Landmark navigation and autonomous landing approach with obstacle detection for aircraft

Simon Fürst[a], Stefan Werner[a], Dirk Dickmanns[b], Ernst-Dieter Dickmanns[a]

[a]Institut für Systemdynamik und Flugmechanik, Universität der Bundeswehr München
[b]Institut für Technik intelligenter Systeme
Werner-Heisenberg-Weg 39, D-85577 Neubiberg

ABSTRACT

A machine perception system for aircraft and helicopters using multiple sensor data for state estimation is presented. By combining conventional aircraft sensors like gyros, accelerometers, artificial horizon, aerodynamic measuring devices and GPS with vision data taken by conventional CCD-cameras mounted on a pan and tilt platform, the position of the craft can be determined as well as the relative position to runways and natural landmarks.
The vision data of natural landmarks are used to improve position estimates during autonomous missions. A built-in landmark management module decides which landmark should be focused on by the vision system, depending on the distance to the landmark and the aspect conditions. More complex landmarks like runways are modeled with different levels of detail that are activated dependent on range. A supervisor process compares vision data and GPS data to detect mis-tracking of the vision system e.g. due to poor visibility and tries to reinitialize the vision system or to set focus on another landmark available. During landing approach obstacles like trucks and airplanes can be detected on the runway.
The system has been tested in real-time within a hardware-in-the-loop simulation. Simulated aircraft measurements corrupted by noise and other characteristic sensor errors have been fed into the machine perception system; the image processing module for relative state estimation was driven by computer generated imagery. Results from real-time simulation runs are given.

Keywords: machine perception, computer vision, autonomous aircraft, hardware-in-the-loop simulation, landmark navigation, obstacle detection.

1. INTRODUCTION

The extended machine perception system (MPS) presented in this paper has been designed to perform autonomous and automatic aircraft or rotorcraft flights [1]. A complete mission from take-off to landing has to be conducted. For this purpose, the system has to have the capabilities to estimate the actual state of the aircraft, to navigate according to the mission waypoints, and to stabilize and guide the aircraft. For full autonomy only sensors mounted onboard the aircraft are being used, i.e. no precision navigation aids with ground-based infrastructure are included.
Optimal state estimation is achieved by combining standard aircraft sensors which provide good information in the high frequency range with an image processing system (IPS), which provides good information at lower frequencies and a very high accuracy during the final landing approach.
The runway obstacle detection and tracking (RODT) module is part of the IPS and can detect obstacles like trucks on the runway. When an obstacle gets detected an alert is sent to the MPS in order to break off the landing approach.
Section 2 describes the process structure of the MPS and the IPS to conduct autonomous and automatic helicopter flight from lift-off along a pre-defined flight path towards a landing spot.
Section 3 explains the hardware implementation designed for hardware-in-the-loop simulation and real flight tests.
Section 4 focuses on results from real-time simulation trials. A horizontal flight loop and a landing approach with obstacle detection will be discussed.
Section 5 gives some conclusions and an outlook on further developments.

2. PROCESS AND DATA STRUCTURE OF THE MPS AND THE IPS

The machine perception system presented here consists of the following basic processes:

Further author information:
S.F.: Email: simon.fuerst@unibw-muenchen.de; WWW: http://www.unibw.muenchen.de/campus/LRT/LRT13; Telephone: +49-89-6004-3580; Fax: +49-89-6004-2082

Paper presented at the AGARD MSP Symposium on "System Design Considerations for Unmanned Tactical Aircraft (UTA)", held in Athens, Greece, 7-9 October 1997, and published in CP-594.

- the multi-sensor state estimator (MSSE)
- the vehicle control process (VC)
- the aircraft interface process (AC-IF)
- the interface process to the image processing system (IPS-IF).

The image processing system consists of the following basic processes:

- the model based landmark tracking process (REALIS)
- the runway obstacle detection and tracking process (RODT)
- the TIP-Bus processes (TIP)
- the interface process to the machine perception system (MPS-IF)

The processes and their communication data paths are shown in Fig. 2.1.
The processes and the data structure of the MPS had already been extensively discussed in [1]. So in this paper only a brief review of the MPS will be given, but the IPS will be discussed in detail.

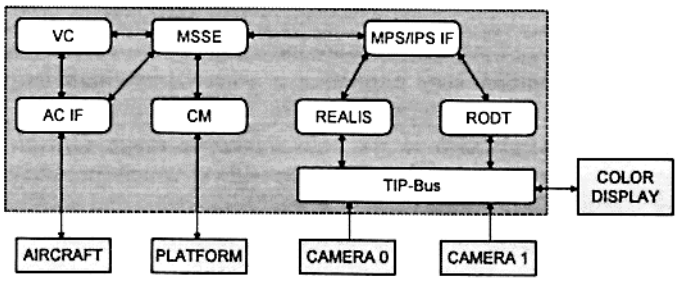


Figure 2.1: Processes and data paths of the MPS and IPS.

2.1 Multi-sensor state estimator

The multi-sensor state estimator uses measurements from several sensor processing modules to generate the optimal state estimate. The state vector of the MPS consists of the state variables shown in equation 2.1; they are related to a body-fixed coordinate system located in the craft's center of gravity and pointing along the main axes of moments of inertia, and a geographic coordinate system.
An Extended Kalman Filter [8] is implemented to perform recursive state estimation. State prediction is done by exploitation of either a non-linear dynamical model of a BO105 helicopter or an integration model.
All sensor processing modules attached to the multi-sensor state estimator receive a predicted state and the related error covariance matrix; they return valid measurements in the given structure. As a result, the state estimate update does not need to have any knowledge about the sensors themselves; this knowledge is stored in the sensor processing modules.

$$\mathbf{x} = \begin{bmatrix} a_x \\ u \\ u_w \\ a_y \\ v \\ v_w \\ a_z \\ w \\ w_w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \\ \lambda \\ \varphi \\ h \end{bmatrix} := \begin{bmatrix} \textit{longitudinal body-related acceleration} \\ \textit{longitudinal inertial speed component} \\ \textit{wind speed} \\ \textit{latitudinal body-related acceleration} \\ \textit{latitudinal inertial speed component} \\ \textit{wind speed} \\ \textit{vertical body-related acceleration} \\ \textit{vertical inertial speed component} \\ \textit{wind speed} \\ \textit{rotation rate about } x-\textit{axis} \\ \textit{rotation rate about } y-\textit{axis} \\ \textit{rotation rate about } z-\textit{axis} \\ \textit{roll angle} \\ \textit{pitch angle} \\ \textit{yaw angle} \\ \textit{geographid atitude} \\ \textit{geographic longitude} \\ \textit{height above Earth ellipsoid} \end{bmatrix} \quad (2.1)$$

2.2 Vehicle control

Vehicle control performs the navigation task, monitors the actual state of the mission and computes the controls for automatic flight. Currently, a state controller is implemented combining feedforward and feedback components; more information about that may be found in [1] and [2].

