

DYNAMIC 3D PATH FOLLOWING FOR AN AUTONOMOUS HELICOPTER

Gianpaolo Conte,¹ Simone Duranti,¹
Torsten Merz¹

*Department of Computer and Information Science,
Linköping University, SE-58183 Linköping, Sweden*

Abstract: A hybrid control system for dynamic path following for an autonomous helicopter is described. The hierarchically structured system combines continuous control law execution with event-driven state machines. Trajectories are defined by a sequence of 3D path segments and velocity profiles, where each path segment is described as a parametric curve. The method can be used in combination with a path planner for flying collision-free in a known environment. Experimental flight test results are shown.

Keywords: aerospace engineering, architectures, autonomous vehicles, finite state machines, helicopter control, splines, trajectories

1. INTRODUCTION

This work is part of the WITAS Unmanned Aerial Vehicle (UAV) Project (Doherty, 2004), a long-term basic research project with the goal of developing information technology systems for UAVs and core functionalities necessary for the execution of complex missions. The main objective is the development of an integrated hardware/software UAV for fully autonomous missions in an urban environment. A research prototype has been developed using a Yamaha RMAX helicopter as a flying platform. A number of interesting missions have been successfully demonstrated in a small uninhabited urban area in the south of Sweden called Revinge, which is used as an emergency services training area.

In order to navigate in an area cluttered by obstacles, such as an urban environment, path planning, path following (PF) and path switching mechanisms are needed. Several methods have

been proposed to solve this type of navigation problem (Egerstedt *et al.*, 1999; Frazzoli, 2001).

Our major achievements in this paper are the development and flight-testing of an algorithm to follow a 3D path with a given velocity profile and a switching mechanism to enable the integration with a path planner in the deliberative part of the UAV architecture (Pettersson and Doherty, 2004). The method developed for PF is weakly model dependent and computationally efficient. The strategy used to follow a desired path is a velocity control tangent to the path and a position control orthogonal to it: the helicopter has to fly close to the geometric path with a specified forward speed. This approach is better known as dynamic PF. In the trajectory tracking problem the system is designed to follow a trajectory in the state-space domain where the state is parameterized in time: the path is not prioritized. In PF methods the path is always prioritized and this is a requirement for robots that for example have to follow roads and avoid collisions with buildings. A theoretical approach to dynamical PF is given in (Sarkar *et al.*, 1994). The path we want to follow

¹ Supported by the Wallenberg Foundation, Sweden

is a three-dimensional parameterized space curve. The motion of the reference point on the curve is governed by a differential equation containing error feedback. Similar methods have also been investigated in (Egerstedt *et al.*, 2001).

In (Harbick *et al.*, 2001) a technique for following planar spline trajectories using a behavior-based control architecture is implemented and tested in flight. The method developed in this paper differs from (Harbick *et al.*, 2001) in that it allows dynamic modification of the trajectory during execution and provides a mechanism that coordinates and monitors the processes to achieve proper control. Furthermore, our implementation allows 3D path tracking and information about the curvature of the path is fed forward in the control loops for enhanced tracking accuracy during manoeuvred flight at higher speeds.

2. SYSTEM OVERVIEW

The WITAS UAV system consists of a slightly modified Yamaha RMAX helicopter and the WITAS on-board system (Fig. 1). In this paper we focus on system components which are relevant for dynamic 3D path following. Our aerial robot has many more skills. A description of the full hard- and software system can be found in (Doherty *et al.*, 2004; Merz, 2004).



Fig. 1. The WITAS helicopter

The helicopter has a total length of 3.6 m (incl. main rotor), a max. take-off weight of 95 kg, and is powered by a 21 hp two-stroke engine. Yamaha equipped the remote-controlled RMAX with an attitude sensor (YAS) and an attitude control system (YACS). For the experiments described here the following components of the WITAS on-board system were used: an integrated INS/GPS with DGPS correction, a barometric altitude sensor, a PC104 embedded computer (700 MHz Pentium PIII), and a wireless Ethernet bridge.

The PC104 computer reads all sensors, runs the control software, and sends commands to the YACS. Sensor measurements and control outputs are logged in this computer and sent simultaneously to a ground station for on-line analysis.

Different control modes and task procedures can be selected by a ground operator during flight.

