AGARD
Symposium on Advances in Guidance and Control of Precision Guided Weapons
Ottawa, Canada, May 1992

Visual Autonomous Automatic Landing of Airplanes

E.D. Dickmanns, F.-R. Schell

Aerospace Technology Department
Universität der Bundeswehr München
W. Heisenberg Weg 39
D-8014 Neubiberg
Federal Republic of Germany

Summary

A visual sensor data processing method has been developed and validated which allows to achieve on-board autonomous landing approaches in the visual flight regime with computing technology available today; sensors are a video-camera, inertial gyros and an air velocity meter. The key feature of the method is the reconstruction and servo-maintained adjustment by prediction error feedback of an internal spatio-temporal model about the process to be controlled (4D approach). This encompasses both the egomotion state of the aircraft carrying the sensors and the relevant geometric properties of the runway and its spatial environment. The efficiency of the approach is proved both in a hardware-in-the-loop simulation and in real test-flights with a twin-turbo-prop aircraft Do 128 of Dornier. For accuracy evaluation of the data gathered, the results of differential GPS and radiometric altitude measurements have been recorded simultaneously.

List of Symbols

ADC	analog digital converter
BVV	image sequence processing system
DAC	digital analog converter
DBS	three-axis motion simulator
u, v, w	translational velocities of the airplane
p, q, r	angular velocities
Φ, Θ, Ψ	Euler angles
x, y, H	distance of the airplane to the runway
η	elevator angle
F	thrust
ξ	aileron angle
ζ	rudder angle
x	state vector
u	control vector
$\Phi_{L,S}$	transition matrix (longitudinal, lateral motion)
A	Jacobian of system model
H	Jacobian of measurement model
K	Kalman gain matrix
P	error covariance matrix
Q	covariance matrix of system noise
R	covariance matrix of measurement noise
G	input matrix of noise process

1. Introduction

Electronic micro-miniaturization of sensors and processors is progressing to a stage where machines may be provided with the equivalent of the human sense of vision. Only a few years ago, the 1 million-instructions-per-second (MIPS) performance class for digital computers has been a magic limit; within a few years the 'GIPS'-class (Giga, i.e. 10^9 instructions per second) will be commonplace. This will allow to process high data rates as produced by imaging sensors in real time. Color video requires a data rate of the order of magnitude of 10 MB/s.

However, data rate is not the essential point since it is the information content of an image which is useful for achieving some goal based on image sequence processing. Within a high frequency image sequence there may be quite a bit of redundancy since the situation changes only slowly over time, in general. Therefore, the main task of real-time image sequence processing is to reduce data rates but to keep as much information about the process to be controlled as possible.

A uniformly grey image contains as many picture elements (that means 8 bit data points) as a highly structured one; yet, the information content of the former may be summarized *completely (without any loss)* by 1. the symbol 'uniformly grey' and 2. the number coding the grey level. For a 1K*1K pixel image this corresponds to a data reduction of the order of 10^{-6}.

This is well appreciated in static image processing where segmentation of regions with similar characteristics is a generally accepted first step; region or contour models allow much denser representation and storage of information than handling individual pixels. However, the same has not been true along the temporal axis in most approaches to image sequence processing. The 4D approach developed at UniBwM [1 to 5] combines both spatial and temporal models about processes in the world and fully exploits continuity conditions along all 3D space axes and along the time axis simultaneously, hence the name '4D approach'.

In this approach, all processing activities are geared to the next point in time when new measurements are going to be taken. There is no storing of previous measurement data for differencing or rate computation; this is of especial interest in image sequence processing where each measurement means huge amounts of data (10^5 to 10^6 Bytes), however, very much less new information once the notion of objects and their states has been introduced. The results of previous measurements and evaluations are stored in parameters and state variables of generically (structurally) defined object models including their motion behavior. In modern control theory this procedure is well known as recursive estimation (Kalman filters, Luenberger observers). This has been extended to perspectively mapped image sequences and was shown to be numerically very efficient. The flexibility of the approach has been demonstrated in the application areas of road vehicle guidance [2, 3], satellite docking [4], landmark navigation for autonomously guided vehicles on the factory floor and for landing approaches of aircraft. The latter one is the most demanding application up to now and will be discussed in the sequel.

2. Multi-point model of airplane dynamics

The most pretentious application of the 4-D approach is the automatically controlled landing approach of an airplane, because here a body is able to move within all six degrees of freedom (three translational and three rotational). According to Newton's law a state vector with 12 components is necessary for the description of the complete dynamics of the airplane. These equations are nonlinear. Contrary to the often used one-point airplane models, for this application a multi-point model description is used. The aerodynamic forces and moments are modelled separately on the wing and on the elevator unit (fig. 1) [6]. Wind effects, which have a considerable influence on the aircraft dynamics are included in this model too.

