Appendices

A Experimental results

We present an evaluation of the performance of the proposed visual control methods, both in simulation and in experiments with real images.

A.1 Simulations

We will first show the simulation results for the omnidirectional visual homing technique. The reference views used in this method were positioned forming a square grid in the simulations, although any arbitrary distribution guaranteeing sufficient geometric diversity on the plane could be chosen. A randomly distributed cloud of 200 points in 3D was generated and projected in each camera.

Three sample homing trajectories with a 16-view reference set and the evolutions of their corresponding motion commands are displayed in Fig. 1. A maximum threshold was set in order to limit the variation of the sizes of the individual sectors S_i between two consecutive steps; this avoids abrupt changes in the linear velocity that may occur when the robot moves right across one of the reference positions.

We also added Gaussian noise to the angles of the projected points to evaluate the performance of the homing method. Figure 2 displays the final position error obtained after adding variable noise in simulations with sets of 4 (the minimum number for our method) and 16 reference images. Increasing the number of reference views makes the system more robust to noise, since the control operates averaging the contributions of the individual views.

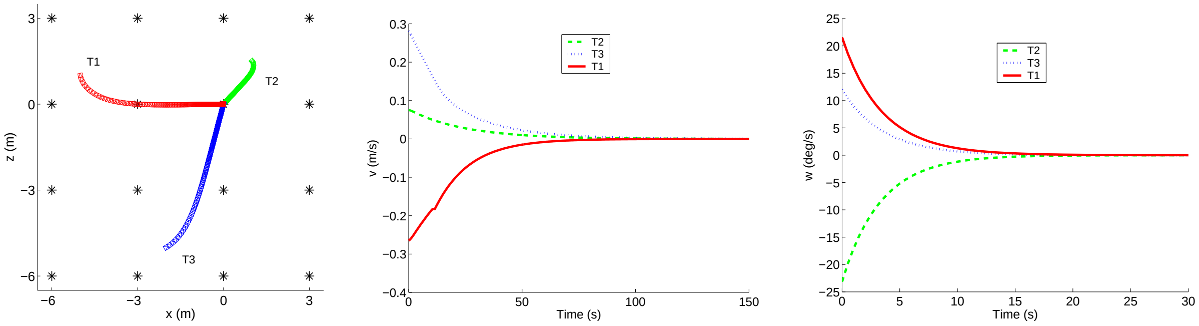

Figure 1: Robot path (left), linear velocity (center) and angular velocity (right) of three sample simulated homing trajectories.

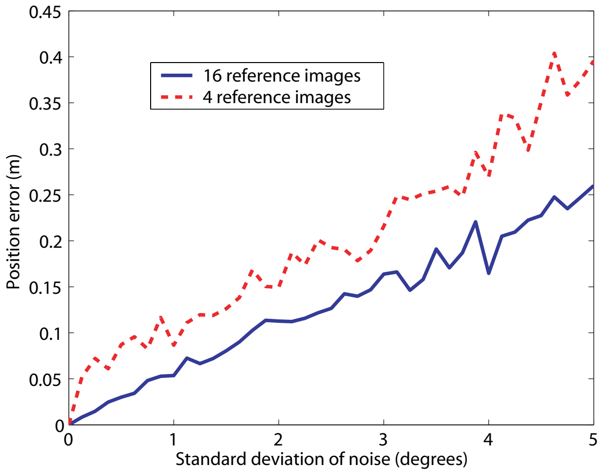

Figure 2: Final position error vs. Gaussian noise for the homing method.

Next, we present some simulation results for the visual control method based on sinusoidal inputs. From the points projected and matched between three cameras, the trifocal tensor was computed and the relative angles between the views were estimated. The state variables of the system were subsequently obtained from this information and used in the closed-loop control. Figure 3 displays three sample trajectories with our method, along with the velocities used. The maximum orientation (ϕ_{max}) was set to 60^o. Smooth trajectories are generated to align the robot with the target, while in the second stage of the control the depth is corrected following a straight-line path.

