

# Sensing for Visual Homing

Kane Usher<sup>a,b</sup>, Matthew Dunbabin<sup>a</sup>, Peter Corke<sup>a</sup>, Peter Ridley<sup>b</sup>

<sup>a</sup> CSIRO Manufacturing and Infrastructure Technology  
P.O. Box 883

Kenmore 4069, Queensland, Australia

<sup>b</sup> School of Mechanical, Manufacturing and Medical Engineering, Queensland University of Technology  
Brisbane 4000, Queensland, Australia

Email: Kane.Usher@csiro.au

## Abstract

In this paper, we outline the sensing system used for the visual pose control of our experimental car-like vehicle, the Autonomous Tractor. The sensing system consists of a magnetic compass, an omnidirectional camera and a low-resolution odometry system. In this work, information from these sensors is fused using complementary filters. Complementary filters provide a means of fusing information from sensors with different characteristics in order to produce a more reliable estimate of the desired variable. Here, the range and bearing of landmarks observed by the vision system are fused with odometry information and a vehicle model, providing a more reliable estimate of these states. We also present a method of combining a compass sensor with odometry and a vehicle model to improve the heading estimate.

## 1 Introduction

The CSIRO Autonomous Tractor (AT) is a ride-on mower which has been retro-fitted with an array of actuators, sensors, and a computer system enabling the implementation and testing of control and navigation algorithms. This paper outlines the Autonomous Tractor's sensing system and how information from disparate sources is 'fused' to reduce noise in the desired measurement variables. The aim of the sensing system is to provide enough information to enable the vehicle to stabilise to some pre-learned target pose, based upon the discrepancies between the current view of the workspace, and that seen at the target pose (see [Usher *et al.*, 2002b]).

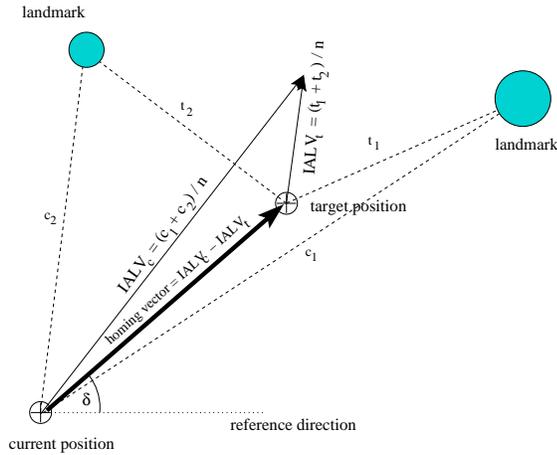
Much research in ground-based mobile robotics has focussed on the problem of *localisation* in which the robot's position is estimated with reference to some 'map', with the map being provided *a priori* or learnt on-line, as in Simultaneous Localisation and Mapping. There are, in general, three approaches to localisation and the inter-related problem of mapping [Thrun, 1998]: grid-based methods, feature-based methods, and topological approaches. Grid-based techniques represent the robot's environment with a matrix of cells, each of which is assigned a probabil-

ity of being occupied by an object. Matching the current 'local' grid map to some global representation provides a means of estimating the robot's position on the map. The feature-based methods use identifiable objects in the environment parameterising them with reference to colour, width, length and position. These methods usually represent the state of the robot, and of the features in the environment with an estimated state-vector and a covariance matrix. An extended Kalman Filter is typically used to track the pose of the robot, with predictions of the robot's future pose provided by odometry and a vehicle model. By periodically sensing features in the environment, and matching these to previously mapped features, the robot-relative feature location can be used to estimate the state vector and decrease its covariance. Topological methods rely on the recognition of a series of 'distinctive locales', each of which is linked through directions to reach other distinctive locales.

