Docking for mobile robots

G.J. Ellwood Y.Zheng J.E.W.Mayhew
Artificial Intelligence Vision Research Unit
University of Sheffield, Sheffield, S10 2TN, England

Abstract

A method of vehicle control is described. Visual feedback from a micro-saccadic tracker is used to provide a docking competence. The vehicle is thereby capable of arriving at goal position with a specified heading angle. The vehicle uses a beacon to estimate it's trajectory and position in the ground plane. This information is used to control the vehicle to move to a "via-point". Complex trajectories can be achieved composed of multiple via-points. This is achieved without the need to plan the path between the via-points. The algorithm is therefore computationally cheap.

1 Introduction

The fundamental problem in navigation is to reach a goal position and orientation. When motion is confined to the ground plane this is defined by three parameters (x, y, α). Vehicles have only two degrees of freedom for control. These are the velocity and the curvature. This is a non holonomic system. In general this means that a pose cannot be achieved by a single command. Therefore several commands must be made for successful docking.

In most cases docking is achieved by *a priori* path planning. Many algorithms have been developed. Typical examples include splines [SG91] or Clothoid curves [KM86, KH89] In general these have been based on the dynamics of wheeled vehicles. Time-Optimal trajectories have been investigated [RP94] based on Pontryagin's maximum principle, and compare favourably with minimum distance algorithms. However, errors will accrue and some feedback [SBPM89] is necessary to continually check and update the vehicle motion. Any deviations from the planned path are then corrected.

It is proposed that docking be achieved using visual feedback alone. Such "on the fly" motion planning has been investigated [JSVG93, CR93] in the context of obstacle avoidance rather than attaining a final pose. Estimates of the target position with respect to the robot are provided by a micro-saccadic tracker. These are used to calculate the motion parameters of the robot. A steering angle is generated using the angle to the target and the rotation of the vehicle. This is updated at the vision sampling rate until the AGV docks with the target.

BMVC 1994 doi:10.5244/C.8.66

2 Docking Using Visual Feedback

The prime objective of the work is to generate a trajectory using visual feedback for a vehicle to achieve a target pose. This approach is similar to that developed by Kosko [Kos92], who developed a fuzzy system to simulate a truck backing up in a planar parking lot.

An estimate of the three dimensional position of the beacon is provided by a micro-saccadic tracker. The data are accurate to within $3cm$ in a $3m \times 3m \times 1m$ workspace at $6.25Hz$. These are used to estimate the motion parameters of the vehicle. The motion parameters R and T are found by solution of the equation

$$q = Rp + T \tag{1}$$

where q and p are the 3D coordinates of the beacon before and after the motion respectively. For motion in the ground plane

$$R = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}, T = \begin{bmatrix} t_x \\ t_z \end{bmatrix} \tag{2}$$

The parameters are estimated using the assumption [ZTB+91] that over the sampling interval the vehicle moves along a circular trajectory. A geometrical analysis shows that $t_x = t_z tan\frac{\alpha}{2}$ so the motion parameters can be found analytically. The three dimensional position of the target and the vehicle motion parameters are used to generate a steering command h. This is given by the equation

$$h = c\theta - o\phi \tag{3}$$

where θ is the angle to the target, or the course angle, and ϕ is the goal orientation. Vision data are used to estimate the virtual course angle θ. ϕ is found by taking the global rotation of the vehicle from the goal orientation angle. This is a very simple and computationally inexpensive algorithm for trajectory generation.

Using these algorithms the vehicle is capable of docking with a virtual target. That is, to use the beacon in order to reach some other goal position. The rotation parameters are generated as before using the beacon. Putting R and T into equation 1 where q is the goal position in global coordinates and p is the goal position with respect to the robot.

$$p = R^{-1}(q - T) = R^T(q - T) \tag{4}$$

p is used to estimate the virtual course angle θ. ϕ is found by taking the global rotation of the vehicle from the virtual goal orientation angle. Equation 3 is used to generate a steering angle. Using multiple virtual goals as "via-points" it is possible to generate complex paths, using only visual feedback to guide between them.