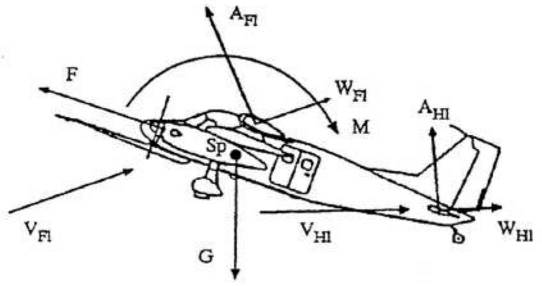


Fig. 1: Multi-point model of the Dornier Do128 airplane

The state vector **x** consists of the translational (u, v, w) and angular (p, q, r) velocities, the Euler angles Φ, Θ, Ψ, and the distance to the middle of the runway threshold (x, y, H), which is the origin of the coordinate system chosen. The four component control input vector **u** is assembled of the elevator angle η, thrust F, the aileron angle ξ and the rudder angle ζ. With this, the set of first order nonlinear differential eqs. may be written in the standard form for a 'dynamical model':

$$\dot{\mathbf{x}} = \mathbf{f}\,[\mathbf{x}(t), \mathbf{u}(t), (t)] \qquad . \qquad (1)$$

After linearisation around a (sliding) reference point x_0, u_0, the 12-th order system splits into two loosely coupled 6-th order systems: the longitudinal one with

$$\mathbf{x}_L = (u, w, q, \Theta, x, H)^T;\ \mathbf{u}_L = (\eta, F)^T \qquad (2a)$$

and

$$\dot{\mathbf{x}}_L = \mathbf{A}_L[x_0, u_0]\,\mathbf{x}_L + \mathbf{B}_L[x_0, u_0]\,\mathbf{u}_L\ ; \qquad (2b)$$

the lateral one with

$$\mathbf{x}_S = (v, p, r, \Phi, \Psi, y)^T;\ \mathbf{u}_S = (\xi, \zeta)^T \qquad (3a)$$

and

$$\dot{\mathbf{x}}_S = \mathbf{A}_S[x_0, u_0]\,\mathbf{x}_S + \mathbf{B}_S[x_0, u_0]\,\mathbf{u}_S\ . \qquad (3b)$$

The linear systems are the basis for developing a feedback controler, while motion simulation is performed using the original nonlinear equations (1).

3. Visual measurement model

For the imaging process from radiating points in 3D space onto the image plane the simple pinhole camera model is adopted (straight line perspective mapping). A point P in the runway plane has the coordinates (x_L, y_L, z_L) in an axis frame with the origin in the center of the runway threshold and the x-axis aligned with the runway center line (fig. 2). The position of the airplane in this coordinate system is at point (x, y, z), where a geodetic coordinate system with z_g in the direction of the Earth gravity vector is affixed to the aircraft center of gravity (cg); the x_g-axis in the horizontal plane is usually defined towards geographic north. For a right handed system the y_g axis then points towards east.

The angular orientation of the aircraft relative to this geodetic system is given by the three Euler angles Ψ, Θ, Φ, where the sequence of rotation is of importance for the final orientation; here, the z-sequence (Φ, Θ, Ψ) has been used since it yields relatively simple results in combination with the viewing direction control. In the aircraft-oriented coordinate system indexed f, the projection center of the camera has the coordinates (l_x, l_y, l_z). This is the origin for the camera-oriented coordinate system (indexed k), the angular orientation of which relative to the f-frame is ψ_{Ca} around the z_f-axis and θ_{Ca} normal to the xy_f-plane, positive upwards. The point P is mapped into the image plane at distance f (focal length of camera lens) normal to the x_{Ca}-direction with the coordinates z_{By} in line-direction (horizontally) and z_{Bz} in column-direction (vertically).

Using homogeneous coordinates, the transformations can be easily computed by 4*4 matrix multiplications; the following sequence is applied: From runway-coordinates translation $\mathbf{T}_g$ into geodetic coordinates, rotation $\mathbf{R}_f$ into airplane coordinates, translation $\mathbf{T}_c$ into platform-base coordinates, rotation $\mathbf{R}_{Ca}$ into camera coordinates and perspective projection P into image coordinates (for details see [9]). The nonlinear overall mapping equation may be written in vector form for the two image coordinates **z**, with **p** as camera mapping parameter vector

$$\mathbf{z} = \mathbf{h}\,[\mathbf{x}, \mathbf{p}\,] \qquad (4)$$

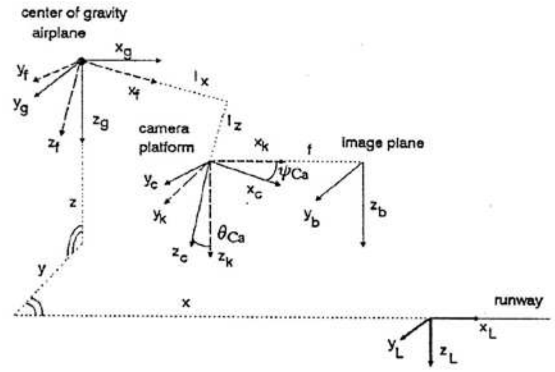


Fig. 2: Mapping a point from the runwy into the image plane of the camera [9]

With dx = X_L - x, dy = Y_L - y, H_z = height above ground (-z), and the denominator

$$D = b_{41}\,dx + b_{42}\,dy + b_{43}\,H_z + b_{44}\ , \tag{4a}$$

a point in the runway plane at (X_L, Y_L, 0) will be mapped into the image plane at

$$z_{By} = (b_{21}dx + b_{22}dy + b_{23}H_z + b_{24})/D \tag{4b}$$

$$z_{Bz} = (b_{31}dx + b_{32}dy + b_{33}H_z + b_{34})/D \tag{4c}$$

where the b_{ij} are coefficients depending on the transformation parameters (see [9]). Real measurements are always noise corrupted; therefore, for image interpretation through recursive estimation an additive noise term v(t) is assumed to be present with covariance matrix **R**. The Jacobian matrix of the right hand side of eq. (4) taken with respect to the aircraft state **x** is abbreviated with **H** (see eq. (7) below); this matrix and the coefficients b_{ij} become especially simple if the viewing direction is fixation controlled towards a point at the horizon where the runway borderlines intersect each other. This is obtained by a two-axis platform on which the camera is mounted. This platform is able to move in azimuth and elevation thus trying to keep the picture of the runway in the center of the image plane.