Our work has focussed on highly local behaviours. We aim to use these local behaviours in a topological navigation framework. In particular, we have looked extensively at pose stabilisation and the related problem of visual homing. As observed by Kelly and Nagy [Kelly and Nagy, 2002], much of the work in pose stabilisation is highly theoretical, with few instances of real functioning systems. On the other hand, there are many examples of functioning visual homing systems, but they are usually implemented on robots with quite simple kinematics and the constraint on attaining a particular target orientation is relaxed (i.e. position as opposed to pose stabilisation).

### 1.1 The Improved Average Landmark Vector

Most visual homing systems use panoramic visual sensing in combination with a compass sense. They rely on the differences between the extracted landmark bearings at the current and target views to derive a vector which drives the robot towards 'home'. An elegant, correspondence free, homing method developed from hypotheses on how desert ants might use visual piloting is the Average Landmark Vector model. An ALV for any particular position in the workspace is found by summing unit vectors towards all



**Figure 1:** Illustration of the the IALV method for two landmarks in a workspace. The IALV’s are found by adding the vectors to the individual landmarks, and dividing the resulting vector by the number of landmarks,  $n$ . The home vector is then calculated by subtracting the target IALV from the current IALV.

currently visible landmarks and dividing by the number of landmarks. By matching the current ALV with a pre-stored ALV of the target location, a homing vector can be formed which drives the agent (robot) towards the target location [Lambrinos *et al.*, 2000]. In order to consistently add the vectors in the ALV model, an absolute reference direction is required, and, unless apparent size information is incorporated, a minimum of three landmarks is needed.

The nature of our sensor, an omnidirectional vision system, led us to investigate improvements to the ALV method. In its original form, the ALV method required the bearings to landmarks only. Range information could be incorporated, in a scaled manner, by including landmark apparent size, and slight improvements to the performance could be made. However, we have found that by including range information directly, a significant improvement is made and in fact, the distance and angle to the goal are yielded directly — we call this the Improved Average Landmark Vector [Usher *et al.*, 2002a]. In addition, the minimum required landmarks is reduced to one. An example of the IALV method is shown in Fig. 1. In essence, the IALV method is equivalent to finding a position relative to the centroid of the landmarks in the workspace.

As with the ALV method, the IALV method is purely sensor based. Landmark bearings are readily ascertained with an omnidirectional camera. If a flat-earth assumption is made, range information can be derived from an omnidirectional camera image through the geometry of the camera/mirror optics, as described in [Usher *et al.*, 2002a]. Alternatively, optic flow techniques could be used to determine landmark range [Chahl and Srinivasan, 1997], but we have found similar techniques to be too susceptible to noise given the poor odometry information on the AT.

One of the advantages of the ALV, and hence the IALV method, is that knowledge of a target location is contained within a single quantity. This reduces the need for complex map-like representations of the environment and is well suited for a topological navigation method, (see e.g. [Gaspar *et al.*, 2000; Kuipers and Byun, 1991]). Additionally, landmarks need not be unique, and the need for landmark correspondence is also bypassed. Many of the other homing algorithms require that the landmarks in the current image be matched with those at the target location, usually by minimisation of the sum of the bearing differences (see e.g. [Weber *et al.*, 1999]). If landmarks are occluded or missing, these methods can fail. Of course, like all sensor-based techniques, this method has a finite catchment area, limited by the omnidirectional sensor’s range and, in addition, has the potential to suffer from perceptual aliasing, or in a similar sense, the local minima problem.

The homing vector provided by the IALV method can be used to drive the agent towards home but does not provide a means of guaranteeing a final orientation. However, the quantities derived from the IALV can easily be converted to the states required by, for example, a pose stabilisation algorithm, as demonstrated in [Usher *et al.*, 2002b].

## 1.2 Paper outline

The remainder of this paper describes the sensing system used to extract the IALV and the steps taken to improve the quality of this information by fusing information from disparate sensing sources. Section 2 outlines the system architecture, Section 3 describes how we filter the AT’s heading sensor; Section 4 describes the AT’s vision system and how we track landmarks, estimate their range, and how we fuse this information with the AT’s odometry system; and Section 5 concludes the paper.