It is noted that there are two singularities in the docking algorithm. When $c\theta = o\phi$ a zero steering angle is returned. This is not considered important as after the next sampling interval this will not be the case unless $\theta = \phi = 0$. The other singularity occurs as the vehicle nears the goal. At this point θ, the course angle becomes unstable. This can be finessed always by using a virtual via-point as a docking position.

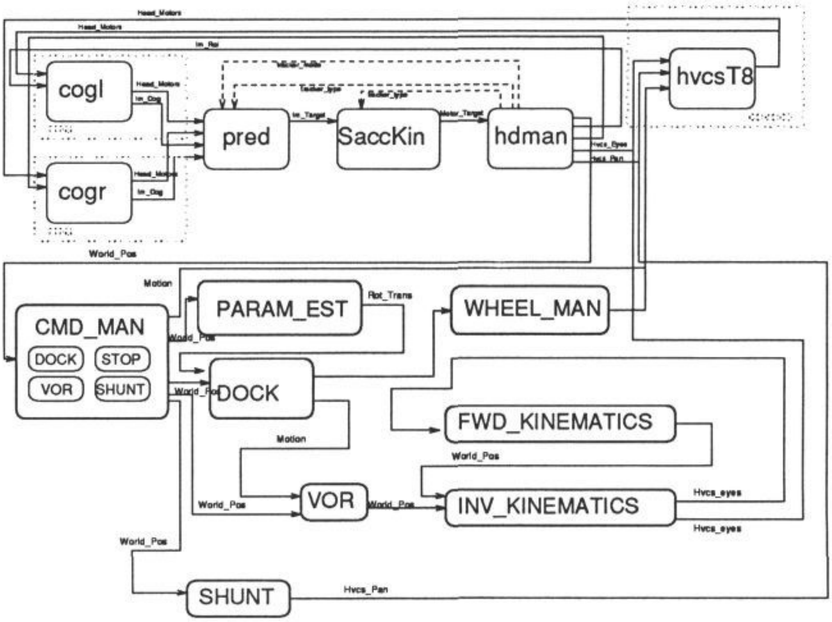

Figure 1: Vehicle Module Configuration

3 Implementation Using ANIT

The competences required for docking were developed under ANIT. This is a system deveolopment environment implemented in C. In brief, the philosophy of ANIT is as follows:

- The lowest level tasks are implemented first and these may be subsumed by higher level tasks.
- Modules may be described as either *experts* or *managers*. Essentially, managers determine what is to be done, and experts determine how.

The vehicle control modules receive 3D data from the visual competence provided by the tracker. The tracker is the lowest level task. The vehicle module configuration can be seen in figure 3.

A manager module CMD_MAN delegates this information to the appropriate module. The docking competence requires the expert modules DOCK and PARAM_EST in order to generate a steering signal. PARAM_EST estimates the motion parameters of the vehicle using the trajectory constraint algorithm described above. It communicates this message to DOCK. DOCK implements equation 3 to produce a steering angle and velocity. This is passed to the WHEELMAN module which converts the angle and velocity into wheel speeds in motor counts which are sent to the vehicle.

The SHUNT module continually nulls off the verge angles of the left and right cameras so the left and right verge motors can maintain foveation of the beacon.

4 Experiments

Experiments to investigate the performance of the algorithm were carried out on the COMODE robot vehicle. This a differential drive AGV with a 4 DOF head. The degrees of freedom are pan, tilt, and verge for each of the cameras. The head provides 3D information as to the position of a beacon. This is described in more detail in [JMZ93].

An experiment showing docking with a virtual target at 0^o at $(1500, 500, 0)$ can be seen in figures 2 and 3. The experiment was repeated with a docking angle of 90^o and the trajectory and heading angle can be seen in figured 4 and 5. Both are relatively smooth and appear to attain the goal angles with a reasonable degree of accuracy.