Eq. (4) is evaluated at ten different points in the runway plane where linearly extended intensity gradients may easily be found by intelligently controlled correlation with gradient templates (elongated ternary masks), see figure 3.

Knowing the shape of the runway, usually a rectangle, the appearance of the borderlines under perspective projection can be computed from the four corner points, given the relativ aspect conditions - in the definition chosen, exactly the aircraft state components. Eight windows are placed on the runway boundaries and two on the horizon in order to determine the roll angle.

Once an initialisation has been achieved, the search regions within the windows can be kept small since, due to motion prediction exploiting the dynamical model and previous control inputs, only the effects of disturbances have to be compensated by the search. Systematic changes in perspective projection are taken into account since all internal representations are simultanously in 3D space and time. This is equivalent to what psychologists call the 'Gestalt' phenomenon: When it is known what to look for, the interpretation of a scene may be much easier and less ambiguous than without any previous knowledge. This has been of great help in road recognition when, due to shadows from trees, intensity gradients are abundant and the highest correlation values do not at all correspond to road boundaries. Runway recognition, usually, is much more simple; however, taxiway entries and exits may be compensated for by dropping the measurements in these areas.

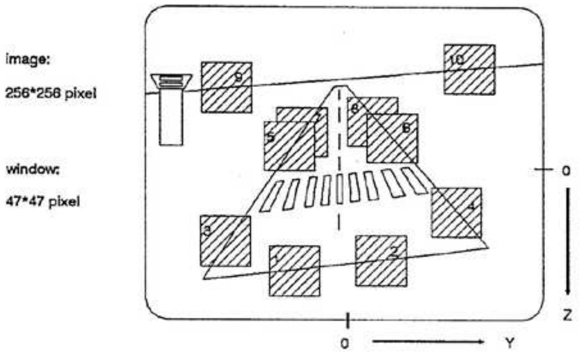


Fig. 3: Regions of evaluation (windows) within one image

4. Filter model for estimation of the complete state vector in real-time

In order to establish an Extended Kalman Filter (EKF) for the estimation of the complete state vector in real-time, the nonlinear equations of motion (1) are used. Based upon this 'dynamical model' the fundamental algorithm of an EKF may be found in [8] and looks like follows:

The transition matrix Φ over one cycle period $\Delta t = t_{i+1} - t_i$ is obtained from the linearised nonlinear system equations (1) with the Jacobian **A**:

$$\mathbf{A}[t; \hat{\mathbf{x}}(t\ |t_i)] = \left.\frac{\partial\, \mathbf{f}\,[\mathbf{x}(t), \mathbf{u}(t), t\,]}{\partial\, \mathbf{x}}\right|_{\mathbf{x} = \hat{\mathbf{x}}(t\ |t_i)} \quad . \tag{5}$$

Φ is defined as

$$\Phi\ [t_{i+1}, t_i; \hat{\mathbf{x}}(\tau\ |t_i)] = e^{\mathbf{A}(t_i)\Delta t} \quad . \tag{6}$$

P denotes the error covariance matrix, **Q** and **R** the covariance matrices of the system and measurement noises, while **G** is the input matrix of the noise process.

H is the Jacobian of the right hand side of the nonlinear measurement equations **h** [**x**, **p**, t_i] with **p** as parameters:

$$\mathbf{H}[t_i; \hat{\mathbf{x}}(t_i^-)] = \left.\frac{\partial\, \mathbf{h}\,[\mathbf{x}, \mathbf{p}, t_i\,]}{\partial\, \mathbf{x}}\right|_{\mathbf{x} = \hat{\mathbf{x}}(t_i^-)} \quad . \tag{7}$$

The final algorithm can be written as:

Extrapolation

$$\hat{x}(t_{i+1}^{-}) = \hat{x}(t_i^{+}) + \int_{t_i}^{t_i+1} \mathbf{f}\,[\hat{x}(t, t_i), \mathbf{u}(t), t]\,dt \qquad (8)$$

$$\mathbf{P}(t_{i+1}^{-}) = \Phi\,\mathbf{P}(t_i^{+})\Phi^{T} + \int_{t_i}^{t_i+1} \Phi\,\mathbf{G}(t)\,\mathbf{Q}(t)\,\mathbf{G}^{T}(t)\,\Phi^{T}\,dt \qquad (9)$$

Innovation (with $\mathbf{H} = \mathbf{H}[t_i;\hat{x}(t_i^{-})]$)

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^{-})\,\mathbf{H}^{T}\left(\mathbf{H}\,\mathbf{P}(t_i^{-})\,\mathbf{H}^{T} + \mathbf{R}(t_i)\right) \qquad (10)$$

$$\hat{x}(t_i^{+}) = \hat{x}(t_i^{-}) + \mathbf{K}(t_i)\left[\mathbf{z}_i - \mathbf{h}\,[\hat{x}(t_i^{-}), t_i\,]\right] \qquad (11)$$

$$\mathbf{P}(t_i^{+}) = \mathbf{P}(t_i^{-}) + \mathbf{K}(t_i)\,\mathbf{H}\,[t_i;\,\hat{x}(t_i^{-})]\,\mathbf{P}(t_i^{-}) \quad . \qquad (12)$$