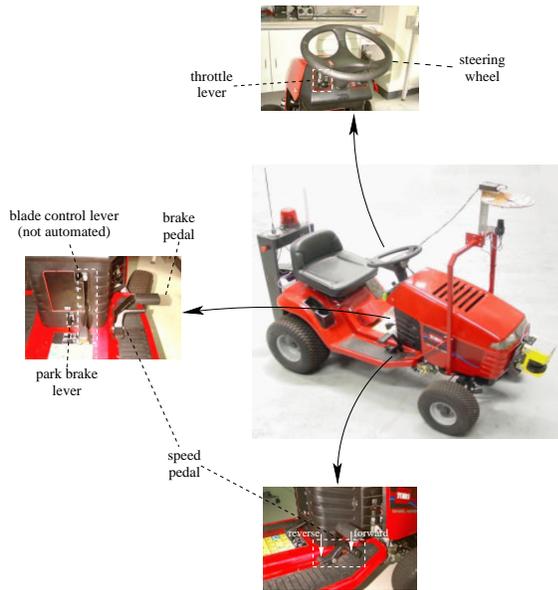
## 2 System Architecture

The Autonomous Tractor, see Figure 2, is a ride-on mower which has been retro-fitted with an array of actuators, sensors, and a computer system enabling the implementation and testing of control and navigation algorithms.

### 2.1 System design

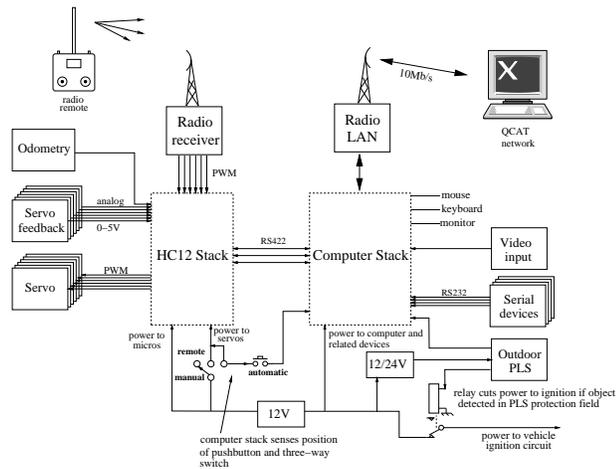
The vehicle’s design is such that it can be operated in three modes: *manual*, *remote*, or *automatic*. Manual operation is the AT’s original mode of operation in which the vehicle is driven by an operator. Remote mode allows a user to control the vehicle from a hand-held radio-transmitter. Automatic mode allows the on-board computer to control the vehicle. There are six axes of control: speed pedal, throttle lever, brake pedal, park brake lever, steering wheel and steering engage. The steering engage axis allows the steering actuator to disengage from the system, allowing the vehicle to be driven manually.

The vehicle can operate as a stand-alone unit but can also communicate to a network of computers through a



**Figure 2:** The experimental platform. Note the omnidirectional camera mounted over the front wheels and the box at the rear which houses the control and computer system.

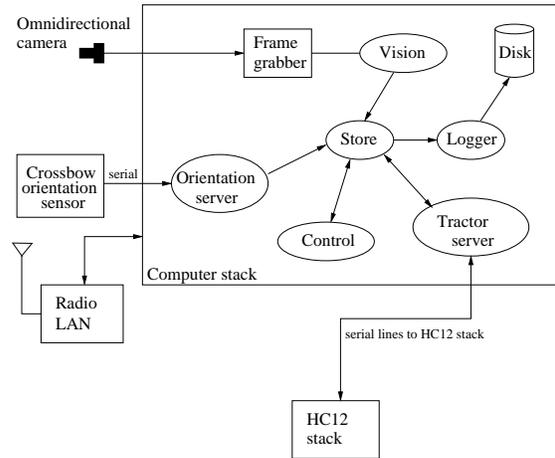
wireless LAN connection. However, all control and computing occurs on-board. The vehicle's control architecture is shown in Figure 3.