The effect of the orientation gain was examined. The vehicle docked with a virtual target of $(1500, 500, 0)$ at 90^o. Gains of 0.5, 0.333 and 0.2 were used. The trajectories and heading angles can be seen in figure 6 and 7. The smallest gain yielded what seemed like the most "expected" path, but it failed to attain the correct heading angle. The vehicle moved towards the target position but did not alter the orientaion appropriately until the course orientation was almost entirely nulled off. This is, of course, to be expected as the influence of the orientation is small. The vehicle reached the goal heading using the largest gain, but the trajectory did not appear to be in any sense "optimal". It is suggested that further work would involve adapting the gains as the vehicle nears the target. Initially the course gain should be large, but as the vehicle nears the target the orientation gain should increase.

Experiments using multiple via-points were carried out. The via-points were specified using the forward kinematics of the head to estimate their postion with respect to the starting position of the vehicle. The through points were at $(865, 409, 0)$, $(1336, 682, 0)$ and $(2023, 1251, 0)$ at 10^o, 30^o and 90^o respectively. These were marked on the floor. The vehicle was seen to move over the via-points and stop at the goal. The results can be seen in figures 8 and 9. The trajectory estimates appear to reverse at the points where the vehicle is turning sharply. This is because when the COMODE turns, the velocity of the robot origin is the average of the two wheel speeds. If the differential velocity is such that this is below zero, the origin of the robot coordinate frame will move backwards. The vision data for this experiment can be seen in figures 10 and 11. Both the depth and lateral distances are smooth. The depth parameter increases when the vehicle turns at the via points, indicating the origin of the robot coordinate frame moves backwards .

5 Conclusions

A docking algorithm for mobile robots has been described and implemented. The vehicle has been seen to dock with a beacon, to dock with a virtual target using the beacon to estimate it's motion and to use multiple via points to move along complex trajectories.

Adaptive orientaion gains will be investigated so the goal orientation is more significant as the vehicle nears the goal. A fuzzy system is currently being researched as a rule base for the gains for future implementation on the COMODE.

References

[CR93] David Coombs and Karen Roberts. Centering behavior using peripheral vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 440–445, June 1993.

[JMZ93] J. Porrill J.E.W. Mayhew and Y. Zheng. The identifcation and kinematic modelling of a 4 dof head. Technical report, AIVRU, 1993.

[JSVG93] F. Curotto J. Santos-Victor, G. Sandini and S. Garibaldi. Divergent stereo for robot navigation: Learning from bees. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 434–439, June 1993.

[KH89] Y. Kanayama and B.I. Hartman. Smooth local path planning for autonomous vehicles. In *Proceedings of IEEE Conference on Robotics & Automation*, pages 1265–1270, 1989.

[KM86] Y. Kanayama and N. Miyake. Trajectory generation for mobile robots. In *Robotics Research*, volume 3, pages 333–340, 1986.

[Kos92] Bart Kosko. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence.* Prentice-Hall, 1992.

[RP94] David B. Reister and Francois G. Pin. Time-optimal trajectories for robots with two independently driven wheels. *International Journal of Robotics Research*, 13(1):38–54, 1994.

[SBPM89] John Porrill Stephen B Pollard and John E W Mayhew. Experiments in vehicle control using predictive feed-forward stereo. In *Robotics Research, The Fifth International Symposium*, pages 173–180, 1989.

[SG91] Z. Shiller and Y. Gwo. Dynamic motion planning of autonomous vehicles. *IEEE Transactions on Robotics and Automation*, 7(2):241–249, 1991.

[ZTB$^+$91] Y. Zheng, N.A. Thacker, S.A. Billings, J.E.W. Mayhew, and J.P. Frisby. Ground plane motion parameter recovery and 3d segmentation using vehicle trajectory constraints. Technical report, AIVRU, 1991.