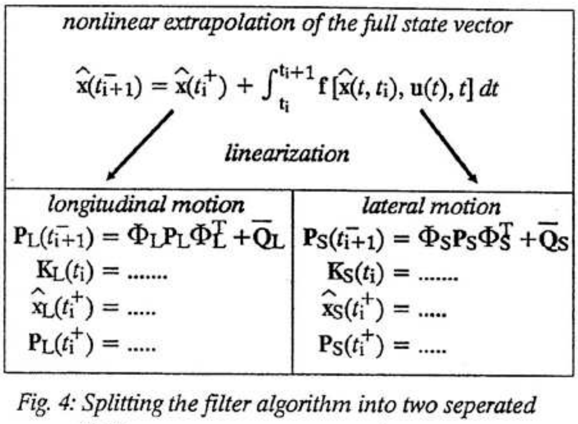
In the extrapolation step (eq. 8 and 9) the system and corresponding error models are integrated from the actual point in time (t_i^{+}) to the next one (t_{i+1}^{-}). After having received the new measurement values, the innovation is performed (eqs. 10 to 12), yielding the best estimate for the system state at time (t_i^{+}), which is in turn the basis for the next filter step.

Up to now, low cost microprocessor hardware is too slow for implementing the complete algorithm on a single unit in real-time. In order to reach real-time performance despite these difficulties, some steps have been investigated in order to split the algorithm from one set for 12 vector components into 2 sets indexed L and S for 6 vector components each running on parallel processors. Since the amount of computation needed is proportional to n^3, this reduces the computation time needed to 12.5 %. Finally, real-time performance has been achieved by substituting in the seperated equation (9) the following terms:

$$\int_{t_i}^{t_i+1} \Phi_L\,Q_L(t)\,\Phi_L^{T}\,dt \overset{!}{=} \overline{Q}_L \qquad (13)$$

$$\int_{t_i}^{t_i+1} \Phi_S\,Q_S(t)\,\Phi_S^{T}\,dt \overset{!}{=} \overline{Q}_S \qquad (14)$$

This allows to also split eqs. 9 to 12 into two sets resulting in two decoupled systems for the aircraft motion (fig. 4).



nonlinear extrapolation of the full state vector

$$\hat{x}(t_{i+1}^{-}) = \hat{x}(t_i^{+}) + \int_{t_i}^{t_i+1} \mathbf{f}\,[\hat{x}(t, t_i), \mathbf{u}(t), t]\,dt$$

linearization

longitudinal motion	*lateral motion*
$\mathbf{P}_L(t_{i+1}^{-}) = \Phi_L \mathbf{P}_L \Phi_L^{T} + \overline{Q}_L$	$\mathbf{P}_S(t_{i+1}^{-}) = \Phi_S \mathbf{P}_S \Phi_S^{T} + \overline{Q}_S$
$\mathbf{K}_L(t_i) = ...$	$\mathbf{K}_S(t_i) = ...$
$\hat{x}_L(t_i^{+}) = ...$	$\hat{x}_S(t_i^{+}) = ...$
$\mathbf{P}_L(t_i^{+}) = ...$	$\mathbf{P}_S(t_i^{+}) = ...$

Fig. 4: Splitting the filter algorithm into two seperated parts

These seperate main parts have been implemented on two different processors and run in parallel. For additional speedup, the sequential algorithm of Bierman has been used [12]. With this, the crucial step to real-time with $\Delta t = 60\,ms$ cycle time was achieved. By sticking to the nonlinear extrapolation for the system state, no knowledge about the system is lost.

The bulk of the measurement vector consists of optical features which are extracted from a single image delivered by a video camera. The rest of the measurements are inertial values obtained by gyros fixed to the aircraft. As indicated in section 3, a feature is a linearly extended intensity gradient within the runway plane. It is extracted by controlled correlation with ternary masks on the runway borderlines and on the horizon along special search paths [3]. The feature extraction algorithms are implemented on different parallel processors PPi in the image processing system BVV (see fig. 5), where one processor analyzes two features within 40 ms. In this approach maximally ten features are used per cycle presently (fig. 3).

5. Hardware

The main part of the underlying hardware is the image processing system BVV, developed at the Universität der Bundeswehr Munich [5, 7]. The BVV is a multi-processor system in which the different processors can communicate via a common system bus. Each pixel processor PPi (Intel 80286) has access to the video image from a CCD camera via a video bus and an ADC (fig. 5).

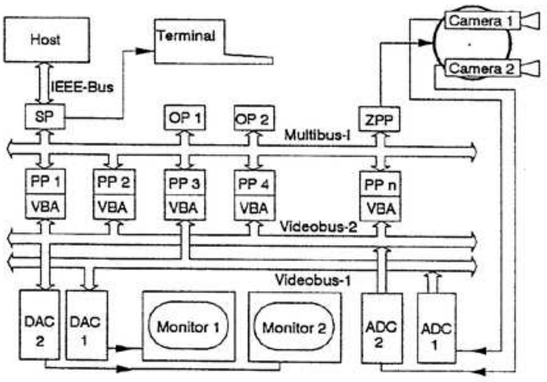


Fig. 5: Multi-processor system BVV (from [3])

The object processors OPi (80386) run the recursive estimation algorithms based on the 'Gestalt' idea of a perspectively mapped runway and the dynamical models, exploiting the feature data in conjunction.

Integrated into the BVV is the controller for the two-axis platform (ZPP). The host for the image processing system is a PC; both are connected by an IEC-bus. The PC is used only for initialization, as a link to the integrated computer of the airplane, and for data collection and final evaluation. For visual control of the process, a video monitor and a video recorder have been integrated (fig. 6).