**Figure 3:** The AT's control system.

There are two computer systems on-board, the *HC12-stack* and the *computer-stack*. The HC12-stack acts as an interface to the AT's actuators and low level sensors (actuator positions and odometry). It is based upon a set of in-house developed HC12 microprocessor and power driver boards. The HC12 and computer-stacks communicate via standard RS232 serial links. The computer stack is based upon a Crusoe 800 CPU running the LINUX operating system. The computer-stack includes a solid-state

disk, frame-grabber and 8-way serial port. The computer also handles the logic for allowing computer inputs to the control system by monitoring a safety card which senses the requested state of the system (manual, remote or automatic). When in automatic mode, each individual axis of control can be switched from remote or computer-sourced demands. At the heart of the software system is the 'store' which allows data to be exchanged between individually running processes [Roberts *et al.*, 1999]. Figure 4 illustrates the communication between processes running on the computer, sensors and the HC12 stack.



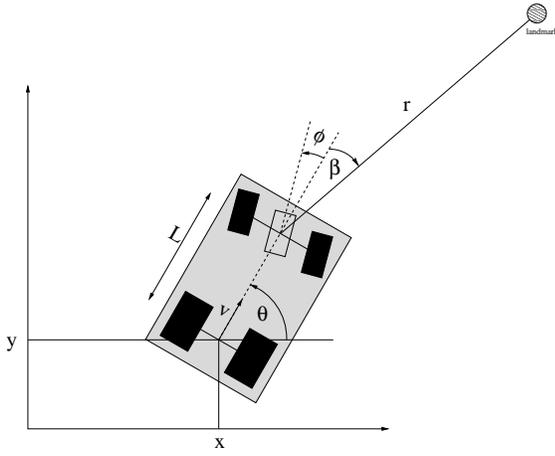
**Figure 4:** The AT's software structure.

## 2.2 Sensors

The AT can be equipped with an array of sensors. Currently fitted to the vehicle are:

- Omnidirectional camera (EyeSee 360) mounted over the front wheels.
- Crossbow high speed orientation sensor (3 axis accelerometer + 3 axis magnetometer).
- SICK PLS for emergency collision avoidance only.
- Differential GPS (not currently used).
- Vehicle speed and steering angle.

The vehicle's speed is measured with a low-resolution quadrature encoder which measures the rotation of the front left hand wheel. This system gives measurements of vehicle speed at 2Hz with a resolution of approximately  $0.035 \text{ ms}^{-1}$  – correction for the mounting location has been found to be unnecessary. Steering angle is measured with an absolute encoder. For visual homing, the primary sensors used are the omnidirectional camera, compass and odometry.



**Figure 5:** The vehicle and the coordinate system used. All angles are counter-clockwise positive. Also shown is a landmark and the AT relative bearing and range measurement.

### 3 Heading Estimation

Heading information is critical to the calculation of the IALV discussed in Section 1. Vehicle orientation is sensed with a Crossbow CXM543 High Speed Orientation Sensor. The azimuth reading is filtered within the Crossbow unit, and corrected for roll and pitch. Due to the vibratory environment, and the presence of EM fields produced by the vehicle's actuators, the vehicle's engine and alternator, the unit's azimuth reading can be extremely noisy, varying by as much as  $\pm 20^\circ$  when the vehicle's engine is running, even when the vehicle is stationary – see the raw azimuth readings plotted in Figure 7 for an example.