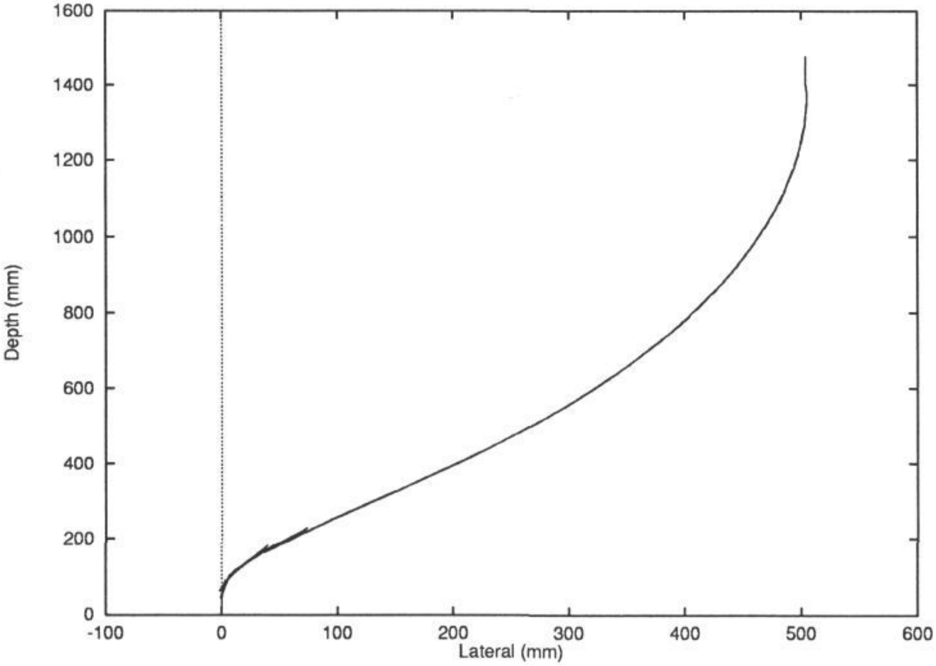

Figure 2: Docking with Virtual Beacon at 0 degrees

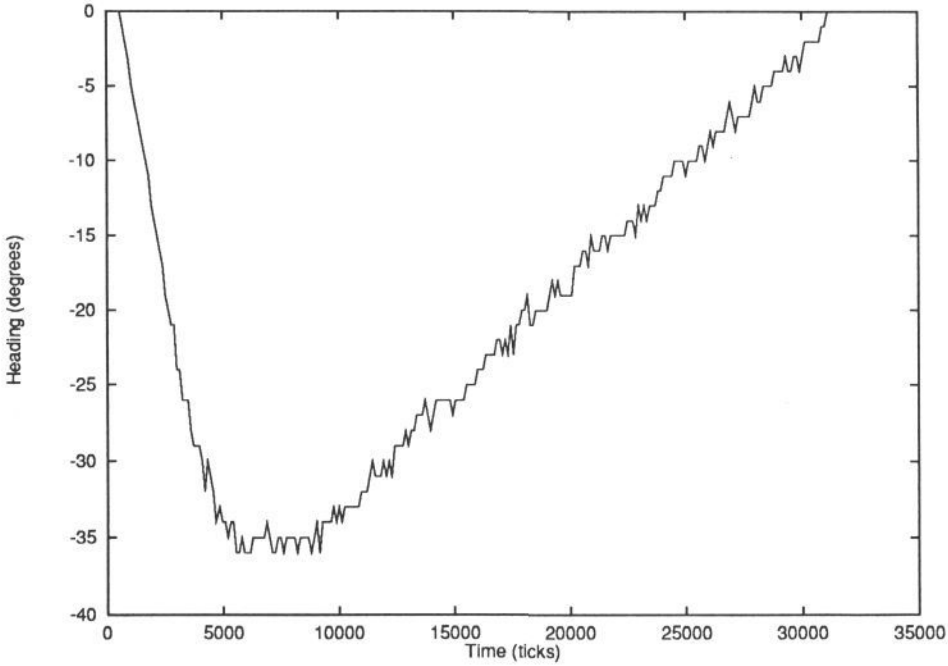

Figure 3: Heading Angle for Docking with Virtual Beacon at 0 degrees