2.3 Aircraft interface

The aircraft interface process receives measurement data from the aircraft onboard computer and transmits control values received from the vehicle control process in the case of automatic flight. State estimates from the multi-sensor state estimator are used to compute predicted measurements, the corresponding Jacobian vectors, which are the derivatives of the measurements related to the state vector, and confidence values, which are used to determine sensor faults for outlier removal.
The simulated measurements are corrupted with noise and other characteristic sensor errors [7]. Currently, accelerometers, rate gyros, artificial horizon, barometric height, aerodynamic data and GPS measurements in S/A-mode are simulated.

2.4 MPS / IPS interface

The image processing interface process receives state estimates from the multi-sensor state estimator. They are used to perform the selection of objects to be surveyed by the image processing system. Location information and window parameters of specific objects stored in a database determine if an object is detectable. If the object is

close enough and lies within the sweeping area of the pan and tilt platform, a command is sent to the platform controller consisting of the angles required to point towards the object. If no object is within reconnaissance range, the platform is directed towards the horizon.

After a short time interval, a message is sent to the image processing system consisting of the actual camera platform angles, the predicted state estimate and the objects to be surveyed. The image processing system starts a state estimation process with these initial values and sends the current state estimates back to the image processing interface process. Similar to the aircraft interface process, a confidence match is performed; if the state estimates stemming from the image processing system are regarded to be out of the confidence range, the image processing system is reset.

2.5 Landmark tracking

The landmark tracking system is based on REALIS which is a model based object tracking system that has been developed for the ROTEX free-flyer experiment during the D2-Spacelab-mission [3].

The REALIS system consists of four major parts:

- **Object modeling**

  The three dimensional geometry of objects is set up by points, edges and surfaces. For all these elements homogenous coordinates are used exclusively. When an object geometry is created its points get connected to edges and the edges get connected to closed surfaces. Ferguson curves are used for modeling edges as they only require tangent parameters to describe a general curve in 3D space.

  Every edge of an object has an indicator whether it should be used for image measurement or not. The horizon, for example, is modeled as a huge rectangle perpendicular to the surface of the Earth with only one edge visible. Furthermore whole objects may be enabled and disabled for image measurement via the MPS-IF. This is an easy way to implement multiple levels-of-detail (LOD) for one object. The object 'runway' for example has been modeled as a rectangle on the coarse level, with its adjacent taxiways on the 2nd, the outline of both thresholds on the third, and the single rectangular patches of the thresholds on the fourth level.

- **Scene tree**

  All objects are linked together in a tree structure which is shown in figure 2.2. Each node, i.e. object has its own local coordinate system. The positions of all son-nodes are given in local coordinates of their unique father's coordinate system. As in a tree there is always exactly one path from one node to another; the relative positions between all objects are described uniquely. The relative position between two nodes can either be fix or will be estimated.

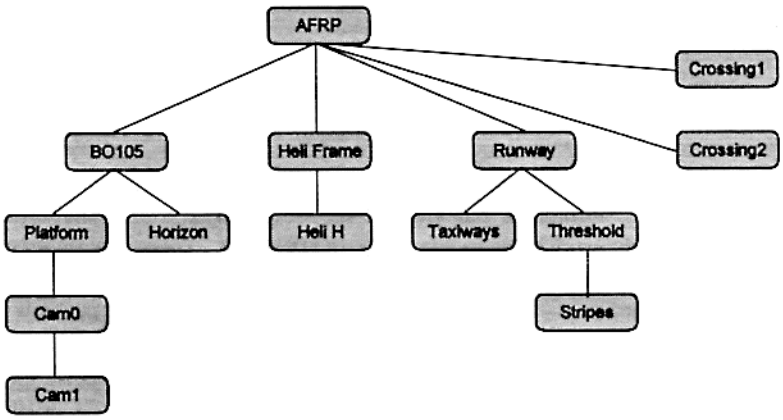


Figure 2.2: Scene tree of the landmark tracking system.

As root node the airfield reference point (AFRP) of the airport Braunschweig / Germany was chosen. The x-axis of this root coordinate system is pointing north, the y-axis is pointing to the east and the z-axis is pointing to the center of gravity.

- **Image measurement**

  Image measurement starts with making a visibility test for all object surfaces. Therefore, the normal vector of each surface is calculated and when it is pointing towards the camera the surface is called 'visible'. Then the homogenous coordinate transformation matrices have to be build. This is done by walking through the scene tree from the present object to the active camera and multiplying all local transformation matrices. Now the edges of the objects can be transformed into the image coordinate system, and measurement commands for the feature extraction library 'Cronos' [5] can be generated.

- **State estimation**

  An extended Kalman Filter [8] is used to estimate all state variables. This is a recursive process performed in two steps:

  Prediction of the state vector and the covariance matrix:

$$\mathbf{x}^{\bullet} = \Theta \cdot \hat{\mathbf{x}} + \mathbf{B} \cdot \mathbf{u}$$

$$\mathbf{P}^{\bullet} = \Theta \cdot \hat{\mathbf{P}} \cdot \Theta^{T} + \mathbf{Q}$$

  For state prediction a simple integration model is used.