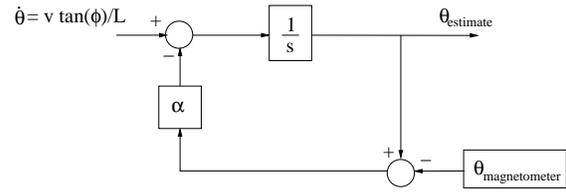
#### 3.1 Complementary filtering

To combat noise and improve the accuracy of the measured variable, we have combined the azimuth reading from the Crossbow unit with an estimate of the vehicle's angular rate of rotation [Buskey *et al.*, 2003]. Referring to Figure 5, estimates of the vehicle's angular rate are given by the well-known kinematic equation of the angular rotation rate of a car-like vehicle:

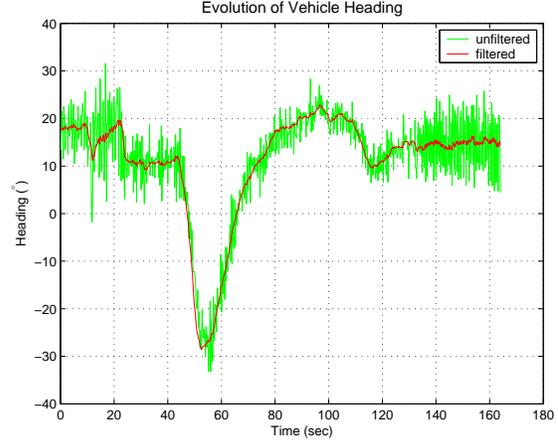
$$\dot{\theta} = \frac{v \tan \phi}{L} \quad (1)$$

where  $v$  is the velocity of the vehicles rear-axle midpoint,  $\phi$  is the steering angle and  $L$  is the distance between the front and rear axles.

Figure 6 illustrates the complementary filter used for heading estimation, while Figure 7 shows the result of the application of the filter on real AT data, along with the original raw azimuth reading. As Figure 7 illustrates, the complementary filtering is highly successful at rejecting noise in the system with very little phase lag. However, in practice we can extract a much 'cleaner' azimuth reading by using the raw magnetometer readings from the Crossbow unit, rather than the azimuth signal which has been



**Figure 6:** The complementary filter on vehicle orientation.



**Figure 7:** Comparison of raw vehicle heading measurement and filtered version.

corrected for roll and pitch using the accelerometer readings. This of course assumes that the AT is operating in a relatively flat region, which is one of the assumptions used in the vision system in any case.

### 4 Vision

Our vision system is designed to track colour objects based upon a pre-learnt, look-up table representation of their colour. When testing our visual pose control systems, we use red road cones (also known as witches hats) as our landmarks. The system uses the YCrCb colour space and relies on a two-dimensional look-up table on the desired Cr and Cb values. After training the system for a particular colour or group of colours, colour segmentation proceeds as described in Algorithm 1. Figure 8 is an example image from the camera/mirror system mounted to the AT, while Figure 10 illustrates the results of the segmentation process described in Algorithm 1, applied to the original image of Figure 8.

After segmentation, the colour objects are tracked over time in an effort to reduce the effects of incorrect image segmentation. This *temporal* filtering process is described in Algorithm 2.

The system is designed to enable the use of multiple lists, and hence tracking of different groups of colour objects. Each list can use its own set of parameters for promotion and demotion etc.. In our case, these different types of objects will be landmarks and obstacles.

---

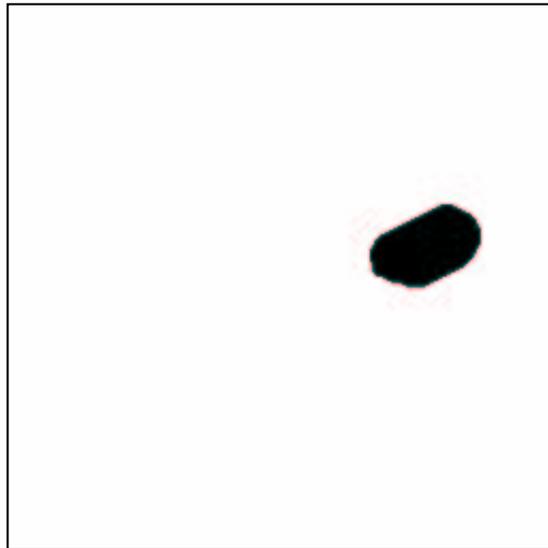
**Algorithm 1** Colour segmentation of an image.