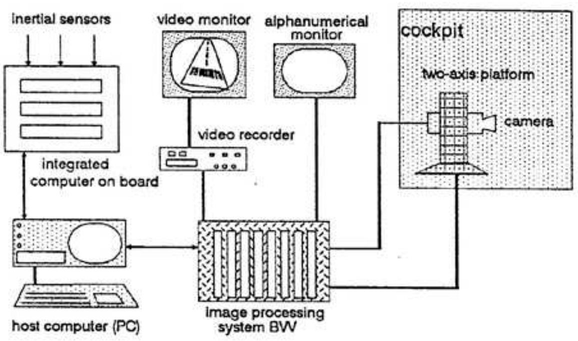


Fig. 6: Hardware architecture

6. Harware-in-the-loop (HIL) simulation

Primary results have been achieved in a hardware-in-the-loop simulation, which was especially built for investigations and developments in the field of image processing [10]. Real sensing and processing components may be integrated into this loop. The dynamics of the airplane and its view onto the runway can be simulated by a three-axis rotational motion simulator (DBS) and a graphics system (SGI 4D). The integration of the non-linear equations of motion is done by a digital computer which controls the other simulation facilities too (fig. 7).

For the simulations within the entire flight envelope, a complete state feedback controller has been developed with the latest theory of linear quadratic design with prescribed eigenstructures [11]. With this method, the Riccatti design is combined with the pole placement and eigencevtor specification, thus allowing to use the advantages of both methods. The nominal trajectory to be flown in 3D-space and time is given in fig. 8.

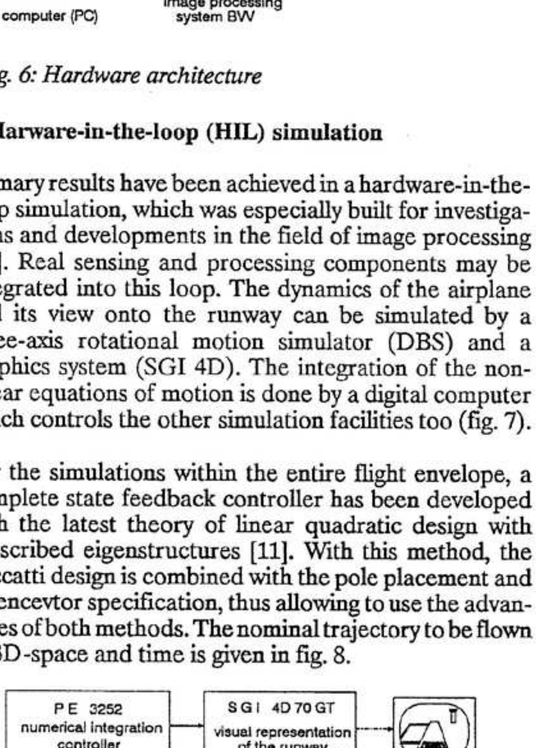


Fig. 7: Harware-in-the-loop-simulation

The advantage of HIL-simulation is, that on the one hand all sensors and data evaluation electronics (with their dirt effects, hard to be modelled), can be included in the investigation, but that on the other hand so called 'ground truth' data for the evaluation of system performance are readily available, since they are part of the numerical simulation. This type of simulator is widespread for both automatic guided weapon simulation and for pilot training; in dynamic vision, especially in the field of artificial intelligence, it is almost never used up to now. However, in several applications investigated at our institute it has proven to be a valuable tool for efficient software and system development in this area also. With the interfaces between modules designed in the same way as they are in the flight hardware, the preparations for real flight tests on the remote airfield in Braunschweig could be kept to a minimum.

Especially in the area of testing feedback control laws with respect to wind and gust responses, the well defined and easily repeatable disturbances in simulation have definite advantages over the irregular, nonrecurring ones in the real world.

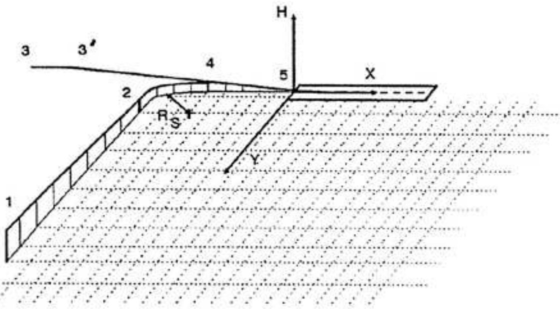


Fig. 8: Nominal trajectory shape

7. Test results

Closed loop performance with active control by the automatic visual guidance system could only be done in the simulation loop, since the airplane available from the Technical University Braunschweig did not have computer controllable actuators.

7.1 Closed-loop HIL-simulation results

The original flight hardware included in the simulation was: The CCD-TV-camera mounted on a two-axis (pan and tilt) platform for viewing direction control with angular rate feedback for inertial stabilization, the image sequence processing system BVV_2 as discussed in section 5, and the host-PC (80386) for system initialisation and data logging. Aircraft angular motion was realized by the DBS motion simulator for testing the viewing direction control platform; the inertial sensors in the aircraft have been simulated on the computer, however.

Wind effects have been incorporated by adding a (lateral) cross-wind component of -1 m/s starting at 800 m in front of the runway threshold; at 650 m an exponentially decreasing gust with a vertical component of 1 m/s and a cross-wind component of 2 m/s has been superimposed on the wind; the time constant of the gust was 6 s.