  Innovation of the Kalman-gain, the covariance matrix and the state vector:

$$\mathbf{K} = \mathbf{P}^{\bullet} \cdot \mathbf{C}^{T} \cdot (\mathbf{C} \cdot \mathbf{P}^{\bullet} \cdot \mathbf{C}^{T} + \mathbf{R})^{-1}$$
$$\hat{\mathbf{P}} = (\mathbf{I} - \mathbf{K} \cdot \mathbf{C}) \cdot \mathbf{P}^{\bullet}$$
$$\hat{\mathbf{x}} = \mathbf{x}^{\bullet} + \mathbf{K} \cdot (\mathbf{y} - \mathbf{y}^{\bullet})$$

The Jacobian matrix C describes the relation between measurements and state variables. It is a function of object geometry, its position relative to the camera and the camera parameters like focal length and resolution.
The measurement variance matrix R is constant and contains the characteristic sensor errors.
After REALIS has been started via the MPS-IF it runs autonomously at a cycle time of 20ms and processes the two camera images sequentially. The current state estimate of the aircraft as shown in equation 2.4 is send back to the MPS every 40ms and the camera platform angles get updated from the MPS.

$$\mathbf{x} = \begin{bmatrix} a_x \\ u \\ x \\ a_y \\ v \\ y \\ a_z \\ w \\ z \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{bmatrix} := \begin{bmatrix} \textit{body-related longitudinal acceleration} \\ \textit{corresponding speed component} \\ \textit{position} \\ \textit{body-related lateral acceleration} \\ \textit{corresponding speed component} \\ \textit{position} \\ \textit{body-related vertical acceleration} \\ \textit{corresponding speed component} \\ \textit{position} \\ \textit{rotational acceleration about } x-\textit{axis} \\ \textit{rotational acceleration about } y-\textit{axis} \\ \textit{rotationnl acceleration about } z-\textit{axis} \\ \textit{rotation rate about } x-\textit{axis} \\ \textit{rotation rate about } y-\textit{axis} \\ \textit{rotation rate about } z-\textit{axis} \\ \textit{roll angle} \\ \textit{pitch angle} \\ \textit{yaw angle} \end{bmatrix} \quad (2.4)$$

2.6 Runway obstacle detection and tracking

The runway obstacle detection and tracking system is based on algorithms used for the detection of cars driving on a road [4]. The known area of the runway is scanned along horizontal search paths for small edge elements. If there is a local accumulation of edge elements found they are aggregated to an object outline. If the area of this new object is of valid size and shape for a truck and its gray level differs from the mean gray level of the white markings on the runway an obstacle is found. When this detection can be repeated for at least 5 video cycles an obstacle alert is sent to the MPS.

2.7 Image acquisition and distribution with the TIP-bus

The transputer image processing (TIP) bus is a fast backbone for transferring video images in real time. It will be explained in more detail in section 3. The process running on the monochrome frame grabber (MFG) is the TIP-Bus master. It controls image acquisition and distribution whereas the other TIP processes are slaves that receive images.
Two standard monochrome TV cameras are connected to the input of the MFG. As TV cameras have interlaced images the input selector of the MFG switches between camera 0 and camera 1 while field 0 and field 1 are digitized. Therefore, a new image with a reduced resolution of 320 × 256 pixel but with no delay time between the lines is available every 20ms. The processes running on the TIP-slaves use double buffering for measurement input and for drawing images for monitoring. The images to be drawn will be sent to a display process that is connected to a color graphics display for real time visualization of the results of the image processing system.

3. HARDWARE IMPLEMENTATION

The whole system is implemented on a parallel transputer system developed by Parsytec. Each transputer offers four high speed communication links with a band width of 20Mb/s and a minimum overlay time for communication setup. As standard processors the inmos T805/30MHZ have been chosen. For the 'number crunching' tasks like state estimation and image processing Motorola's PowerPC 601 with 66MHz respectively 80MHz is used. The MPS processes are running at a cycle time of 40ms, the IPS processes are running even at 20ms.
The hardware implementation as shown in figure 3.1 consist of three main parts:

3.1 Machine perception system

The MSSE module performing the multi-sensor state estimation is placed on a PowerPC 601; it is connected to the VC module performing the vehicle control process and the AC-IF module running the aircraft interface process, both placed on T805 transputers. The communication module CM is implemented to manage the data exchange with the camera platform controller. The MPS / IPS-IF handles the data flow between the MSSE and REALIS respectively RODT.
The connection to the simulation computer is realized by a RS422 link with the same data exchange structures as specified for future flight test.

3.2 Image processing system

Real time images are transferred via the TIP-Bus developed by Parsytec. It is a 32Bit bus running at 25MHz with a maximum transfer rate of 100MBs. The processes running on the TIP boards carry out the slightly complicated initialization of the TIP-Bus and, in the real-time phase, they take care of the correct sending and receiving of the images.
Two monochrome TV cameras with focal lengths of 25mm and 50mm are connected to the TIP-MFG. Digitizing of the camera signals is controlled by the MFG process running on a T805. The TIP-MPCs have 2MB of triple ported VRAM. The TIP-Bus controller process runs on the T425 that is also used by the PPC 601 for communication purposes. REALIS and RODT are both running on a PPC 601 of their own, so that they have direct access to the camera images and a maximum of computing power available. They are both connected to the IP-IF via a direct link. The TIP-CGD has a video controller that displays the contents of its VRAM on a standard VGA monitor. So the image draw buffers receive from the TIP-MPCs can be displayed or saved on a VCR. The process running on the T805 mounted on the TIP-CGD can save images as PCX files in the real-time phase.

3.3 Camera platform controller

The camera platform controller receives the position commands from the MPS and sends back the current yaw and pitch angle of the platform. The digital controller runs at a cycle time of 2ms and stabilizes the platform inertially by the use of two rate gyro signals.

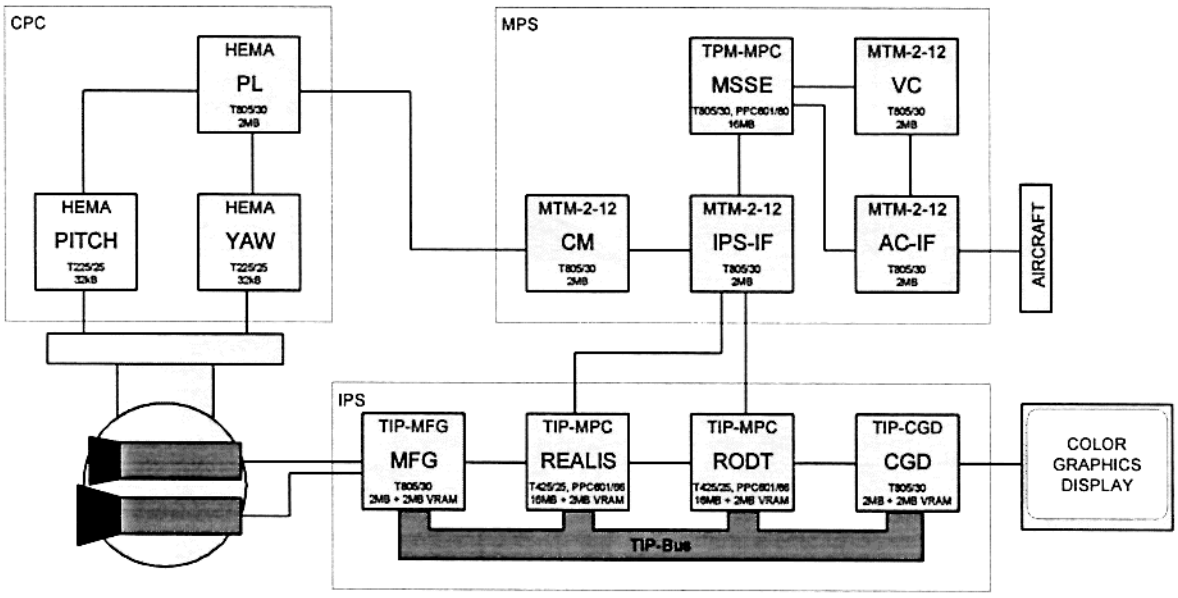


Figure 3.1: Hardware architecture of the system.