---

- 1: Perform look-up using current image and 2-d look-up table of the desired Cr and Cb values
  - 2: Extract blobs from the resulting binary image i.e. perform image labelling
  - 3: Eliminate blobs which are too large or too small
  - 4: Calculate individual blob properties
  - 5: Find blob  $(x,y)$  point closest to centre of image
  - 6: Calculate blob pixel radius and bearing w.r.t image centre
  - 7: Estimate blob range using geometric model of camera/mirror and the flat-Earth assumption
- 

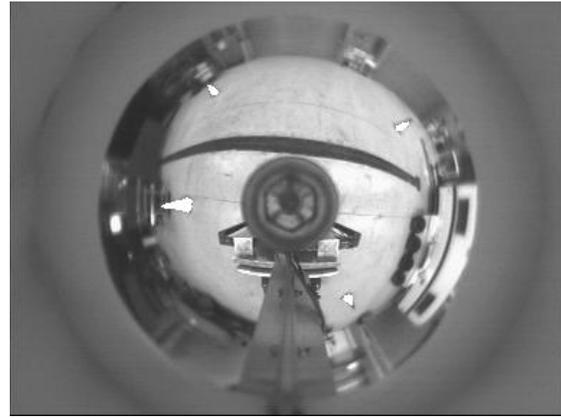


**Figure 8:** An example image from the omnidirectional camera



**Figure 9:** An example lookup table on the Cr and Cb values of the red witches hats' contained in the image of Figure 8. The image has been inverted for clarity, the dark 'blob' is the set of 'ON' Cr and Cb values.

The temporal filtering is based upon a vehicle/beacon



**Figure 10:** Image resulting from segmentation. The input image used was that shown in Figure 8, while the colour lookup table used was that shown in Figure 9.

---

**Algorithm 2** Temporal filtering

---

- 1: Grab the first image
  - 2: Extract the beacons from the image
  - 3: Boot strap the list with the first set of beacons
  - 4: **loop**
  - 5:   Grab next image
  - 6:   Extract beacons from image using Algorithm 1
  - 7:   Match beacons in list to current beacons,
  - 8:   **if** no match **then**
  - 9:     add another node to the list
  - 10:   **end if**
  - 11:   Upgrade relevant nodes to 'good' status, based on times seen and when last seen
  - 12:   Demote relevant nodes to 'bad' status, based on when last seen
  - 13:   Eliminate old nodes, based on when last seen
  - 14:   Perform complementary filtering using vehicle odometry
  - 15:   Write properties of 'good' beacons to the STORE for use by other processes
  - 16: **end loop**
- 

relative motion model. Referring to Figure 5, the equations for the motion of an individual beacon are (for a car-like vehicle):

$$\dot{r} = -v \cos(\beta) \quad (2)$$

$$\dot{\beta} = \frac{v \sin(\beta)}{r} - \frac{v \tan(\phi)}{L} \quad (3)$$

where  $r$  is the beacon's ground-plane range relative to the AT,  $v$  is the vehicle's velocity,  $\beta$  is the relative orientation of the beacon with respect to the AT,  $\phi$  is the vehicle's steering angle, and  $L$  is the length of the vehicle.

Thus, if we know or measure  $v$ ,  $\phi$  and  $L$ , we can calculate  $\dot{r}$  and  $\dot{\beta}$  and hence predict future values of  $r$  and  $\beta$ , through simple Euler integration. In practice, we use this

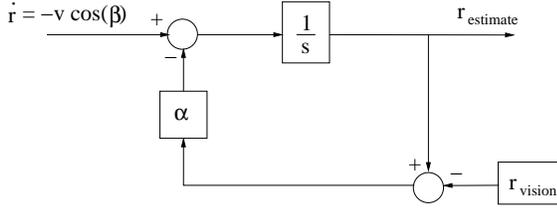


Figure 11: The complementary filter on landmark range.

