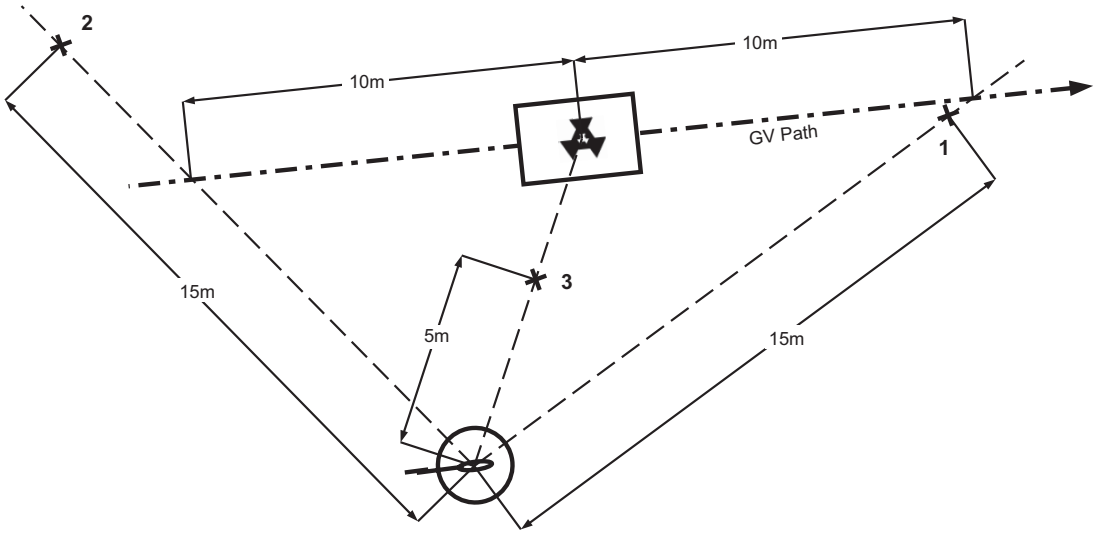

Figure 17. Target locations: (1) Moving target, positive speed demand; (2) moving target, negative speed demand; (3) stationary target.

even if the target itself does not move. Second, a real but small velocity can be masked by the noise, making the true direction difficult to resolve. With a ground vehicle moving faster than 0.5 m/s and appropriate filtering, it was found that the heading of the ground vehicle could be reliably calculated.

For the purposes of these flight tests, the outer-loop control system architecture and tuning were designed to be robust. Because of this, it was tuned to respond relatively slowly to the ground vehicle movement, allowing the flight tests to be carried out methodically, predictably, and accurately. Once the visual lock is established, the velocity of the GV is estimated by RAPiD relative to the air vehicle and used in the outer-loop controller. The onboard estimate of the velocity of the SR20 was used throughout, the combination of which with the RAPiD estimate provides an estimate of the GV velocity.

4. AUTOMATIC LANDING ON MOVING TARGET

Flight trials were carried out progressively in order to incrementally add capability. The previous tests had shown that the GPS tracking was accurate enough to acquire a visual lock, the controller was robust and accurate enough to track the moving vehicle, and the visual lock was reliable and the glide slope was controllable and predictable. The final approach and landing were carried out in four distinct phases:

1. Initial rendezvous carried out using relative GPS signals, at a constant altitude of 10 m.
2. Once visual lock was achieved, the target was tracked using vision data feedback to the control system.

Figure 18. SR20 on approach.

3. Once landing criteria had been satisfied, the descent was initiated. Landing criteria used were UAV within 2 m (horizontally) of the target with the lock maintained for at least 3 s.
4. Vision lock maintained to within 30 cm of target. Landing approach continued to touchdown on the current trajectory.

Figures 18 and 19 are two images taken from the first moving landing. The first is from a stationary camera with the aircraft tracking the platform from approximately 8 m above it, and the second is from a camera onboard the ground vehicle just prior to touchdown.