4. RESULTS FROM SIMULATION TRIALS

Several closed-loop simulation trials with a BO105 helicopter have been conducted under real-time conditions at a cycle-time of 40ms. The simulation environment was a model of the Braunschweig airport region [2]. The upper part of the snap-shots is the output of the weak tele camera (25mm), the lower part stems from the stronger tele-lens camera (50mm); the two-axis platform was engaged during the approach. Within the images the outlines of the currently active landmarks are drawn. They represent the internal estimate of the IPS about the relative position of the helicopter. The short lines mostly perpendicular to the outlines are the search paths of the edge detectors.

4.1 Horizontal flight loop

The flight plan of the go-around trip in the region of the Braunschweig airport with all its way-points and landmarks is shown in figure 4.1. The helicopter lifted-off at the point called 'Heli H East' and passed the first way-point 'Threshold 27' at a height of 30m and a speed of 20m/s. On the way to the next way-point 'Threshold 09'the snap-shots shown in figure 4.2 to 4.4 have been taken. These figures show the different LODs of the runway-model. The next way-point called 'Crossing 1' is about 3km west of the runway. During this flight period at a speed of 30m/s in a height of 60m the IPS was set inactive and the MPS accumulated an

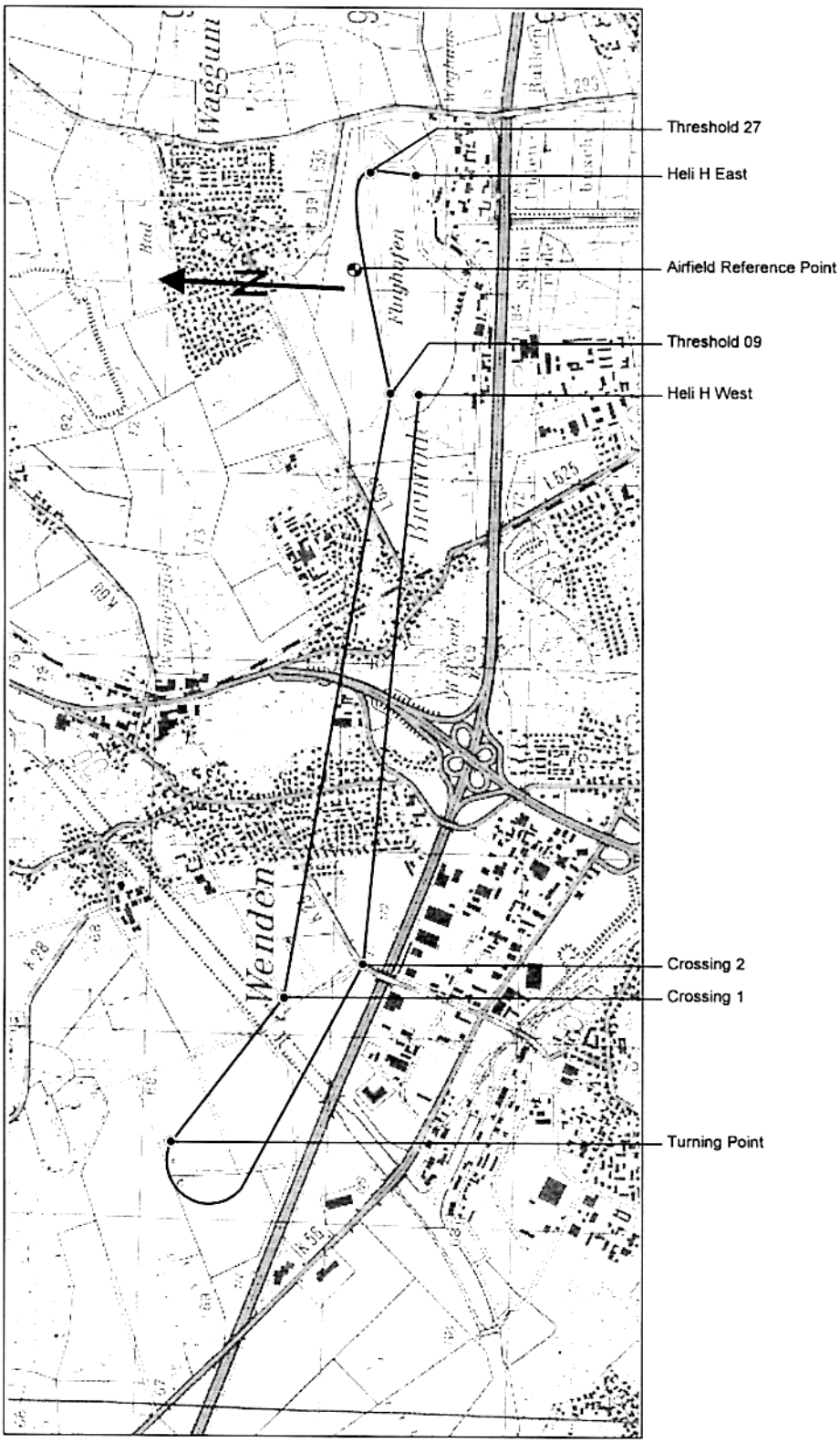


Figure 4.1: Flight plan of the circular flight test in Braunschweig.

error in the position estimate as GPS was used in S/A mode. When the IPS started tracking the 'Crossing 1' (fig. 4.5) the position accuracy could be increased significantly. After flying a sharp curve at the 'Turning Point', 'Crossing 2' was the next landmark (fig. 4.6). The flight loop ended with a successful landing approach on the 'Heli H West'. At a distance of 600m tracking of the taxiways was started; at a distance of 150m the frame and the white 'Heli H' were also used for tracking.