A simulation with added Gaussian noise is illustrated in Fig. 4. The standard deviation of the noise introduced in the angles projected in the cameras was 1^o. We found it was useful to average the measurements of the system variables over short intervals in order to reduce noise. This is particularly advisable when computing ρ, since it is a variable we are obtaining indirectly. The robot velocities can also be smoothed out by limiting their instantaneous variation. There exists a noise amplification effect near the end of the motion period due to the low values of the denominator of the closed-loop expressions used to compute the amplitude of the velocities (a and b). This can be compensated by setting a minimum threshold on these values. The robot trajectories are still fairly smooth and the position error remains low, as illustrated by the example shown in the aforementioned figure.

Simulations with motion drift are illustrated in Fig. 5. The added drift is proportional to the linear and angular velocities of the robot. It can be seen that the closed-loop control is capable of compensating the drift through the variation of the amplitudes of the sinusoids, and the system reaches the desired state at $t = T/2$. In order to illustrate this effect, only the sinusoidal part of the control (i.e. the first step) is shown.

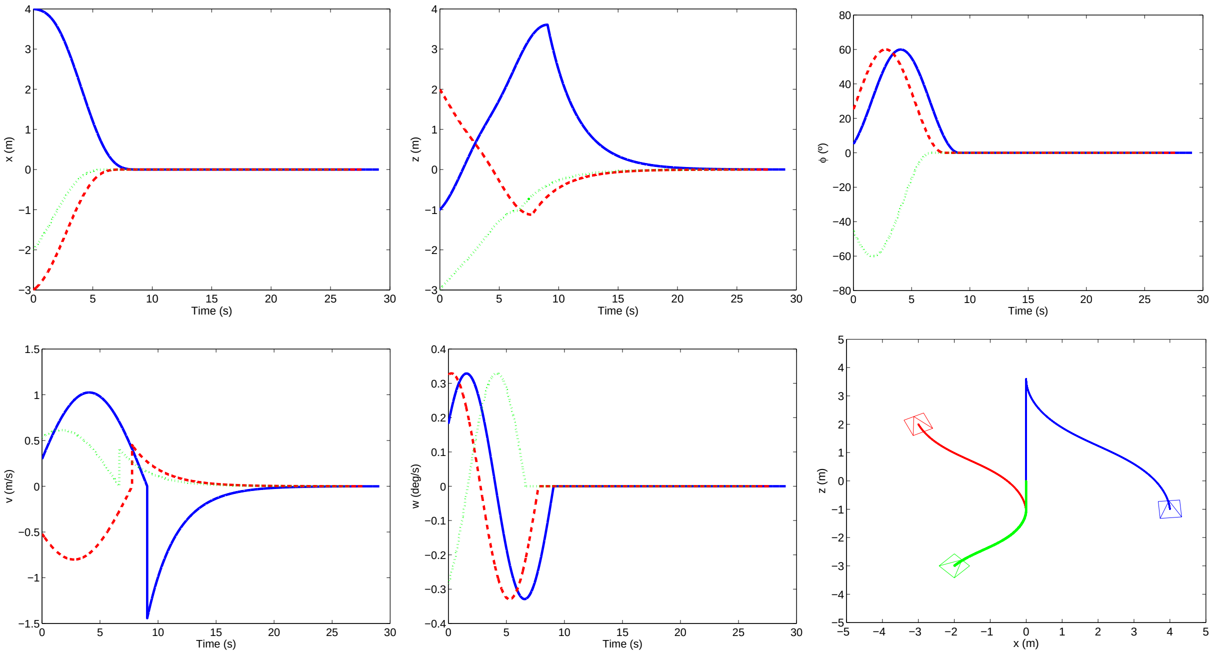

Figure 3: Three sample robot trajectories for the sinusoidal input-based control, from starting locations $(4, -1, 5^{o})$, $(-3, 2, 25^{o})$ and $(-2, -3, -45^{o})$. The evolutions of the state variables x (left), z (center) and ϕ (right) are displayed in the top row. The bottom row shows the linear velocity (left), angular velocity (center) and the robot paths (right) for each trajectory.