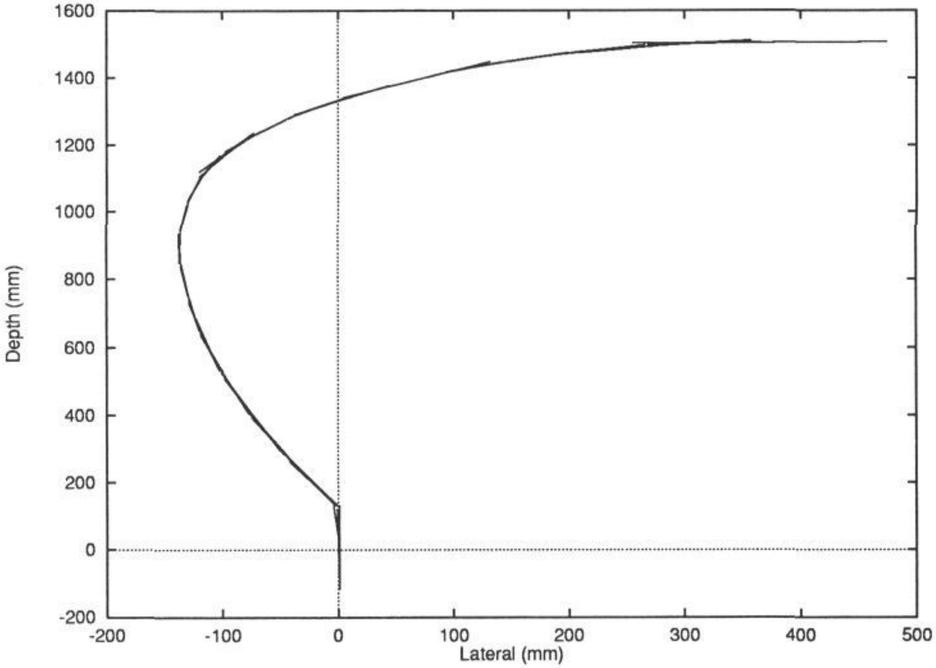

Figure 4: Docking with Virtual Beacon at 90 degrees

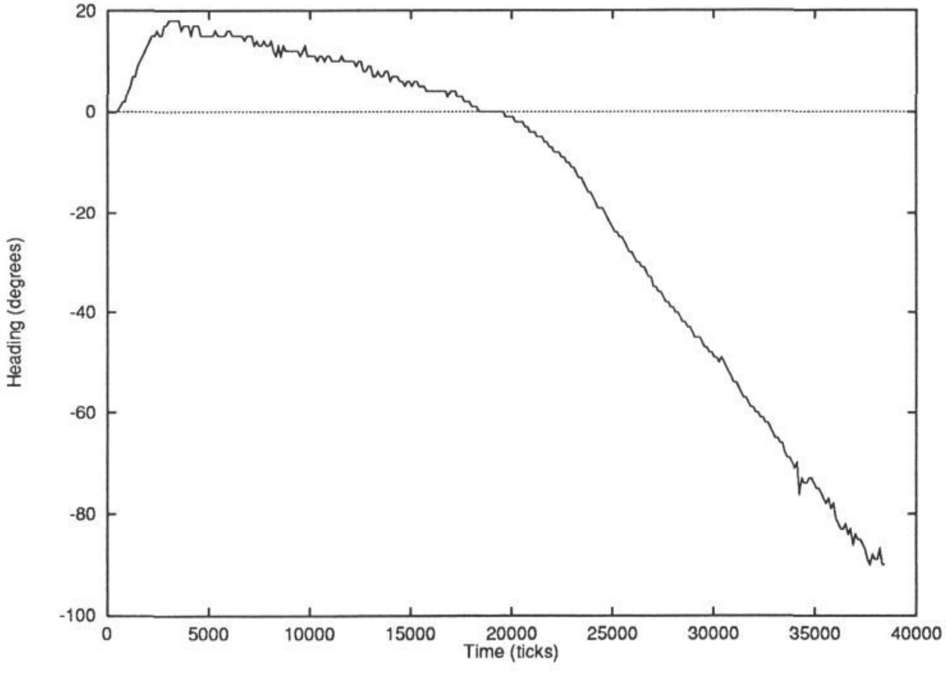

Figure 5: Heading Angle for Docking with Virtual Beacon at 90 degrees