The simulation started at 950 m in front of the landing strip with a velocity of 50 m/s at an altitude of about 51 m; it ended at about 550 m down the runway with touch down of the landimg gear. The ensuing transition to taxying has not been investigated, since a completely different dynamical model would have been required both for state estimation and for control. It should be noted, however, that the vision sensor is well suited for vehicle guidance both in this and the following ground roll guidance task along the taxyways. The capability of performing these tasks has been demonstrated in essence with our road vehicle guidance program [2, 3, 5].

At about 250 m in front of the runway threshold the engines are throttled and the aircraft starts slowing down towards 43 m/s at touch-down; at the same time, the elevator is pulled in order to initiate the vertical flare decelerating vertical speed exponentially towards zero.

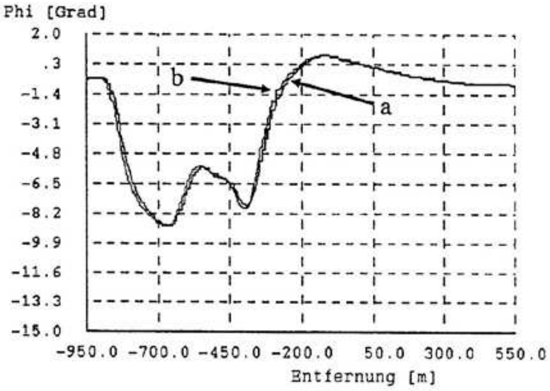


Fig.9a: a) estimated b) simulated

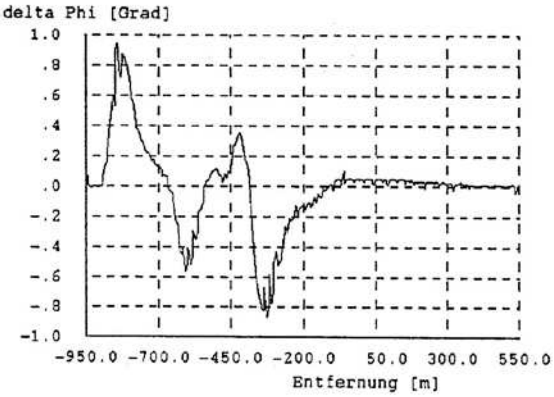


Fig.9b: Difference between estimated and simulated

Fig. 9: Roll angle in degree

As long as the runway threshold is visible, the distance to it is estimated within 5 m accuracy; after loosing the threshold out of sight at about 120 m distance from it, range x is merely predicted according to the model. The end of the runway can not be measured accurately enough from this low altitude to be of any use. Altitude above the runway could be estimated to within 1 m during approach and to less than half a meter over the runway; a rather large error briefly occurred during flare onset. This was due to the delay in the tilt viewing direction control loop at the sudden pitch rate change; it rather soon disappeared again.

The controller designed consists of a full state vector feedback with additional integral components for speed and altitude, for which a strict time history profile is essential during landing approach. The effect of the vertical gust component is hardly noticeable in the trajectory flown.

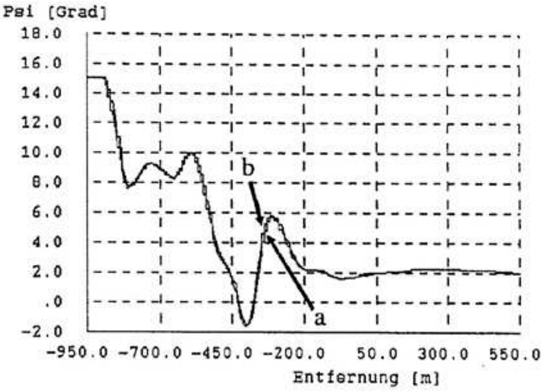


Fig.10a: a) estimated b) simulated

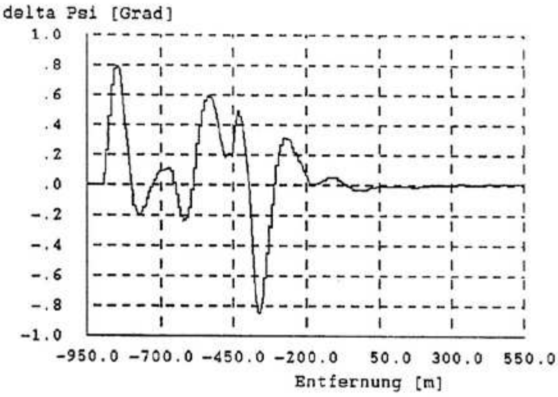


Fig.10b: Difference between estimated and simulated

Fig. 10: Yaw angle in degree

In the lateral degrees of freedom the maneuver started with a lateral offset from the runway centerline of 85 m at 950 m distance and a heading error of 15 degrees towards the centerline. Within half a kilometer the lateral offset was reduced to less than 5 m, and within 0.75 km the heading angle settled at about 2 degrees relative to the runway, apparently mainly for crosswind compensation. During the initial maneuver, roll angles of up to 8 degrees occurred; the estimation errors in both roll and yaw angles were always less than one degree in magnitude (fig. 9, 10). The lateral gust induced an estimation error in lateral position of about 3 m for a short time; otherwise it was always less than one meter soon after initialisation (fig. 11). By adding another Kalman filter for runway width estimation the system was improved to be able to deal with not exactly known runway parameters; a 29 m wide runway could be validated to within 1 m accuracy. When the runway threshold was lost out of sight due to the close approach, the average estimated value was within 0.4 m of the true one.