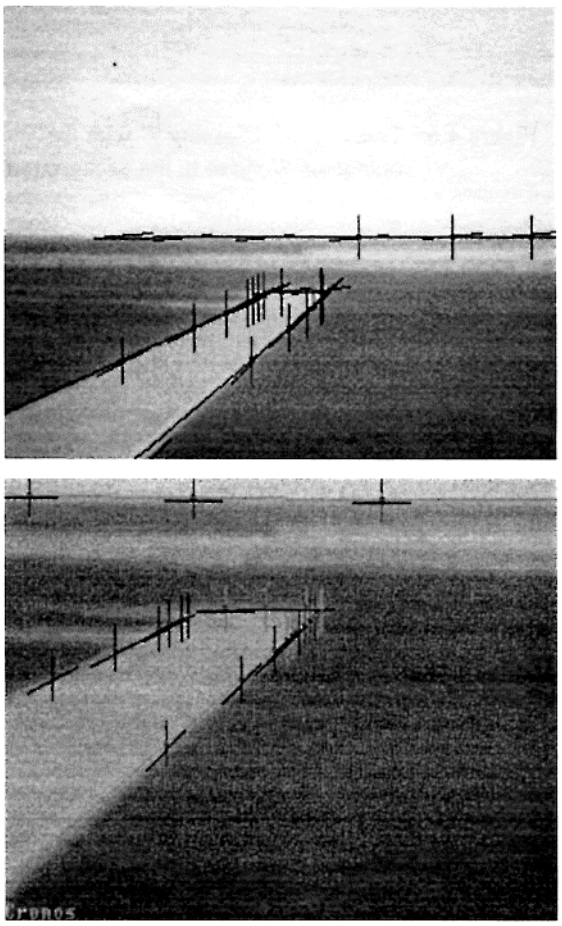
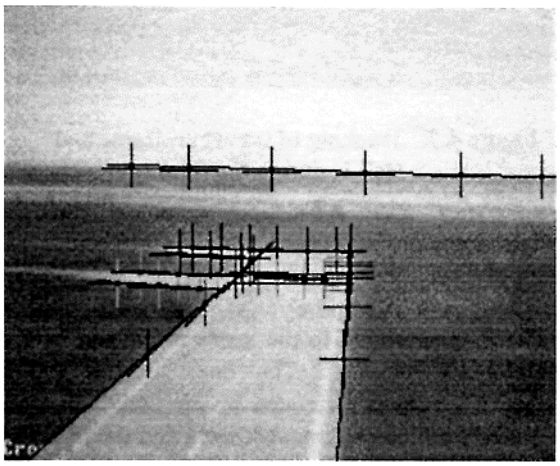


Figure 4.2: Runway at lowest LOD as a rectangle.



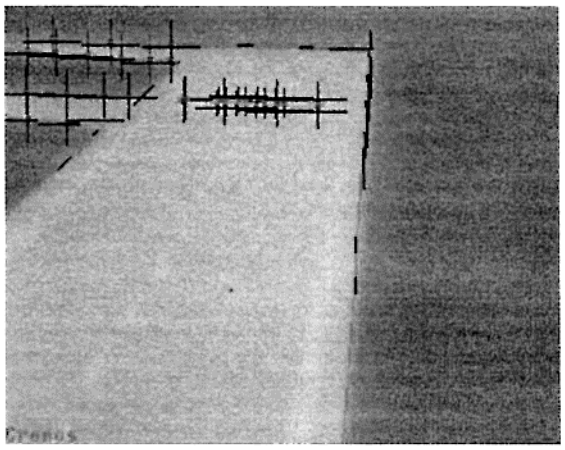


Figure 4.3: Runway and taxiways active; threshold had just been activated.

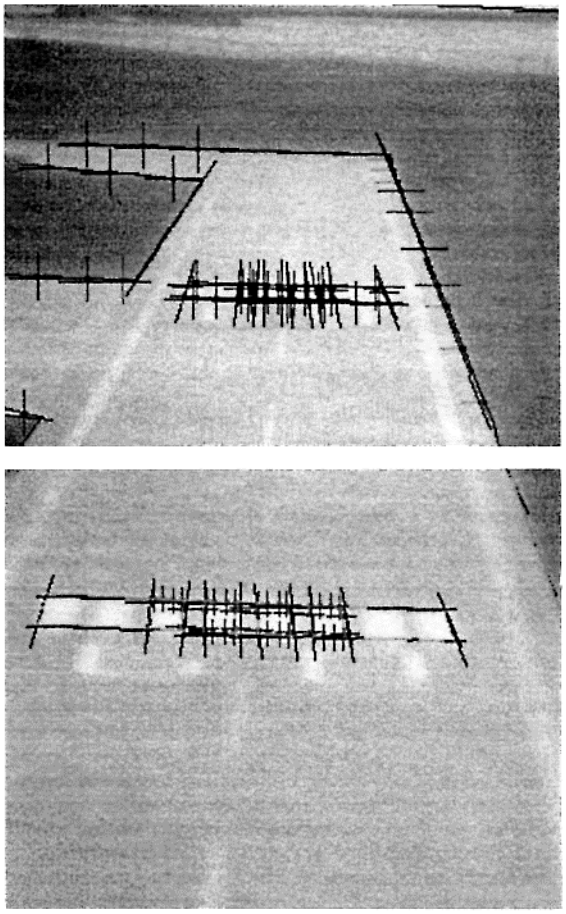


Figure 4.4: Low altitude fly-by at 'Threshold 09' with all LODs active.