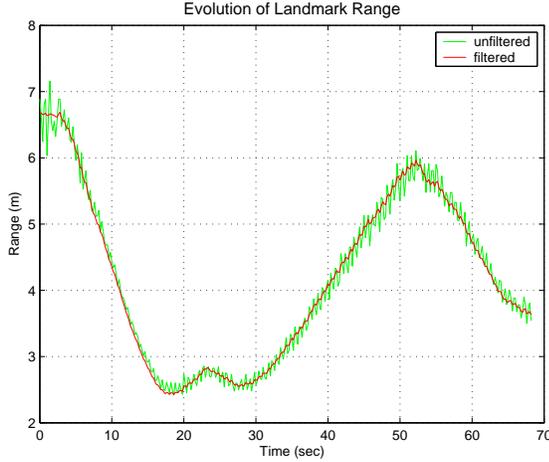


Figure 12: Evolution of the landmark range with respect to the AT as the vehicle moves through a workspace.

method to predict future beacons positions, adding a tolerance to account for noise. This allows us to get ‘correspondence’ between tracked points but this correspondence is not a necessary feature of our system, it just helps with noise reduction in the event of falsely detected beacons resulting from poor image segmentation. For an example of the temporal filtering process, refer to the raw range and bearing results in Figures’ 12 and 14, which show tracking of a witches hat over a period of approximately 70s.

#### 4.1 Complementary filtering

As described earlier, the idea of a complementary filter is to fuse the complementary features of different sensing sources to produce a more accurate measurement of the desired quantity. In the case of *range* estimation, we have an estimate of range gleaned directly from the vision system. We also have an estimate of how a particular objects range should change based on the motion of the vehicle and the relative orientation of the landmark, as given by Equation 2. The range determined from vision is then combined with this rate of change of range measurement as shown in Figure 11. Some representative results of this process are given in Figure 12. For these results, the gain was set equal to 1.4 at a vision sampling rate of 5 Hz. In practice we vary the gain parameter with the velocity of the vehicle.

Similarly, for the bearing estimate of an object we can

combine the bearing angle from vision, with the rate of change of bearing angle given by Equation 3, as shown in Figure 13. The results of this process are given in Figure 14. Here we see that although the data has been smoothed somewhat, the filtering process on the bearing measurement actually gives a phase lead to the estimate. This is probably due to the fact that the measurements of  $\beta$  and  $\dot{\beta}$  are coupled somewhat.

In both cases we compared this technique with using a Butterworth filter but found that the complementary filtering has far superior performance in terms of noise reduction and phase lag. We also applied an Extended Kalman Filter on the combined data (range, bearing and odometry) but found that tuning and software implementation was less favourable against the single parameter complementary filter.

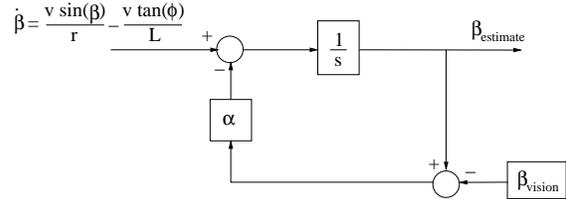


Figure 13: The complementary filter on AT relative landmark bearing.

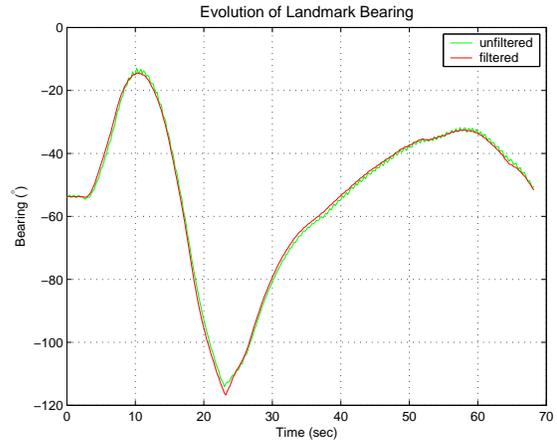


Figure 14: Evolution of the landmark bearing with respect to the AT as the vehicle moves through a workspace. Tracking for this filter is not as good and in fact leads the original signal. This is due to coupling in the filtering equations as  $\beta$  appears in the calculation of  $\dot{\beta}$ .