Figures 20 and 21 provide the error for the SR20 relative to the ground vehicle over the course of the entire flight test

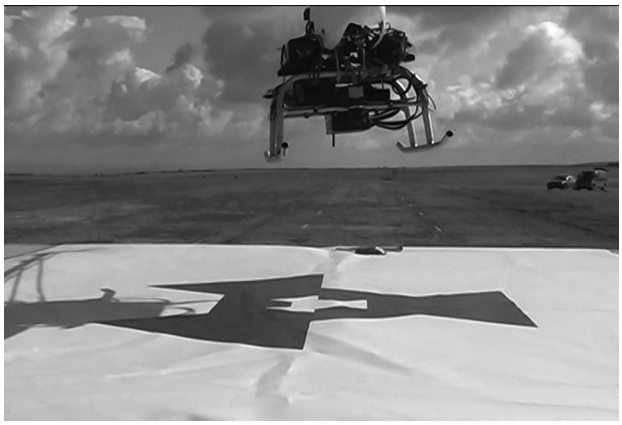

Figure 19. SR20 just prior to landing.

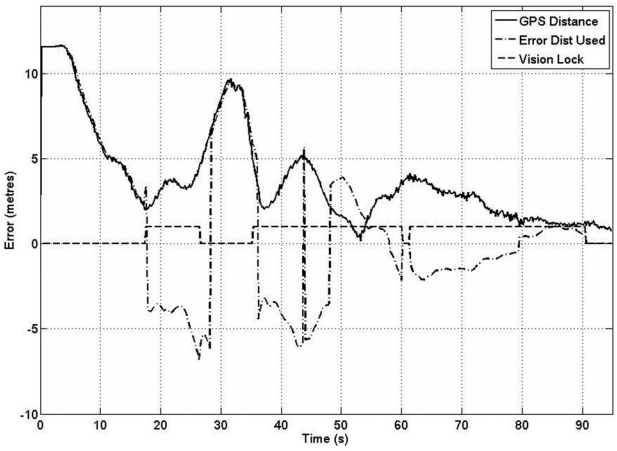

Figure 20. Measured error for the full approach (vision lock=1). Switching is shown between GPS and RAPiD estimates.

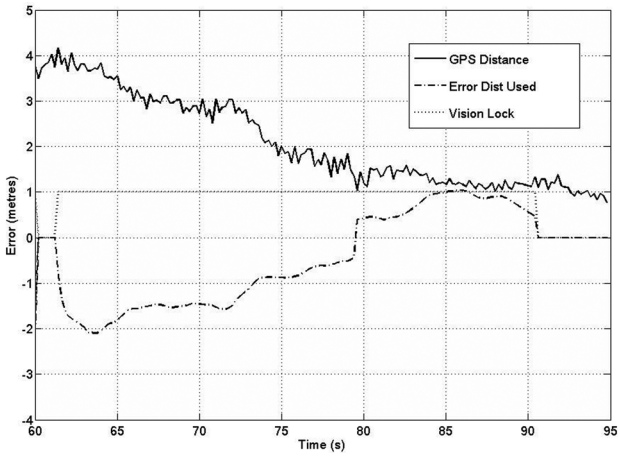

Figure 21. Measured error for the final approach (vision lock=1). GPS error and the error input to the control system are given.

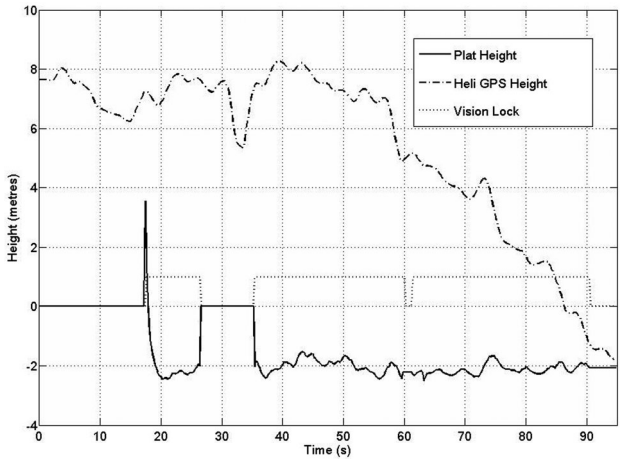

Figure 22. SR20 GPS height, estimated platform height, and vision lock indicator. Platform height is taken to be zero in the absence of vision lock.