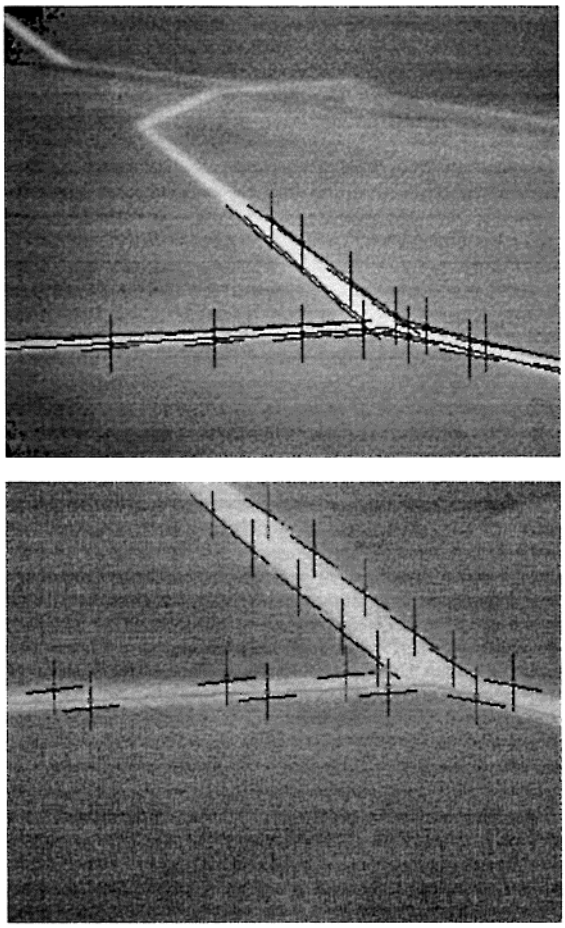
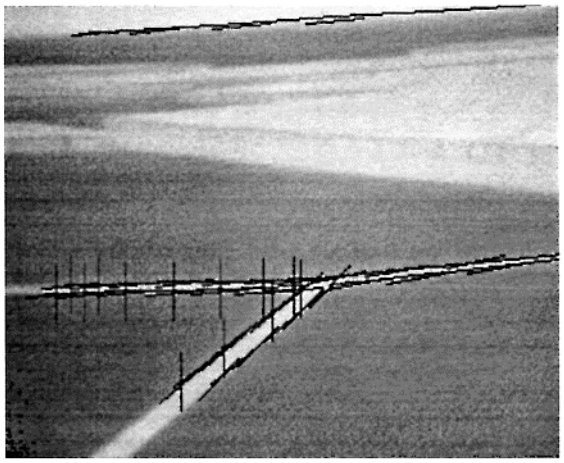


Figure 4.5: Tracking of 'Crossing 1' with the position error estimate still decreasing.



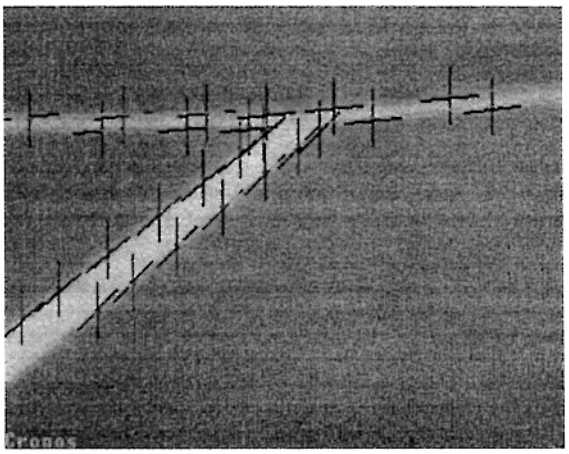


Figure 4.6: Tracking of 'Crossing 2' with the village of Wenden in the background.

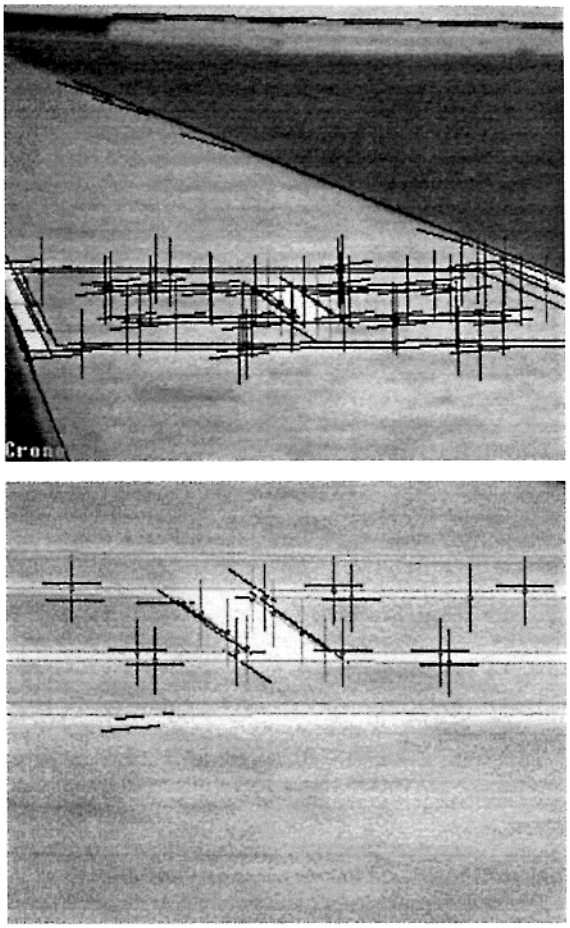


Figure 4.7: Tracking of taxiways, frame and Heli H during the final approach.

In the next three subsections timeplots from the flight loop are discussed. The figures show the longitudinal and lateral position of the aircraft plotted over the time axis. The lines marked 'TRUE' correspond to the states calculated by the simulation computer. The lines marked 'MPS' present the states estimated by the MSSE that are derived from conventional sensors and the IPS. The

'IPS' lines correspond to the state estimated by the image processing system that runs independently after activation. The 'GPS' lines are the simulated GPS signals with C/A code and selective availability. This signal has a 1Hz refresh rate like common GPS receivers.

4.1.1 Measurement of the runway

The IPS was activated at 38s with the rectangular runway model. After a short time of oscillation the state estimate of the IPS comes very close to the real values, with a very good longitudinal estimate as there are more measurement windows on the sidelines than on the endlines of the runway. The stronger oscillation of the lateral estimate are eliminated when the taxiways get activated at 44s.

4.1.2 Measurement of crossing 2

The position estimates show oscillations when the IPS gets activated. Their amplitudes die away over a period of about 4s and converge on the real values. This effect can be traced back to the distribution of the measurement windows in this scene. The concurrent estimation of range, height and pitch angle is based on features of the horizontal road in fig. 4.6. These feature are very indistinct when the IPS gets started at a distance of 800m, but they become clearer as the aircraft approaches the crossing.