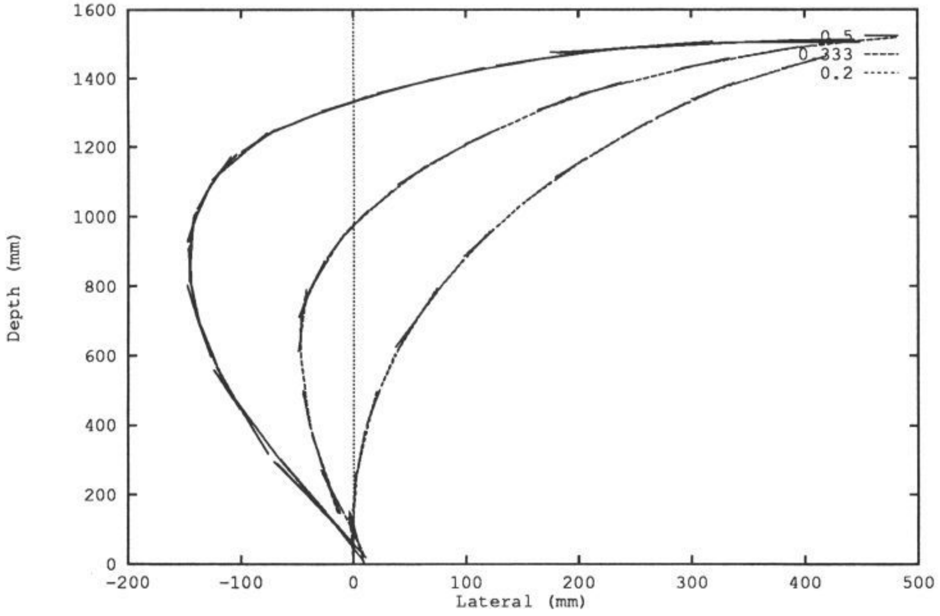

Figure 6: Docking at 90 degrees with multiple orientation gains

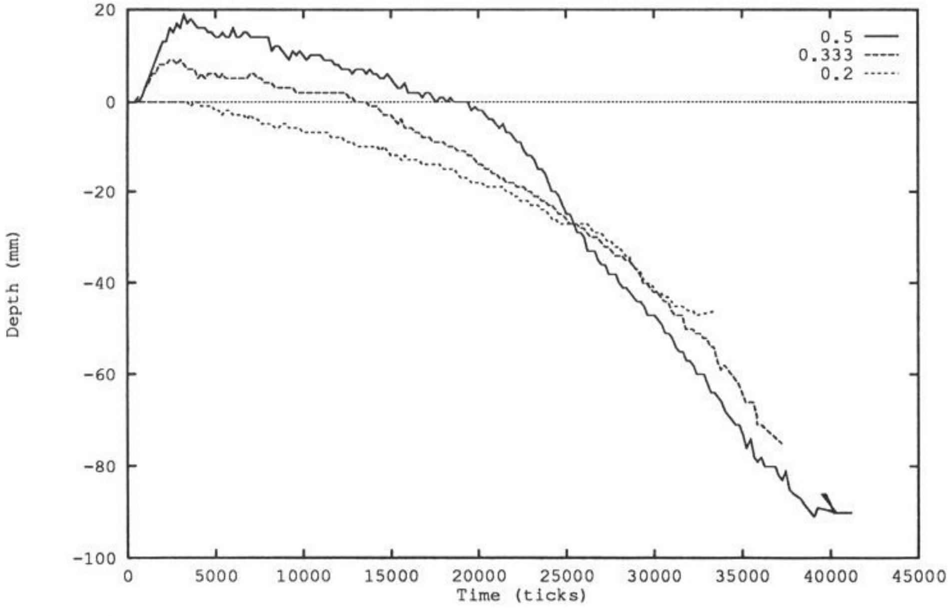

Figure 7: Heading Angle for Docking at 90 degrees with multiple orientation gains