Paths are decomposed into path segments, which are requested by the dynamic path following controller during execution. This method is chosen, as it allows to model almost any space curve and makes path modification easy. If a segment is not available in time, the system switches into a safety mode.

The structure of the hybrid control system for dynamic path following is shown in Fig. 2. Each block represents a *functional unit*. All functional units can be executed concurrently and asynchronously. At the highest level a *task procedure* provides control with path segment data. A task procedure is a computational mechanism that achieves a certain behavior of the WITAS UAV system (Doherty *et al.*, 2004). It is coupled to a state machine which coordinates data transfer, reports errors to the task procedure, and switches control modes. It uses statements derived from sensor measurements as conditions for state transitions. A set-point generator computes a number of set-points from path segment data and sensor measurements and passes it to an outer loop control. The inner loop is the Yamaha Attitude Control System (YACS) that stabilizes the attitude angles, the yaw and the vertical dynamics.

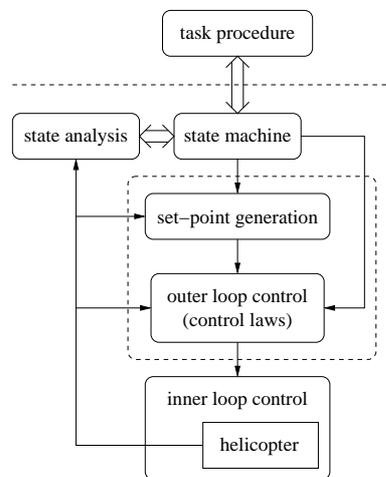


Fig. 2. Structure of the hybrid control system

3. TASK PROCEDURE AND STATE MACHINE

The interaction between task procedures and low level control is handled by an event-driven state machine (hybrid control). In the system considered here, a hierarchical concurrent state machine is implemented (HCSM). It is represented as a set of state transition diagrams similar to Harel's statecharts (Harel, 1987).

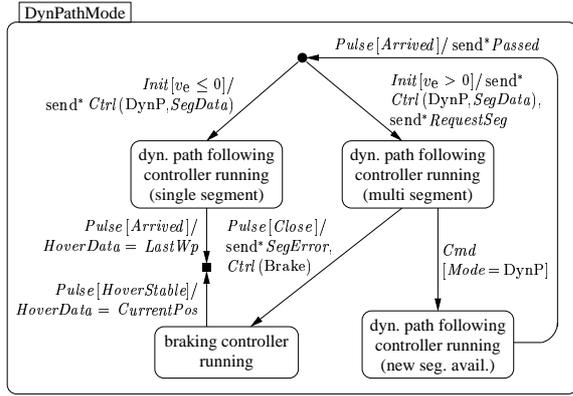


Fig. 3. State machine for dynamic path following

In the following, the state machine for dynamic path following is explained. The state transition diagram² is shown in Fig. 3. For a path with one segment (end velocity $v_e \leq 0$), the segment parameters $SegData$ are passed to the set-point generator and the dynamic path following controller is started. A segment is defined by start and end points, start and end directions, target velocity, and end velocity. In the case of several segments (end velocity $v_e > 0$), the state machine passes the segment parameters, starts the same controller and sends a *RequestSeg* event to the task procedure. The next segment has to be provided before the helicopter reaches a point from where it is impossible to stop at the end point of the current segment (*Close* becomes true). This is a safety mechanism which prevents the helicopter from leaving the current path in case no new segment is available. In this case, or if the helicopter is not able to slow down to the desired velocity at the end point, a *SegError* event is sent and a braking controller is started which will brake the helicopter with maximum deceleration. When the helicopter passes the end point of a path segment (*Arrived* becomes true) a *Passed* event is sent to the task procedure. The state machine exits, when the helicopter hovers.

Fig. 4 shows an example of a state machine for a path with two segments³. The upper part models a task procedure (user state machine) and the lower part a flight mode switching mechanism. Both machines run concurrently. When the autonomous mode is engaged (*AutoSwitch* becomes true) the hovering controller is started. As soon as the helicopter hovers stably, the first segment is flown. The hovering controller is started again when the helicopter arrives at the final waypoint.

² *Pulse* is an event sent periodically, *Init* triggers a transition from an entry state (circular node) when condition holds, *Exit* is sent to the superstate when entering an exit state (square node).