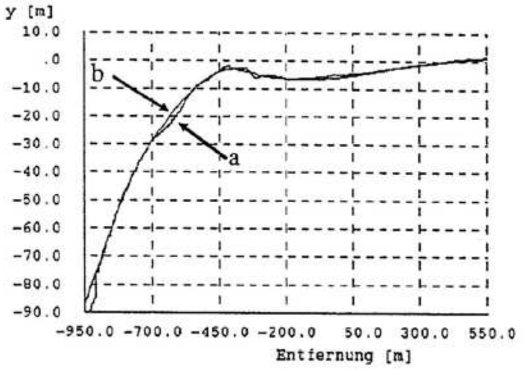


Fig.11a: a) estimated b) simulated

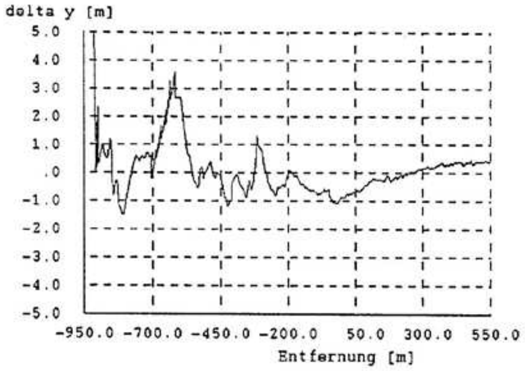


Fig.11b: Difference between estimated and simulated

Fig 11: Lateral offset y to the middle of the runway in meter

7.2 Flight test results in relative state estimation

After having gathered fundamental experience in simulation, test flights have been performed with the twin-turboprop airplane Do 128 of the University of Braunschweig. For evaluation of the accuracy of the optical results, a differential GPS system recorded the flight data in parallel to the optical system. The results in the translational degrees of freedom are given in fig. 12 to 14.

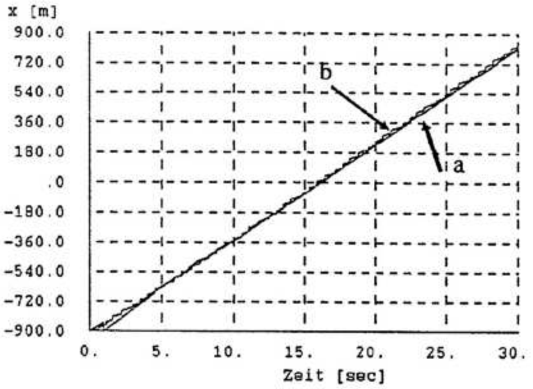


Fig.12a: a) optical b) GPS system

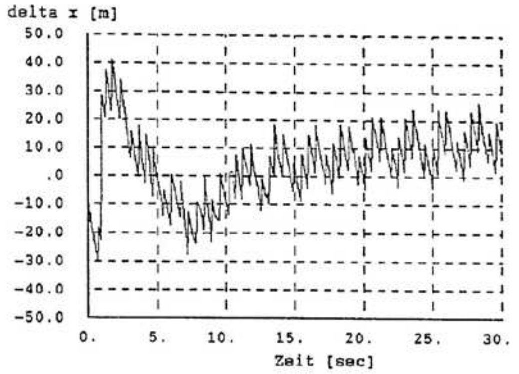


Fig.12b: Difference between optical and GPS system

Fig. 12: Distance to the runway x in meter

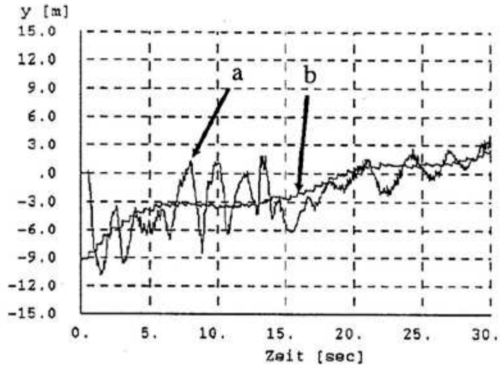


Fig.13a: a) optical b) GPS system

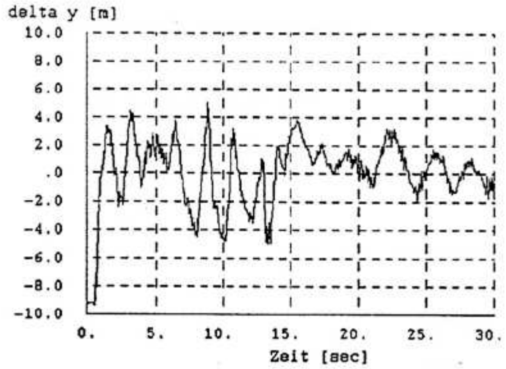


Fig.13b: Difference between optical and GPS system

Fig 13: Lateral offset y to the middle of the runway in meter

It can be seen clearly from fig. 13, that the optical evaluation process is getting better the nearer the airplane approaches the runway. The reason is, that the image of the runway is getting larger and the optical measurement values become better. For the evaluation of the altitude, the results of the GPS system showed errors. For comparison, a radiometric altitude-above-ground measurement system was used in addition, in order to record this value too.