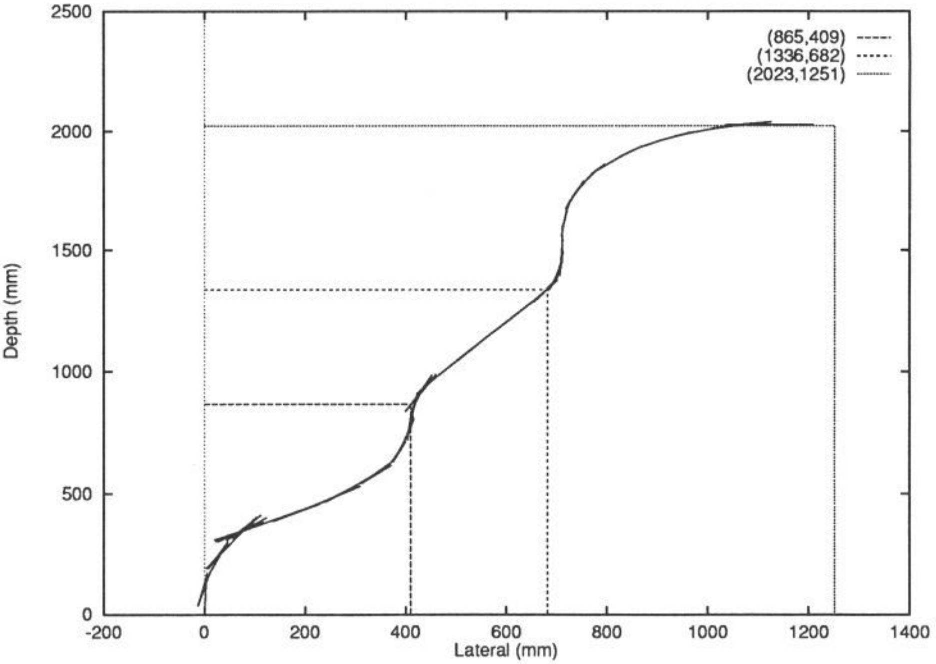

Figure 8: Docking with Multiple Via-Points

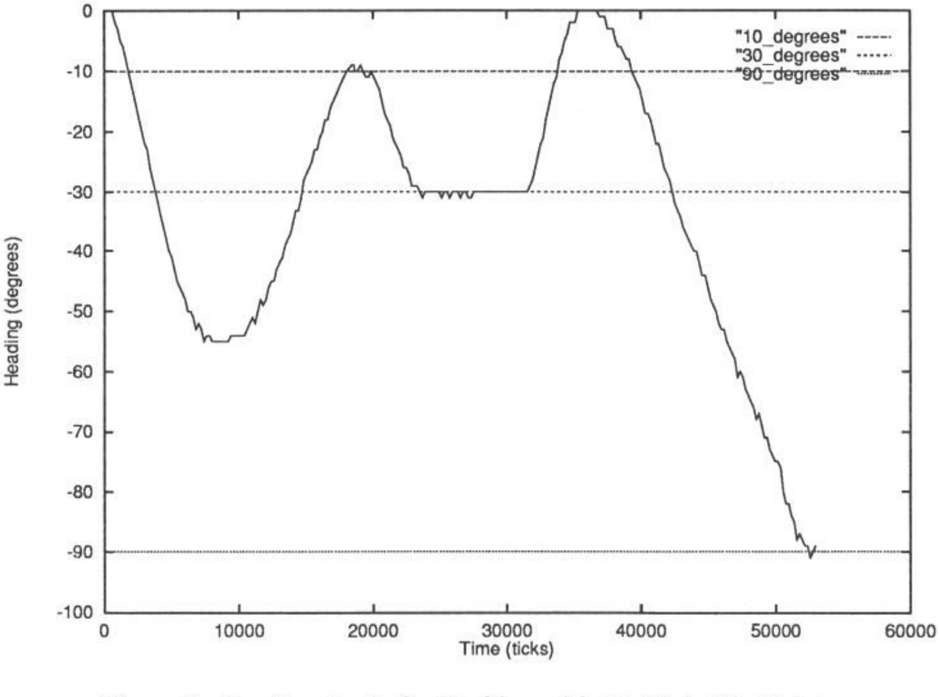

Figure 9: Heading Angle for Docking with Multiple Via-Points