³ Rectangular boxes within state nodes denote nested state machines. Superstate transitions are executed prior to substate transitions.

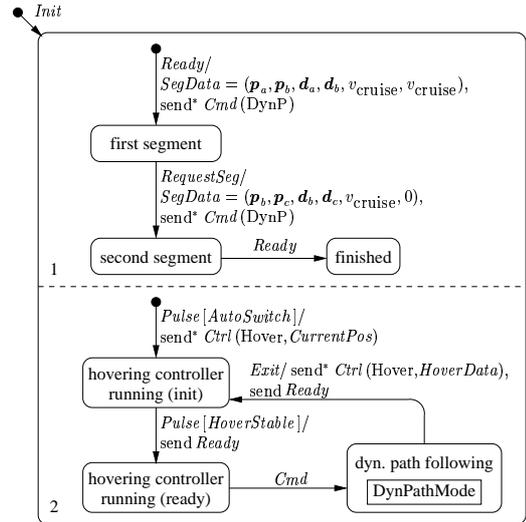


Fig. 4. Example of a state machine for a path with two segments

In the real system, the state machine for mode switching handles more flight modes and is separated from the user state machine (Merz, 2004).

4. SET-POINT GENERATION

The PF algorithm provides the set-points for the outer loop control. The inputs are provided by the event handler and the position sensor (INS/DGPS).

The analytical description of the 3D path is a cubic spline that has second-order continuity (C^2) at the joints, this is a requirement which avoids discontinuity in the helicopter's acceleration. A global reference frame is associated with each segment where the X-axis points north, the Y-axis points east and the Z-axis points down. The analytical form of the curve is:

$$P = As^3 + Bs^2 + Cs + D \quad (1)$$

where A, B, C and D are 3D vectors defined by the boundary conditions and s is the linear coordinate of the curve.

For each value of s the path generator provides the path parameters: position, tangent and curvature. The curvature is used to compute the centripetal acceleration needed to follow the path (feed-forward term in the lateral control law), while the tangent T is used to align the helicopter body to the path. The curvature K is a 3D vector and is calculated in the global frame as follows:

$$K = T \times Q \times T / |T|^4 \quad (2)$$

$$T = 3As^2 + 2Bs + C \quad (3)$$

$$Q = 6As + 2B \quad (4)$$

where Q is the second order derivative.

The reference point on the nominal path is found by satisfying the geometric condition that the scalar product between the tangent vector and the error vector has to be zero:

$$\mathbf{E} \bullet \mathbf{T} = 0 \quad (5)$$

where the error vector \mathbf{E} is the helicopter distance from the candidate control point. The control point error feedback is then calculated as follows:

$$e_f = \mathbf{E} \bullet \mathbf{T} / |\mathbf{T}| \quad (6)$$

that is the magnitude of the error vector projected on the tangent \mathbf{T} . The control point is updated using the differential relation:

$$d\mathbf{P} = \mathbf{P}' \cdot ds \quad (7)$$

Equation 7 is applied in the discretized form:

$$s(n) = s(n-1) + \frac{e_f}{\left| \frac{d\mathbf{P}}{ds}(n-1) \right|} \quad (8)$$

where $s(n)$ is the new value of the parameter. Once the new value of s is known, all the path parameters can be calculated.

The PF algorithm receives as inputs the target velocity v_t and the final velocity v_e that the helicopter must have at the end of the segment. The path planner assigns the target velocity which is related to the mission specification only. This means that the path planner doesn't have to take into account any dynamic limitation of the helicopter itself. The control law tries to keep the target velocity, but when it is not compatible with the local curvature of the path and the helicopter performance limitations, the algorithm provides an automatic limit on velocity. Velocity limitations can be activated for two reasons: due to the turn bank or the yaw rate limit of the helicopter. In order to make a coordinated turn at constant altitude, the flight mechanics provides the relation between the velocity, the roll angle and the curvature radius of the turn. Mechanical limits exist on the maximum achievable swash plate angles, and furthermore the helicopter envelope has currently been opened up to $\phi_{\max} \pm 8$ deg for the roll angle (relative to the hovering bank angle that is about 4.5 deg), and $\omega_{\max} \pm 26$ deg/sec for the yaw rate. Under the described limitations, it is possible to calculate the maximum speed:

$$V_{max1} = \sqrt{Rg\phi_{max}} \quad (9)$$

$$V_{max2} = \omega_{max}^2 R \quad (10)$$

where R is the local curvature radius and g is the gravity acceleration. The target speed assigned to the path is compared with these two limits and the lower speed is taken as target.