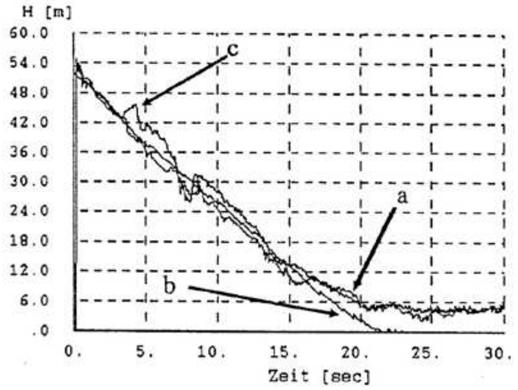


Fig.14: Altitude in meter a) optical b) GPS c) radiometric

In order to even allow approaches to airports, for which the width of the runway is not known, an additive Kalman Filter was used to estimate this value. The width of the runway in Braunschweig is 29 m (difference between the white lane markings). Fig. 15 shows the result of the estimation, where the mean value estimated till about 200 m distance and 20 m elevation relative to the runway threshold is 29.1 m. After having lost this line within the image, the estimation is stopped and the mean value is taken.

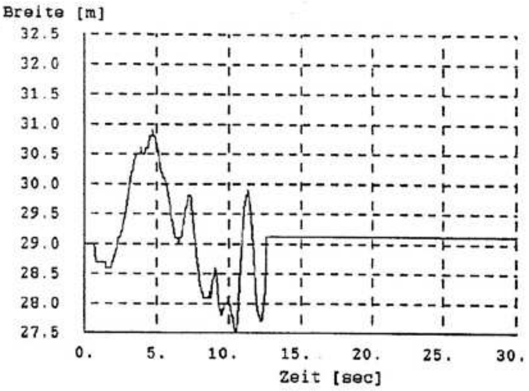


Fig. 15: Estimated runway width

8. Conclusion

Taking advantage of both spatial and temporal models for motion processes of objects, today's microprocessors, already, allow real-time image sequence processing, dynamic scene understanding and visual motion control in visually not too complex scenes in the 10 to 30 Hz range. The powerful microprocessors of the near future (200 MIPS-class), interconnected by high data rate communication networks, will enable the sense of vision for machines in well structured environments.

The method developed for video image sequence evaluation in the optical range may easily be adapted to other sensors like infrared, low-light-level TV or even imaging radar. With these, night vision or all-weather visual capabilities may become possible.

Combining visual and inertial sensor data evaluation has complementary beneficial effects: motion blur at high angular rates will deteriorate image processing; high rates, however, can easily be measured inertially at low cost. Inertial sensors become expensive when they have to be trimmed to long term stability. Static references for long term stabilization, however, can be measured easily by image processing, once the sensors have been roughly stabilized inertially. Therefore, good overall performance at low cost may be expected by a proper combination of both sensors.

A similarly beneficial effect may be achieved for navigation by combining vision with GPS: The latter provides a rough estimate of geographic coordinates, so that an intelligent vehicle capable of visual landmark navigation can start its visual search for landmarks in a rather small search area; accurate position determination relative to the landmark may then be achieved by visual tracking over time taking models for the ego-motion into account.

This new technology will allow autonomous on-board navigation and control capabilities unachievable up to now.

Fully autonomous control in a landing approach till touch-down has been demonstrated in a hardware-in-the-loop simulation in real time (16 Hz) including wind and moderate gusts. The same hard- and software has been installed in a twin-turboprop aircraft; within five days of testing, first results in relative state estimation during manually flown landing approaches have been obtained. These results compare well to differential GPS and radio altimeter results for the same flights.

References

[1] Dickmanns E.D.: Dynamic Vision for Locomotion Control - an Evolutionary Path to Intelligence. CCG-Lehrgang SE 3.02, 1991.

[2] Dickmanns E.D., Mysliwetz B., Christians T.: "Spatio-temporal guidance of autonomous vehicles by computer vision". IEEE- Trans.on Systems, Man and Cybernetics, Vol.20, No.6, Nov/Dec 1990, Special Issue on Unmanned Vehicles and Intelligent Robotic Systems, pp 1273-1284

[3] Mysliwetz B.: Parallelrechner-basierte Bildfolgen-Interpretation zur autonomen Fahrzeugsteuerung. Dissertation an der Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, München, 1990.

[4] Wünsche H.J.: Erfassung und Steuerung von Bewegungen durch Rechnersehen. Dissertation an der Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, München, 1987.

[5] Dickmanns E.D., Graefe V.: a) "Dynamic monocular machine vision", b) "Application of dynamic monocular machine vision". J. Machine Vision Application, Springer-Int., Nov. 1988, pp 223-261.

[6] Proskawetz K.O.: Ein Beitrag zur Genauigkeitssteigerung bei der Parameteridentifizierung nichtlinearer Prozesse am Beispiel der Flugzeugbewegung. Dissertation TU Braunschweig 1989.

[7] Graefe V.: Dynamic Vision Systems for Autonomous Mobile Robots. IEEE Workshop on Intelligent Robots and Systems - IROS'89; Tsukuba, Sept. 89.

[8] Maybeck P.S.: Stochastic Models, Estimation and Control, Vol.1, 2; Academic Press, New York 1982

[9] Schell F.-R.: Bordautonomer automatischer Landeanflug aufgrund bildhafter und inertialer Meßdatenauswertung. Dissertation an der Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, München, 1992.

[10] Dickmanns E.D., Zapp A., Otto K.D.: "Ein Simulationskreis zur Entwicklung einer automatischen Fahrzeugführung mit bildhaften und inertialen Signalen". In Breitenecker, e.a. (Hrsg.); Simulationstechnik; Informatik-Fachberichte 85, Springer-Verlag, 1985

[11] Otto K.D.: Linear-quadratischer Entwurf mit Strukturvorgaben. Dissertation Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, München, 1990

[12] Bierman G.J.: Factorization Methods for Discrete Sequential Estimation. Academic Press, New York 1977.