and for the final approach. Figure 20 also shows the sections of the flight for which the visual lock was acquired. In these regions, the controller is using the visual data for localization in place of the GPS data. Figure 20 also shows that the error was constantly switching from positive to negative. This is because there was a high magnitude of lateral error as seen from the northeast plot, which meant the SR20 kept crossing the 90 degree perpendicular line. This behavior is expected in an event of a high lateral error magnitude as the SR20 effectively moves closer to the ground vehicle along the 90 degree line. In Figure 21, the vision lock can also be seen to be robust up until the point where it loses track at a distance of approximately 30 cm, at which point the current trajectory is followed.

The distance error also increases significantly at around 25 s due to a single anomalous reading. This causes the controller to sense that there has been an overshoot since the perceived ground vehicle velocity is toward the SR20 and it responds accordingly. However, the controller recovered well, and once this error was reduced and a reliable visual lock was reacquired, the switching ceased and the distance error was reduced.

Figure 22 is a plot of the height (+ve vertical upwards) of both the ground vehicle and the air vehicle. The ground vehicle platform height is assumed to be at zero until a visual lock is acquired, after which the height is given relative to the initial ground plane. The final phase can be seen to be initiated at approximately 60 seconds with a well-controlled descent onto the platform. It is interesting to note that even during the short time before this test was carried out, the GPS drift results in the landing target appearing at approximately −2 m in terms of height. The initiation of the final descent is based upon the requirement that the SR20 position is maintained using vision tracking to within a 2 m

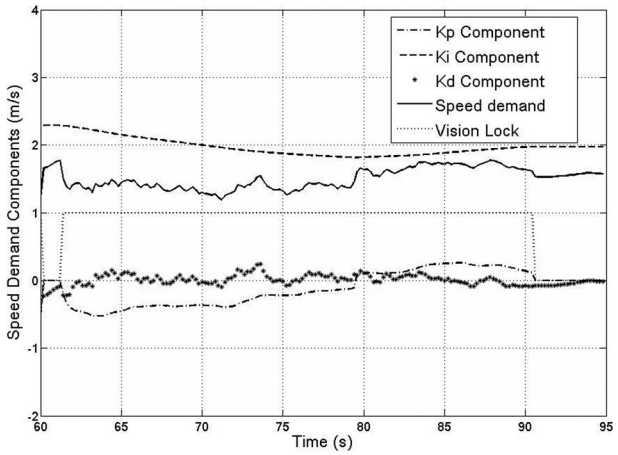

Figure 23. Speed demand during the final approach. Components of the control signal are shown together with the vision lock indicator.

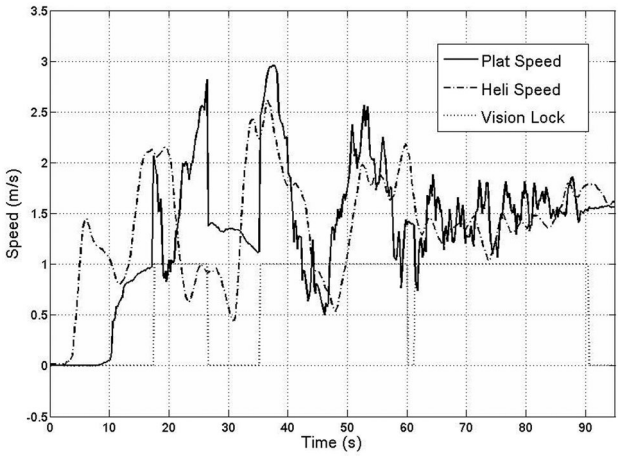

Figure 24. UAV and GV speed for the full test. Note the rapid changes in estimated GV speed when the vision lock status switches.

radius of the target for 3 s. As with numbers previously identified, the reason for this was to provide a robust and safe approach to the landing. This requirement could easily be relaxed given an expanded flight envelope or different air vehicle.