The braking algorithm continually checks the distance between the helicopter and the end point of the path. If the required acceleration to reach the

final target velocity exceeds a given value (currently set to 1 m/s²), the current target velocity is limited in order to maintain a constant deceleration. In order to know the distance between the helicopter and the end of the path, an estimate of the final arc length of the curve has to be calculated. The arc length of the spline between the control point and the end point of the path is:

$$l_{end} = \int_S^{S_{end}} \sqrt{[x'(s)]^2 + [y'(s)]^2 + [z'(s)]^2} ds \quad (11)$$

If an analytical solution of the integral cannot be found, a numerical method is used (rectangular integration with for example 20 integration steps).

To gain computational time, the increments of the flown path l_n are subtracted from l_{tot} (total length of the path, i.e. l_{end} at first iteration) to get a good estimate of l_{end} at each control cycle:

$$l_n = \sqrt{\begin{matrix} [x_n - x_{n-1}]^2 + \\ [y_n - y_{n-1}]^2 + \\ [z_n - z_{n-1}]^2 \end{matrix}} \quad (12)$$

A path segment is considered finished when l_{end} is small enough. It should be emphasized that l_{end} is the arc length between the control point on the curve and the end point of the path and not between the helicopter and the end of the path. This makes the system more robust with regard to position error of the helicopter on the path.

5. CONTROL LAWS

The outer loop control (velocity and position control) provides inputs for the YACS in order to follow the path with the desired velocity. The inner loop deals with the coupling dynamics of the helicopter, so that the outer loop can handle the four degrees of freedom as decoupled (i.e. yaw rate, vertical velocity, pitch and roll angles). The position and velocity error and centripetal acceleration vectors are computed in the global frame and then transformed into control inputs after rotation in the helicopter's body frame. The acceleration vector is used as feed forward input in the control law to improve the tracking in the presence of path curvature. As regards the acceleration vector, only the component in the horizontal plane orthogonal to the path is used. PD and PI compensators are used respectively for position and velocity control. The algorithm described in the previous section makes sure that the position error vector is orthogonal to the path and the velocity error vector is tangent to the path. Given that the two error vectors are orthogonal the velocity control doesn't interfere with the position control. The control equations for the four channels are the following:

$$\begin{aligned}
\theta_C &= K_{px}\delta X + K_{dx}\delta\dot{X} + K_{pvx}\delta V_X + \\
&\quad + K_{ivx}\delta V_{Xsum} + K_{fx}A_X \\
\Delta\phi_C &= K_{py}\delta Y + K_{dy}\delta\dot{Y} + K_{pvy}\delta V_Y + \\
&\quad + K_{ivy}\delta V_{Ysum} + K_{fy}A_Y \\
V_{ZC} &= K_{pz}\delta Z + K_{dz}\delta\dot{Z} + K_{pvz}\delta V_Z + \\
&\quad + K_{ivz}\delta V_{Zsum} + K_{fz}A_Z \\
\omega_C &= K_{pw}\delta\psi
\end{aligned} \tag{13}$$

where the subscripted K 's are control gains, the δ 's are control errors, the pedices *sum* indicate the integral terms and the A 's the components of the centripetal acceleration vector. θ_C is the target pitch angle, $\Delta\phi_C$ is the desired roll angle relative to the hovering roll angle, ω_C is the target yaw rate and V_{ZC} is the target vertical velocity.

6. EXPERIMENTAL RESULTS

The PF mode has been tested first in simulation and then in flight. The flight dynamics mathematical model of the augmented RMAX has been developed within the WITAS project and implemented in C. Simulations are done using hardware in the loop.

Only results from the flights are reported in the following.