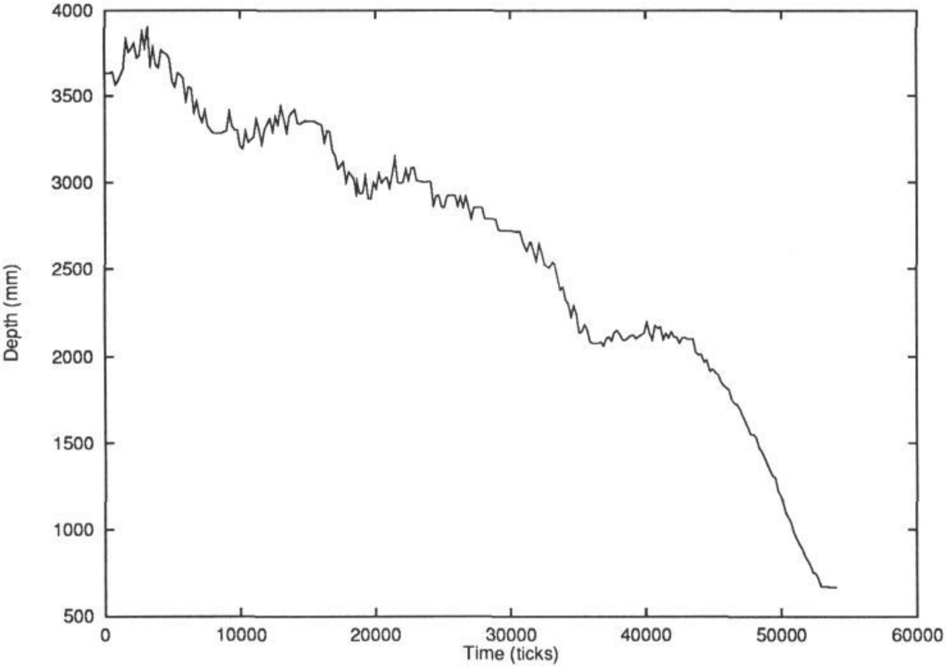

Figure 10: Vision Data for Depth

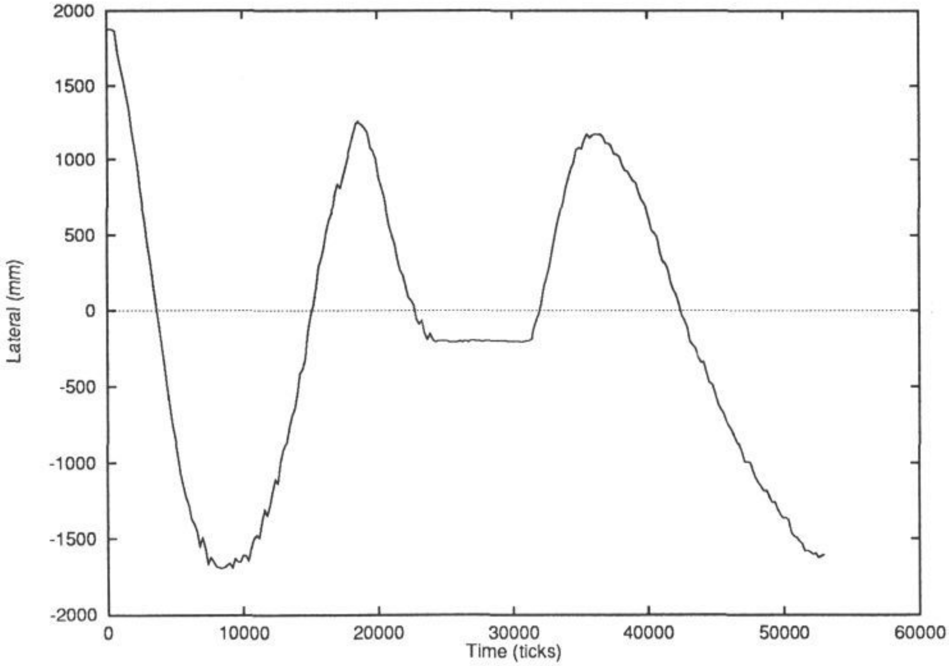

Figure 11: Vision Data for Lateral Distance