Figure 23 shows the different terms acting within the speed controller during the final descent phase. The components labeled Kp and Ki are the signals from the proportional and integral terms acting on the relative error, and together with the derivative signal Kd, they are fed through to provide the overall speed demand. This is therefore in essence the outer loop, with the inner loop autopilot on the SR20 acting on the speed demand. At the start of the flight test, the proportional term dominates the response (not shown), however once the SR20 has settled down near steady state it can be seen that the bulk of speed demand comes from the integral term with only small variations in demand coming from the proportional and derivative terms. The ground vehicle speed was kept constant during each of the landing runs, with speeds ranging between 1 and 2 m/s. For the results shown here, the landing platform was moving at a constant speed of approximately 1.5 m/s along the runway.

Figure 24 provides the UAV and GV speeds for the full flight test. Both are stationary at the start of the test, with the UAV starting in the hover behind the GV. The control system begins to move the aircraft toward the GV a few seconds before the GV starts to accelerate. At the point where a visual lock is acquired, the speed data become significantly more noisy, with rapid changes in the estimated speed at the point when the vision system acquires and loses lock. During the final descent phase however, as shown in Figure 25, the speeds are shown to match well and, although noisy, track relatively closely. As mentioned before, the control system

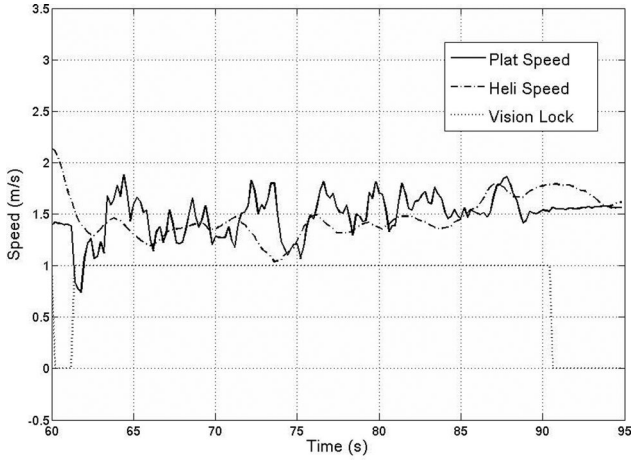

Figure 25. UAV and GV speed (target 1.5 m/s) for the final approach together with the vision lock indicator.

is designed to react relatively slowly to changes in the GV speed, and as such there is an implicit requirement in this current configuration that any accelerations that the GV undergo are relatively small.

The landing track is given in Figures 26 and 27. The northeast plot in Figure 26 shows that the SR20 started off with a significant error in the lateral axis relative to the body axis of the SR20, however this was minimized relatively quickly with the final descent phase tracking the ground vehicle closely. After the final loss of visual lock, the SR20 continued on the last known trajectory at the last demanded speed and successfully landed on the moving target approximately 0.3 m from the target, as shown in Figures 28

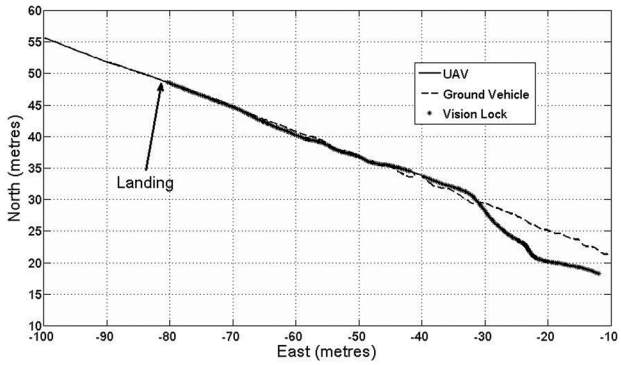

Figure 26. Top view: final approach showing the GPS position of both the SR20 and the GV.