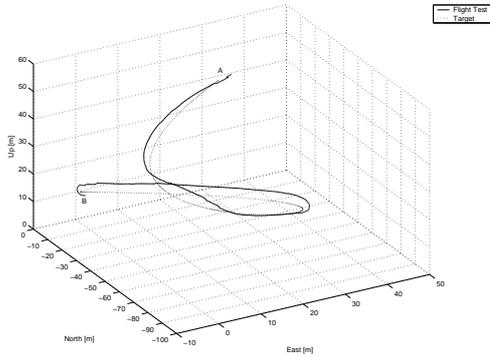


Fig. 5. Target and actual 3D helicopter path

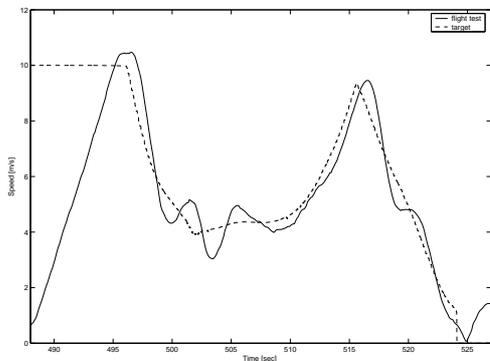


Fig. 6. Target and actual speed of the helicopter

Fig. 5 and 6 show a 3D segment and the velocity profile during one of the flight-tests. The helicopter hovers at point A at 40 meters altitude,

starts the descending spiral, brakes and hovers at point B at 10 meters altitude. The maximum speed for the flight was set to 10 m/s, and the controller limited the target speed according to the local curvature and the braking algorithm. The maximum vertical speed component was around 3 m/s.

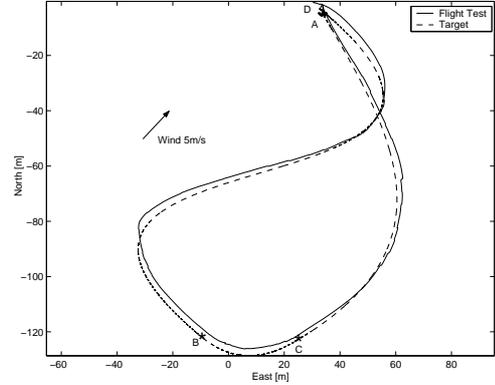


Fig. 7. Multisegment 2D path

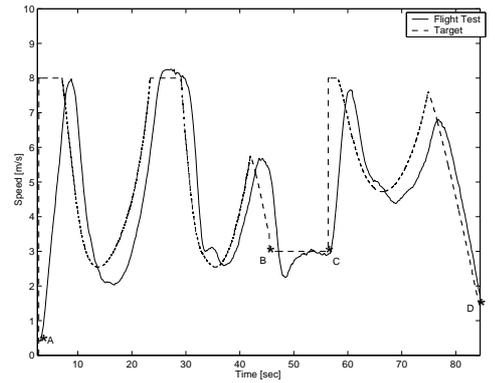


Fig. 8. Speed profile of a multisegment path

Fig. 7 and 8 show a trajectory consisting of 3 path segments at constant altitude. The mission starts with autonomous hovering in point A, then the helicopter flies the first path segment with maximum speed of 8 m/s; at point B the first segment is finished and a path switching leads the helicopter to the second segment with a maximum speed of 3 m/s; in point C the switch to the third path segment with maximum speed of 8 m/s takes place. Finally the helicopter brakes and hovers in point D where the mission ends. The wind was blowing constantly at 5 m/s. The tracking error depends on the angle between the path and the wind direction. In this case the maximum error is about 3 meters.

Table 1 shows the results of several paths flown with different wind conditions and different velocities. The table reports three flight sessions (separated by horizontal lines) flown on three different days so as to cover three different wind conditions. In order to give more generality to the results,

| Path | Av Err | Max Err | St Dev | Speed | Wind |
|------|--------|---------|--------|-------|-------------|
| [-] | [m] | [m] | [m] | [m/s] | [m/s] |
| HR | 1.2 | 3.4 | 0.7 | 10 | 4 |
| HL | 1.9 | 4.1 | 1.3 | 10 | 4 |
| DR | 1.5 | 2.8 | 0.7 | 10 | 4 |
| DL | 1.8 | 3.5 | 1.1 | 10 | 4 |
| CR | 1.7 | 3.3 | 0.7 | 10 | 4 |
| CL | 1.9 | 4.1 | 1.3 | 10 | 4 |
| HR | 1.1 | 2.7 | 0.8 | 10 | 2 |
| HL | 0.8 | 2.2 | 0.6 | 10 | 2 |
| DL | 0.9 | 1.8 | 0.5 | 10 | 2 |
| SLN | 0.3 | 0.8 | 0.2 | 3 | ≈ 0 |
| SLN | 0.5 | 1.4 | 0.3 | 3 | ≈ 0 |
| SLN | 0.5 | 1.9 | 0.5 | 3 | ≈ 0 |
| SLN | 0.6 | 1.4 | 0.3 | 3 | ≈ 0 |
| SLN | 0.4 | 1.3 | 0.3 | 3 | ≈ 0 |