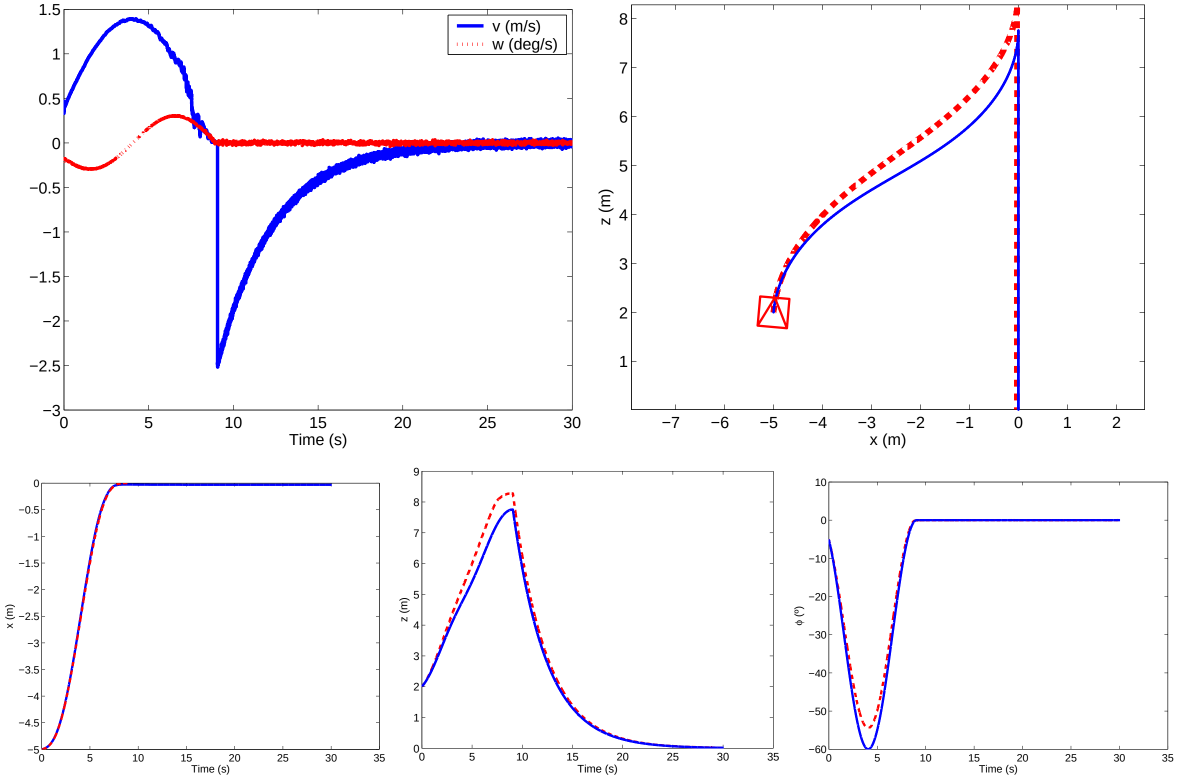

Figure 4: Simulation of the sinusoidal-based control with added Gaussian noise ($\sigma = 1^o$). The robot velocities are displayed in the top left plot. The top right plot shows the robot paths from starting location $(-5, 2, -5^o)$ with noise (dashed line) and without noise (solid line). The bottom row displays the evolutions of variables x (left), z (center) and ϕ (right). Dashed lines correspond to the simulation with added noise, solid lines to the noiseless case.