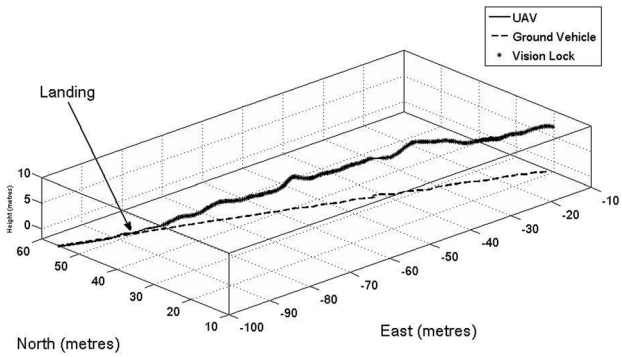

Figure 27. 3D view: final approach showing the GPS position of both the SR20 and the GV.

Figure 28. Post landing: front view.

and 29. In Figure 28, the camera used for localization can be seen mounted to the platform and suspended between the undercarriage skids. Note that what is being shown in Figures 26 and 27 are the GPS positions for both vehicles.

Figure 29. Post landing: side view.

Table III. Flight tests performed.

	Number of flights	Purpose
Initial testing	6	operational development
AFCS interfacing	4	control system testing
Vision tracking	2	RAPiD evaluation
Stationary target approach	30	including over 50 landings
Speed response	2	modeling SR20 response
Target tracking	2	testing the speed controller
Visual tracking	10	testing the lateral controller
Moving target approach	14	including five landings

Two additional points should be noted. First, for simplicity during landing, the SR20 UAV orientation was kept constant throughout the flight and independent of the landing platform direction of movement. Second, during this test there was a cross headwind of approximately 15 kts from 30 degrees to the right of the ground vehicle direction of travel. Test results found that the control system response was repeatable and relatively insensitive to the atmospheric conditions within these limits. The ground trajectory was therefore defined by the available track, and not by the wind direction.

In total for the project, there were over 50 successful landings onto stationary targets and five onto moving platforms. Provided the vision lock was solid, stationary target landings were highly repeatable, and in one series of tests, multiple successful consecutive touchdowns were made on each flight. With regard to the moving target, these were less repeatable and highly dependent on variations in lighting and the initial conditions. As can be see in Table III, one in three resulted in a successful touchdown. It is important to note that one of the key reasons for

Figure 30. Descent onto a UGV.

Figure 32. UGV landing.

Figure 31. UAV final approach.

failure during the landing approach was the susceptibility of the camera and tracking software to changes in the ambient conditions, such as light levels. For example, if the sun was covered by cloud during a run, this would lead to rapid changes in light level and subsequent loss of tracking. Once the initial acceleration of both vehicles had taken place, and with visual lock acquired by the air vehicle, the system was seen to be repeatable, with failures arising from tracking loss and not from the accuracy of the vehicle tracking or the approach taken.

The flights themselves were often constructed for more than one purpose, but in Table III an indication of the approximate number of flights for each has been given. In addition to these, there were a number of flights carried out at Roke Manor. One of these is shown in Figures 30–32, where an SR20 with an identical setup to that described in this paper automatically lands on an unmanned ground vehicle (UGV).

5. CONCLUSIONS

This paper has presented selected results from a systematic series of flight trials whose objective was to automatically land a UAV onto a moving landing area using vision tracking for localization. The first step was to perform autonomous landings on a stationary landing area, a capability demonstrated to be robust and repeatable. This built on previous work and was taken forward to a moving target, with a landing protocol established using GPS for initial guidance and visual tracking for the final stages. The tests were done in the field and subject to the associated environmental limitations, such as wind and lighting conditions.