HR = Horizontal Right HL = Horizontal Left
DR = Descending Right DL = Descending Left
CR = Climbing Right CL = Climbing Left
SLN = Straight Line

Table 1. Experimental data

representative paths of typical flight manoeuvres have been chosen. In the HR path the helicopter describes a complete turn in the horizontal plane turning right, in the DR path the helicopter makes the same turn while it is descending from 40 to 10 meters and in the CR path the helicopter turns while climbing from 10 to 40 meters. The same flights are repeated turning left instead. Fig. 5 for example is a DL path.

The first column of the table shows the kind of path flown, the second, third and fourth column are the average error, maximum error and standard deviation error, and the fifth and sixth column are the maximum ground speed reached and the average wind speed. The error is the distance of the helicopter to the reference path and is calculated using the INS/GPS signal, which is also used as control signal during flight (an independent source would have been a better reference for the purpose of this statistics). Because of the occurrence of sudden jumps of the INS/GPS position signal, the maximum errors shown in the table are not always imputable to control errors; to evaluate the performance of the PF, the average error gives more reliable information.

To summarize the results of the table, the first session gives the worst results because of the wind, moreover the right turn gave better results than the left one because the wind was blowing from the side. In the second session the overall performance increases because of less wind. In the third session several straight lines of 170 meters at low speed were flown, during the test the wind was negligible.

7. CONCLUSIONS AND FUTURE WORK

The results of the experimentation show a satisfactory tracking behavior. The position control

error is well within the accuracy of the available position measurement. The algorithm has also been successfully tested in relatively severe weather conditions, with wind levels up to 15 m/s. In the presence of the strongest wind levels, the algorithm could be improved in order to reduce the lateral error; an integrative compensator could be added for this purpose but the tuning would not be straight forward in presence of high curvature. The PF mode is now implemented in the software architecture of the WITAS helicopter and is being used as a core functionality in complex mission tasks.

REFERENCES

- Doherty *et al.* (2004). A distributed architecture for autonomous unmanned aerial vehicle experimentation. In: *Proc. of the 7th International Symposium on Distributed Autonomous Robotic Systems*. pp. 221–230.
- Doherty, P. (2004). Advanced research with autonomous unmanned aerial vehicles. In: *Proc. of the 9th International Conference on the Principles of Knowledge Representation and Reasoning*. pp. 731–732.
- Egerstedt, M., T. J. Koo, F. Hoffmann and S. Sastry (1999). Path planning and flight controller scheduling for an autonomous helicopter. *Lecture Notes in Computer Science* **1569**, 91–102.
- Egerstedt, M., X. Hu and A. Stotsky (2001). Control of mobile platforms using a virtual vehicle approach. *IEEE Transactions on Automatic Control* **46**(11), 1777–1782.
- Frazzoli, E. (2001). Robust Hybrid Control for Autonomous Vehicle Motion Planning. PhD thesis. Massachusetts Institute of Technology.
- Harbick, K., J. Montgomery and G. Sukhatme (2001). Planar spline trajectory following for an autonomous helicopter. In: *Proc. of the 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation*. pp. 408–413.
- Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming* **8**(3), 231–274.
- Merz, T. (2004). Building a system for autonomous aerial robotics research. In: *Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicles*.
- Petterson, P-O and P. Doherty (2004). Probabilistic roadmap based path planning for an autonomous unmanned aerial vehicle. In: *ICAPS-04 Workshop on Connecting Planning Theory with Practice*.
- Sarkar, N., X. Yun and V. Kumar (1994). Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots. *International Journal of Robotics Research (MIT Press)* **13**(1), 55–69.