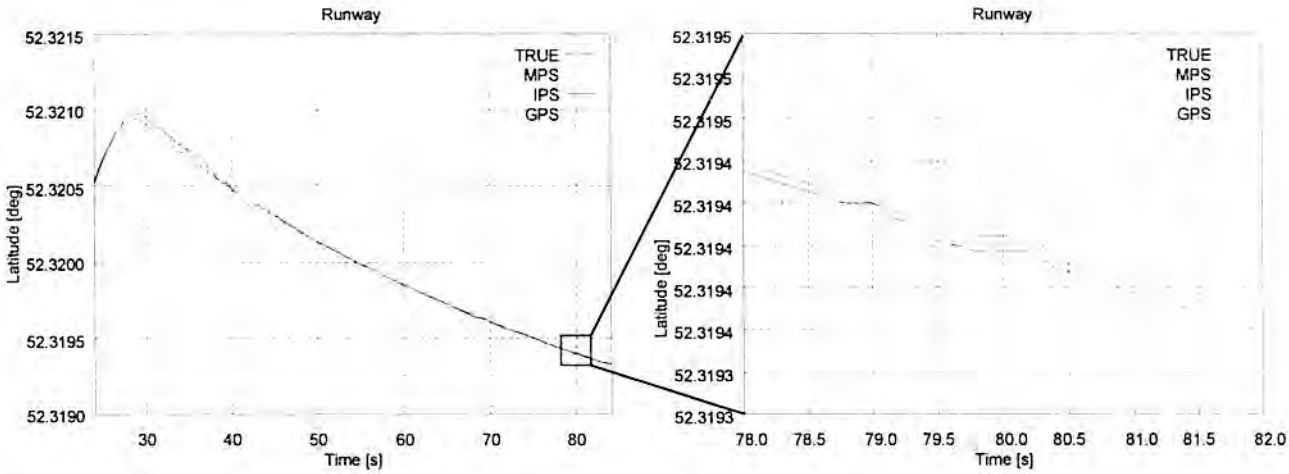


Figure 4.8: Latitude position of helicopter during runway fly-by.

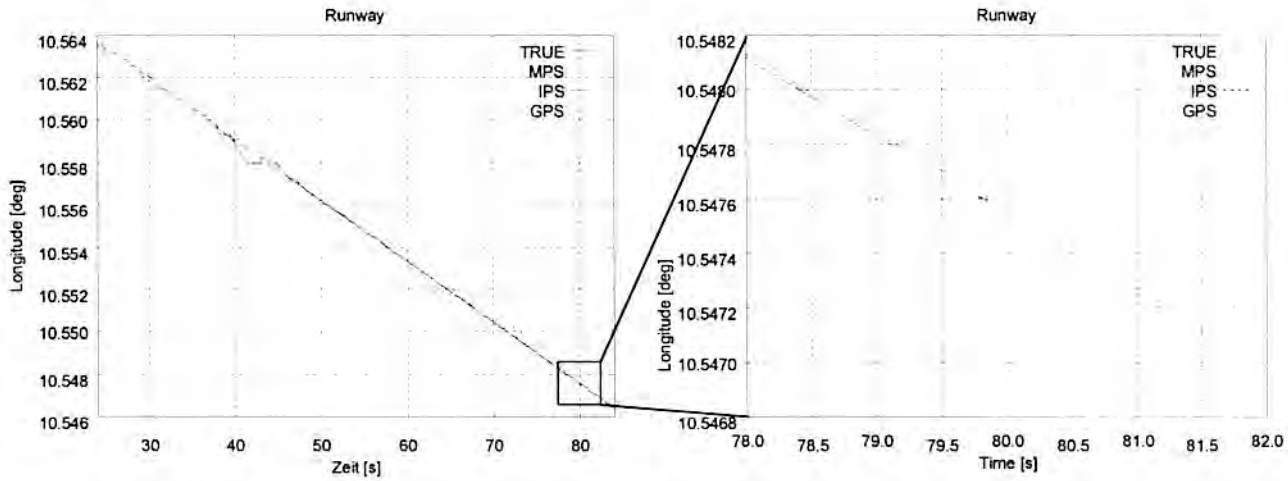


Figure 4.9: Longitude position of helicopter during runway fly-by.

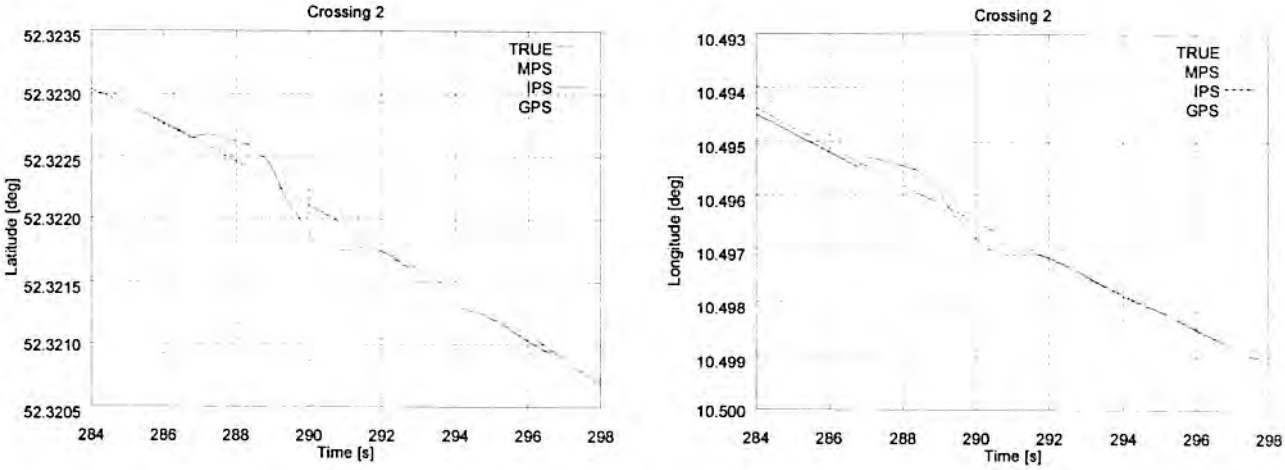


Figure 4.10: Latitude and longitude position of helicopter during crossing 2 fly-by.