The Autoland system developed through the SEAS DTC Integrated Control Demonstration has shown that vision is a viable contender as a sensor for localization in unmanned vehicles. The resulting closed-loop system has improved the tracking precision of the candidate air vehicle by an order of magnitude and enabled the precise requirements for an automatic landing on a moving target to be achieved. Landing on a moving platform has been shown to be possible, and a practical solution to the problem of automated UAV recovery. This capability has a wide range of possible different civil and military applications, and work is ongoing to extend the system to deal with unknown landing sites using a "click to land" capability.

The main challenges identified during the flight tests included the influence of the ground effect on landing precision, as noted in the literature. This was more noticeable when performing landings on stationary targets. The second was the effect that significant variations in light levels had on the performance of the vision tracker. It is expected, however, that with additional work, the combination of camera and tracker could be refined and made more robust to changes in the local lighting conditions. As noted within the literature, it was found that within the limits given, the wind speed had little impact on the landing performance.

Failure in a landing approach was seen to be primarily due to the loss of visual lock, or variations in the initial starting conditions of both vehicles. Once visual tracking had been initiated and with the environmental conditions constant, the landing approach was found to be relatively robust and repeatable. As with all tests in the field, simulation, including a sensor in the loop with the visual tracker, was found to greatly reduce the need for flight tests.

These tests have also dealt primarily with linear motion of the ground platform. Further work will look at extending the controller to consider rapid variations in the heading and speed of the landing platform. In this case, a more responsive and agile air vehicle would be required and could increase the envelope in which landings are performed. Finally, vision lock is currently lost at approximately 30 cm above the target. Thus, the final touchdown is carried out along a predetermined open-loop path. Current and future developments include moving the camera to provide tracking up to the point of contact, and the inclusion of sensors to indicate positive contact with the landing platform.

NOMENCLATURE

D = vertical displacement in the global reference frame
E = displacement east in the global reference frame
N = displacement north in the global reference frame
r = demanded UAV ground speed
U = Component of V_t along the x-body axis
U_{GV} = GV ground speed
U_{UAV} = UAV ground speed
V = Component of V_t along the y-body axis
W = Component of V_t along the z-body axis
x_c = vertical displacement in the camera reference frame
X = longitudinal displacement in the UAV body reference frame
X_{GV} = vertical displacement in the GV body reference frame
y_c = lateral displacement in the camera reference frame
Y = lateral displacement in the UAV body reference frame
Y_{GV} = lateral displacement in the GV body reference frame
z_c = longitudinal displacement in the camera reference frame
Z = vertical displacement in the UAV body reference frame
Z_{GV} = longitudinal displacement in GV body reference frame

ACKNOWLEDGMENTS

The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre.

REFERENCES

Amidi, O. (1996). An autonomous vision-guided helicopter. Ph.D. thesis, Robotics Institute, Carnegie Mellon University.

Barber, B., McLain, T., & Beard, R. (2009). Vision based precision landing of fixed-wing mavs. Journal of Aerospace Computing, Information, and Communication, 6(3), 207–226.

Barber, D. B., Griffiths, S., McLain, T. W., & Beard, R. W. (2007). Autonomous landing of miniature aerial vehicles. AIAA Journal of Aerospace Computing, Information, and Communication, 4(5), 770–784.

Bryson, A. (1994). Control of spacecraft and aircraft. Princeton University Press.

De Wagter, C., & Mulder, J. (2005, Aug). Towards vision-based UAV situation awareness. In AIAA Guidance, Navigation, and Control Conference, 15–18.

Evans, R. (1990). Kalman filtering of pose estimates in applications of the rapid video rate tracker. In Proceedings of the British Machine Vision Conference, Oxford, UK.

Garratt, M. A., Pota, H. R., Lambert, A., Eckersley-Maslin, S., & Farabet, C. (2009). Visual tracking and lidar relative positioning for automated launch and recovery of an unmanned rotorcraft from ships at sea. Naval Engineers Journal—The Official Journal of the American Society of Naval Engineers, 121(2), 1559–3584, online 0028–1425.