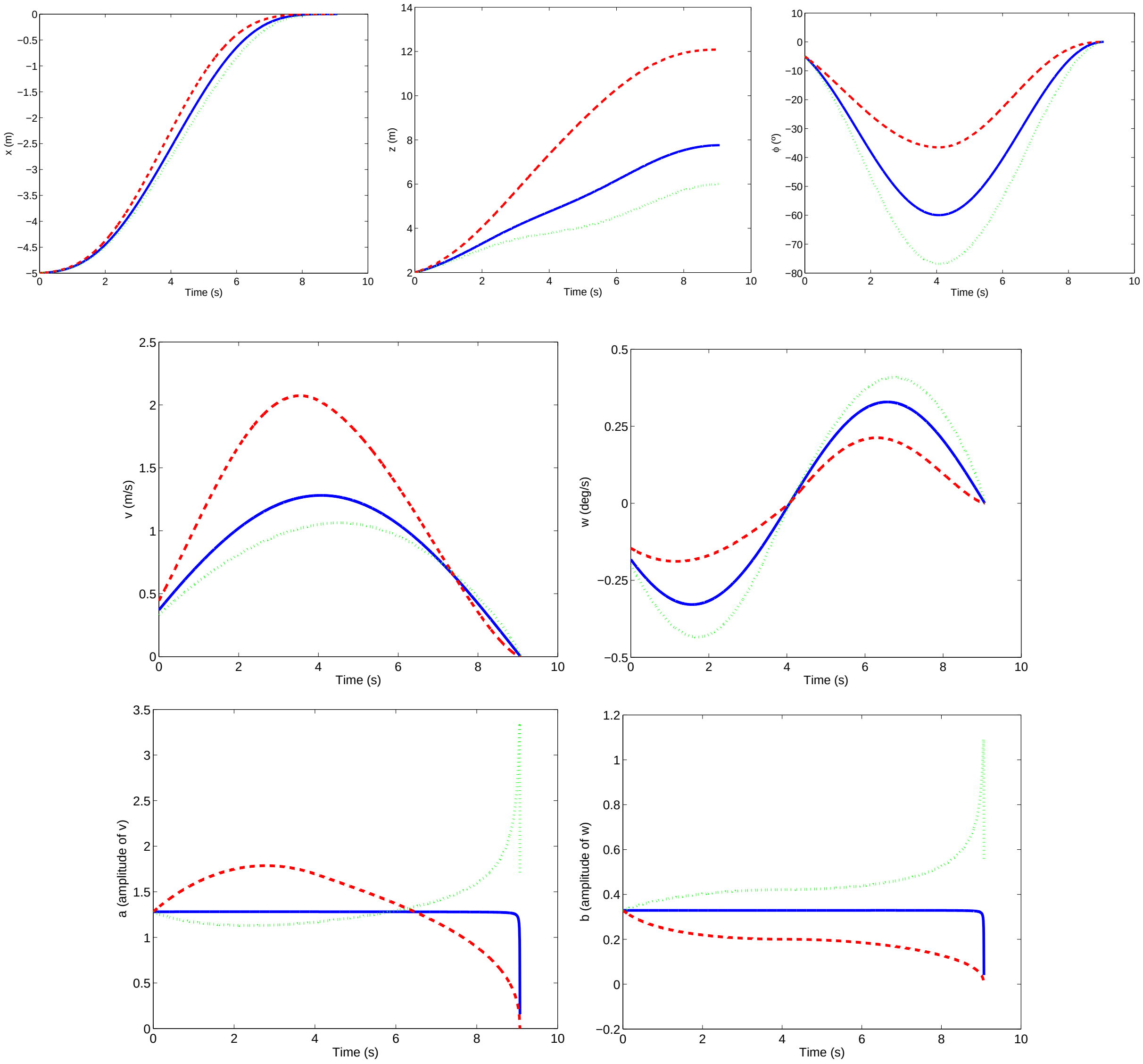

Figure 5: Simulation results with motion drift in the sinusoidal input-based part of the control. A driftless simulation with initial location (-5,2,-5^{o}) is shown in solid line. A simulation with the same parameters and a +20% drift added to both the linear and angular velocities of the robot is displayed with a dashed line. The dotted line shows the results obtained with a drift of -10%.

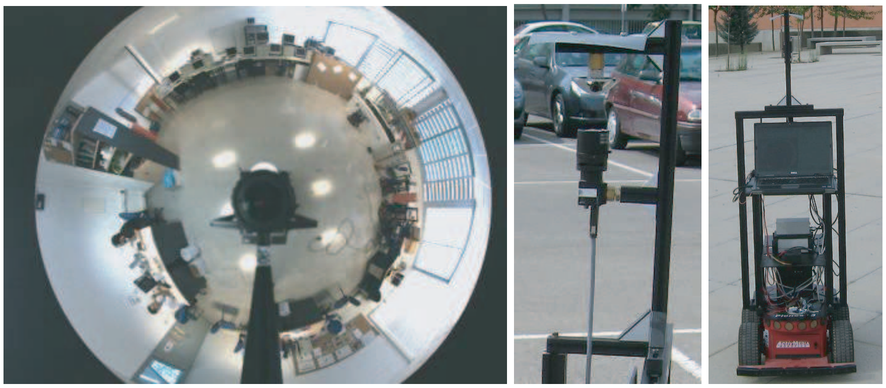

Figure 6: Example image (left), omnidirectional camera (center) and complete setup (right) used for the experiments.