## 5 Conclusions

In this paper we have presented the software and hardware architecture of our experimental car-like vehicle, the Autonomous Tractor. We have highlighted the sensing system used for the task of visual pose control and described

how we fuse information from vision and odometry to obtain better estimates of the range and bearing to objects in the environment. We have presented experimental results highlighting the effectiveness of this ‘complementary’ filtering technique. These results show that for the sensing arrangement presented in this paper, the complementary filtering of the range estimate vastly reduces noise in the system. For the landmark bearing estimate, the data is smoothed but a phase lead is introduced, believed to be due to coupling in the measurements given to the filter.

## Acknowledgements

The authors would like to thank the automation team for their invaluable assistance. In particular, the technical support of the CMIT Tractor Team — Peter Hynes, Stuart Wolfe, Stephen Brosnan, Graeme Winstanley, Jonathon Roberts, Pavan Sikka, Elliot Duff, Les Overs, Craig Worthington and Steven Hogan — is gratefully acknowledged. Jonathon O’Brien at UNSW is gratefully thanked for the loan of the Toro ride-on mower.

## References

- [Buskey *et al.*, 2003] G. Buskey, J. Roberts, P. Corke, P. Ridley, and G. Wyeth. Sensing and control for a small-size helicopter. In B. Siciliano and P. Dario, editors, *Experimental Robotics*, volume VIII, pages 476–487. Springer-Verlag, 2003.
- [Chahl and Srinivasan, 1997] J. S. Chahl and M. V. Srinivasan. Range estimation with a panoramic visual sensor. *Journal of the optical society of America*, 14(9), September 1997.
- [Gaspar *et al.*, 2000] Jose Gaspar, Niall Winters, and Jose Santos-Victor. Vision-based navigation and environmental representations with an omnidirectional camera. *IEEE Transactions on Robotics and Automation*, 16(6):890–898, December 2000.
- [Kelly and Nagy, 2002] A. Kelly and N. Nagy. Reactive nonholonomic trajectory generation via parametric optimal control. *International Journal of Robotics Research*, 22(7/8):583–602, 2002.
- [Kuipers and Byun, 1991] Benjamin Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8:47–63, 1991.
- [Lambrinos *et al.*, 2000] Dimitrios Lambrinos, Ralf Möller, Thomas Labhart, Rolf Pfeifer, and Rudiger Wehner. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30:39–64, 2000.
- [Roberts *et al.*, 1999] J.M. Roberts, P.I. Corke, R.J. Kirkham, F. Pennerath, and G.J. Winstanley. A real-time software architecture for robotics and automation. In *International Conference on Robotics and Automation*, pages 1158–1163, Detroit, Michigan, 1999. IEEE.
- [Thrun, 1998] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99:21–71, 1998.
- [Usher *et al.*, 2002a] Kane Usher, Peter Corke, and Peter Ridley. Home alone: Mobile robot visual servoing. In *International Conference on Intelligent Robots and Systems - Visual Servoing Workshop*, Lausanne, Switzerland, October 2002. IEEE/RSJ.
- [Usher *et al.*, 2002b] Kane Usher, Peter Corke, and Peter Ridley. Visual servoing of a car-like vehicle – an application of omnidirectional vision. In *Proceedings of the 2002 Australian Conference on Robotics and Automation*, Auckland, New Zealand, December 2002. published via CDROM.
- [Weber *et al.*, 1999] Keven Weber, Svetha Venkatesh, and Mandyam Srinivasan. Insect-inspired robotic homing. *Adaptive Behavior*, 7(1):65–97, 1999.