Hamel, T., Mahony, R., & Russotto, F.-X. (2010). The landing problem of a vtol unmanned aerial vehicle on a moving platform using optical flow. In IEEE International Conference on Intelligent Robots and Systems, Fontenay-aux-Roses, France.

Harris, C., & Stennett, C. (1990). Rapid—A video rate object tracker. In Proceedings of the British Machine Vision Conference, Oxford, UK.

Iran, M., & Anandan, P. (2000). About direct methods. Vision Algorithms: Theory and Practice, 1883, 267–277.

Kendoul, F. (2012). Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. Journal of Field Robotics, 29(2), 315–378.

Lenge, S., Sunderhauf, N., & Protzel, P. (2008, November). Autonomous Landing for a Multirotor UAV Using Vision. In Workshop Proceedings of SIMPAR 2008 Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots, 482–491.

Marconi, L., Isidori, A., & Serrani, A. (2002). Autonomous vertical landing on an oscillating platform: An internal-model based approach. Automatica, 38, 21–32.

Merz, T., Duranti, S., & Conte, G. (2004). Autonomous landing of an unmanned helicopter based on vision and inertial sensing. In International Symposium on Experimental Robotics.

Merz, T., Duranti, S., & Conte, G. (2006). Autonomous landing of an unmanned helicopter based on vision and inertial sensing. Experimental Robotics IX, Springer Tracts in Advanced Robotics, 21, 343–352.

Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27, 1615–1630.

Mondragn, I. F., Campoy, P., MartÃ-nez, C., & Olivares-MÃ©ndez, M. A. (2010). 3d pose estimation based on planar object tracking for uavs control. In ICRA.

Nonami, K., Kendoul, F., Suzuki, S., Wang, W., & Nakazawa, D. (2010). Autonomous flying robots: Unmanned aerial vehicles and micro aerial vehicles (chap. 10, pp. 219–250), Springer.

Oh, S.-R., a d Sunil, K., Agrawal, K. P., Pota, H. R., & Garratt, M. (2005). Autonomous helicopter landing on a moving platform using a tether. In IEEE International Conference on Robotics and Automation, Barcelona, Spain.

Oh, S.-R., a d Sunil, K., Agrawal, K. P., Pota, H. R., & Garratt, M. (2006). Approaches for a tether guided landing of an autonomous helicopter. In IEEE Transactions on Robotics and Automation, 22(3), 536–544.

Sabatini, R., & Palmeri, G. (2008). Differential global positioning system (dgps) for flight testing. NATO Research and Technology Organisation, October.

Saripalli, S., Montgomery, J. F., & Sukhatme, G. S. (2002). Vision-based autonomous landing of an unmanned aerial vehicle. In Proceedings of the IEEE International Conference on Robotics and Automation.

Saripalli, S., & Sukhatme, G. (2003). Landing on a moving target using an autonomous helicopter. In Proceedings of the International Conference on Field and Service Robotics, Mt. Fuji, Japan.

Scherer, S., Chamberlain, L., & Singh, S. (2012). Autonomous landing at unprepared sites by a full-scale helicopter. Robotics and Autonomous Systems, 60(12), 1545–1562.

Shakernia, O. R. V., Sharp, C., Ma, Y., & Sastry, S. (2002). Multiple view motion estimation and control for landing an unmanned aerial vehicle. In Proceedings of the International Conference on Robotics and Automation.

Systems Engineering for Autonomous Systems Defence Technology Centre, http://www.seasdtc.com/(2010). Checked March 25, 2012.

Thales (2011). http://www.thalesgroup.com/Press_Releases/Markets/Aerospace/2011/20110614_-_Thales_completes_successful_flight_demonstration_for_full_automatic_landing_and_deck_landing_of_one-ton_class_rotary_wing_UAVs/.

Unmanned Aircraft Systems Roadmap–DoD (2005). Retrieved March 25, 2012, from http://www.fas.org/irp/program/collect/uav_roadmap2005.pdf.

Wenzel, K., Masselli, A., & Zell, A. (2011). Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle. Journal of Intelligent and Robotic Systems, 61, 221–238.