4.1.3 Measurement of the helicopter landing spot

The aim of this flight was the western helicopter landing spot that is located on the taxiways. Tracking of the taxiways began in a range of 600m so that the position estimate got accurate enough to

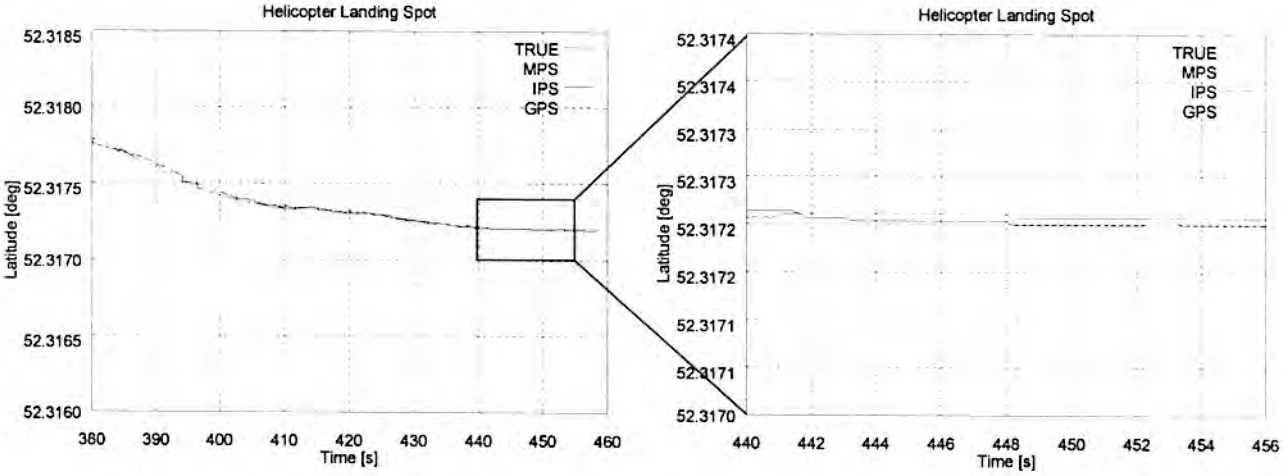


Figure 4.11: Latitude position of helicopter during final landing approach.

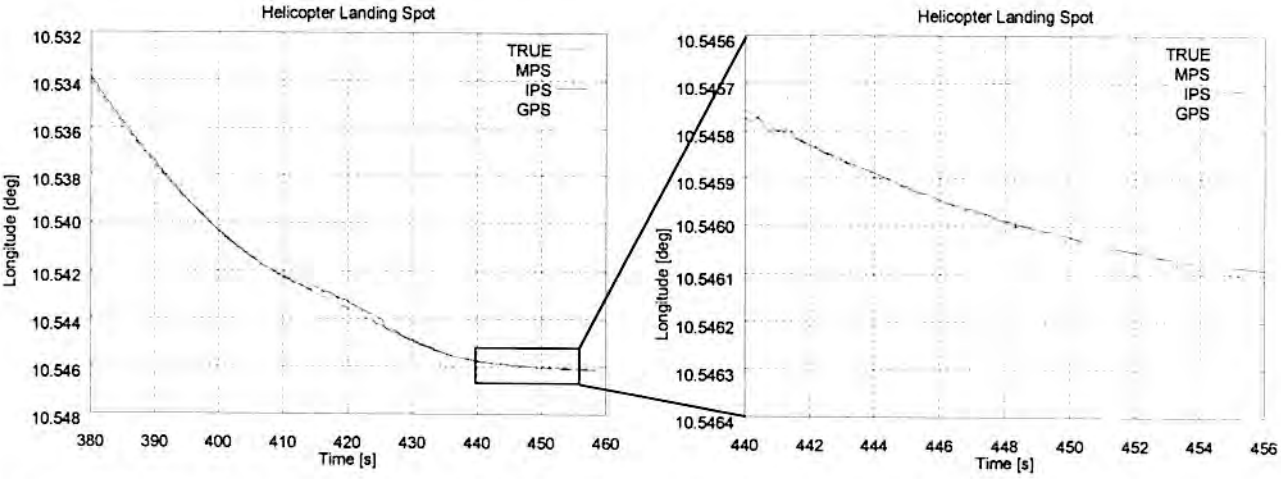


Figure 4.12: Longitude position of helicopter during final landing approach.

start tracking of the difficult object helicopter H and its frame in a range of 150m. In the final approach, the precision of the position estimate was below 0.5m.

4.2 Landing approach with obstacle detection

A straight landing approach of the BO105 helicopter at Threshold 09 was conducted to test the RODT module. Snap-shots are shown in Figure 4.13, where a truck, standing in longitudinal direction was used as an obstacle. At a distance of 2.1km it could be detected in the stronger tele camera and at a distance of 1.3km in the mild tele camera.

Figure 4.13: Obstacle detection during landing approach.

5. CONCLUSION

Results from real-time hardware-in-the-loop simulations for autonomous landmark navigation and landing approaches have been demonstrated. Image processing as a sensor for relative state estimation in combination with inertial sensors is able to deliver the accuracy required to perform precise automatic flight and landing guidance. Further developments we are working on are:

- Improvement of overall system performance by porting the algorithms to a new, more powerful and versatile hardware.
- Upgrading of the image generation system to get a more realistic image simulation at higher resolutions.
- Implementation of area based image operators.
- Development of a landmark and landscape database based on DTED and DFAD.

REFERENCES

1. S. Werner, S. Fürst, D. Dickmanns and E.-D. Dickmanns, "A vision based multi-sensor machine perception system for autonomous aircraft landing approach ", *Proc. SPIE AeroSense '96 Conf. on Enhanced and Synthetic Vision*, Orlando, FL, USA, 1996.
2. S. Werner, A. Buchwieser and E.-D. Dickmanns, "Real-time simulation of visual machine perception for helicopter flight assistance", *Proc. SPIE AeroSense '95 Conf. on Synthetic Vision for Vehicle Guidance and Control*, Orlando, FL, USA, 1995.
3. D. Dickmanns, "Knowledge Based Real-Time Vision". *Proc. 2nd IFAC Conf. on Intelligent Autonomous Vehicles '95*, Helsinki, 1995.
4. F. Thomanek, "Visuelle Erkennung und Zustandsschätzung von mehreren Straßenfahrzeugen zur autonomen Fahrzeugführung", Dissertation Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, München, Germany, 1995.
5. S. Fürst, "Cronos". Universität der Bundeswehr München, Institut für Systemdynamik und Flugmechanik, Internal Report 95-1, München, Germany, 1995.
6. F.-R. Schell, "Bordautonomer automatischer Landeanflug aufgrund bildhafter und inertialer Meßdatenauswertung", Dissertation Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, München, Germany, 1992.
7. R. Holland, "Digitale Verarbeitung von Flugversuchsdaten ohne Verwendung von Anti-Aliasing-Filtern am Beispiel eines Hubschraubers", *DFVLR-Mittelung 87-12*, Braunschweig, 1987.
8. A. Gelb (Ed.), "Applied Optimal Estimation",THE M.I.T. PRESS, Massachusetts Institute of Technology, Cambridge, MA, USA, 1984.