A.2 Experiments with real images

The performance of the omnidirectional visual homing method was tested with real images. The setup for the real experiments consisted of an ActivMedia Pioneer nonholonomic unicycle robot base with a catadioptric vision system, made up of a Point Grey FL2-08S2C camera and a Neovision HS3 hyperbolic mirror, mounted on top. The resolution of the employed images, obtained in an indoor, laboratory setting, was 800 × 600 pixels. No calibration is used, other than assuming that the camera and mirror axis are vertically aligned. Fig. 6 illustrates the experimental setup.

The reference set of views consisted of 20 images acquired from locations forming a 5 × 4 rectangular grid with a spacing of 1.2 m., thus covering a total area of 4.8 × 3.6 m^2. Image features were extracted and matched, and a RANSAC estimation was used to compute the 1D trifocal tensors between the views. The number of three-view correspondences employed lied in the range of 30 (the threshold below which the results started to become unreliable) to 70. Although images taken on opposite sides of the room could not be matched, the connections between adjacent or close sets of views were sufficient to recover the relative angles of the complete reference set. Fig. 7 shows vector field representations for two different goal locations within the grid. The arrows at each location represent the displacement vectors associated with the motion that a vertically oriented robot with nonholonomic constraints would perform from that spot, according to the proposed control law. They all have been scaled by an equal factor. As can be seen, the magnitude of the vectors becomes larger as the distance to the target increases. The line segments show the estimated directions of the epipoles of the goal position in each of the reference locations. The results show good accuracy despite the presence of outliers in the putative matches.

A sequence of 170 images was captured by the robot while moving at constant speed along a straight-line, 5 m. long diagonal path crossing the grid from one of its outer sides to reach a goal position near the opposite side. The linear velocity commands that the homing method would generate at every step in the sequence and the estimated current-to-goal angle (to which the angular velocity of the control law would be proportional) are displayed in Fig. 8. The results of these preliminary experiments show that the homing method can be successful in an environment with sufficiently large sets of feature matches.

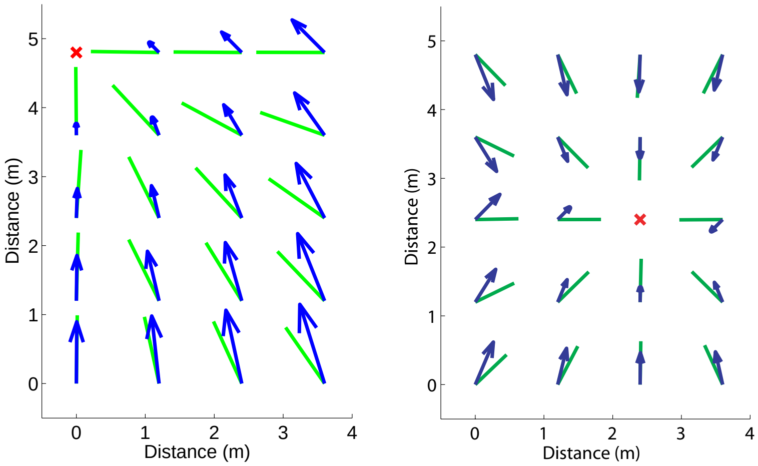

Figure 7: Displacement vectors (arrows) and directions of the epipoles (line segments) with respect to the goal estimated at every reference position for two different goal locations (marked with a cross) in real setting.

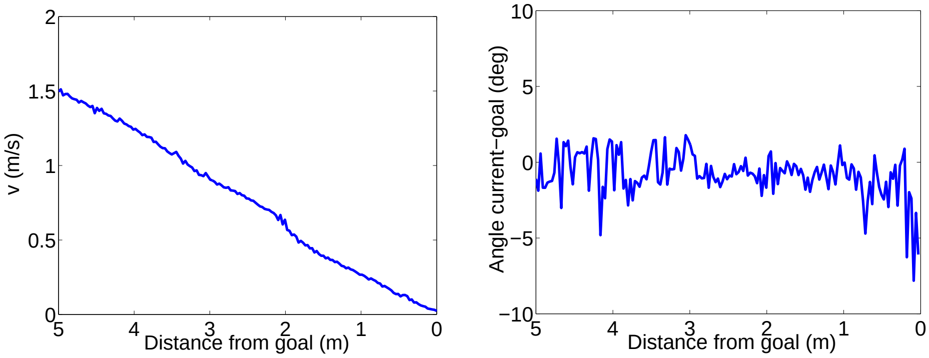

Figure 8: Linear velocity (left) and angle to the goal (right) estimated in real image sequence.