

Direct Homography Control for Highly Maneuverable Vehicles

Alexander Schaub and Tilman Bünte
Institute of System Dynamics and Control
German Aerospace Center
82234 Wessling, Germany
Email: firstname.lastname@dlr.de

Darius Burschka
Institut für Informatik VI
Technische Universität München
85748 Garching bei München, Germany
Email: burschka@cs.tum.edu

Abstract—This paper introduces a new vision based controller that uses the entries of a homography matrix to control a highly maneuverable mobile robot. It aims to reduce the sensitivity of the control to calibration errors through connection of the perception and control in the data space of the sensor. The control law is formulated in a way that it is only dependent on features of the current image without measuring or calculating any position in the Cartesian space. Only the target image must be known and the target position should be estimated relatively to the feature plane. An evaluation of the robustness of the controller is presented and verified by simulation.

I. INTRODUCTION

Vision based control techniques have been research topic for over 40 years [1], whereas the basic principle is the use of visual information in the feedback loop to complete positioning or navigation tasks [2]. It is a crossover between image processing and control theory and therefore the characteristics of the visual sensor and of the robot have to be taken concurrently into consideration when designing the vision based control scheme. Good surveys about vision based control or also called visual servoing can be found e.g. in [1], [3]. There are two basic classes of visual servoing schemes namely image-based visual servoing [4] and position-based visual servoing [5]. Image-based visual servoing is also called 2D visual servoing, since the task is expressed directly in the image space by describing the control law e.g. in dependence of 2D feature points. Contrary to that, the position-based visual servoing or 3D visual servoing formulates the control task in Cartesian coordinates. The pose is estimated by the essential matrix, whereas the target must not be planar and the motion between the two images must not be a pure rotation [6]. Additionally, combinations of image-based and position-based visual servoing, so called 2.5D visual servoing, were proposed e.g. in [7]. The task function contains variables in the Cartesian space as well as in the image space to overcome shortcomings and use strengths of both visual servoing types. Vision based control techniques have the potential to be used for a robust, direct, reactive control, as the position is measured directly without building an environment map by data accumulation first and then localizing itself. Nevertheless, even the control law of the standard 2D approach requires the depth values, z in camera direction, of the features, which

impairs the direct use of the sensor. One solution for a vision based control scheme without estimating or measuring the depth value is the use of a homography matrix [8], which is explained in detail in Section 2.

In [9] vision-based car platooning, i.e. an autonomous vehicle following a leading vehicle, is realized by tracking the back of the leading vehicle and calculating the homography matrix to match a target picture, which was taken by a camera located directly behind the leading vehicle. After the calculation the homography matrix is decomposed in order to get the respective translation and rotation. Hence, the control task itself is then described in Cartesian coordinates.

A control scheme that tries to completely relinquish Cartesian variables in the control law is proposed in [10], where they try to control the entries of the homography matrix directly to increase the robustness and decrease the computational effort. This is not possible for all motion primitives and hence a decomposition of the homography matrix has to be executed for certain trajectories.

In this paper, we want to propose a direct homography based control method for a highly maneuverable vehicle as described in [11]. The special kinematics together with a well chosen control-interface support an advantageously structured mapping of the entries of the homography matrix into the control space. The robot can be controlled only by measurements in the image space without neither any online measurements or estimations in the Cartesian space nor a decomposition of the homography matrix.

After deriving the control law in Section 2, the robustness of the controller is evaluated in Section 3. Simulation results are presented in Section 4, before conclusions are drawn in Section 5.

II. DIRECT HOMOGRAPHY CONTROL

In this Chapter the homography matrix is described in detail and how it is utilized via decomposition. After that a control method that directly sets the entries is derived together with a simplification for highly maneuverable vehicles. From those tools the controller structure is deduced and the analytical solution is discussed.

A. Homography Matrix

The motion of the camera can be reconstructed from two views if enough matching feature points can be found in both images. The standard eight-point algorithm for calculating the rotation R and translation t from the essential matrix can only be applied to generally distributed feature points, which means that they must not lie on a planar surface [6]. For the planar case the reconstruction must be done by homography. Figure 1 depicts the basic principle with the camera starting at pose o_1 and being moved by rotation R and translation t to the pose o_2 . The plane P is at least partially visible from both poses and a feature point $p \in P$ is projected into the image plane of o_1 as m_1 and into the image plane of o_2 as m_2 .

Let \hat{m}_1, \hat{m}_2 be the unscaled 3D coordinates of p in the

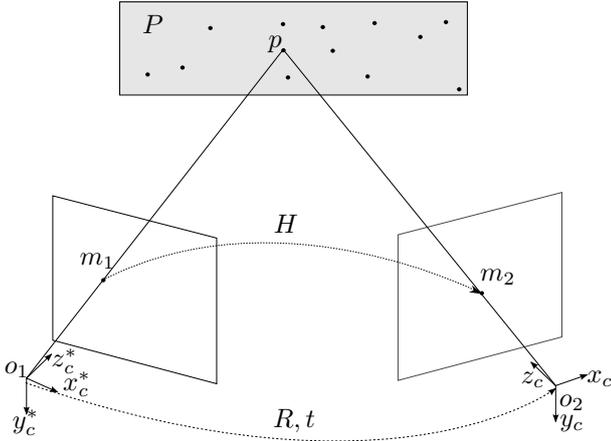


Fig. 1. Homography Principle

respective camera frames, hence the transformation between the two systems is:

$$\hat{m}_2 = R\hat{m}_1 + t \quad (1)$$

Let $n \in \mathbb{R}^3$ be the unit normal vector to plane P expressed in the reference frame o_1 and $d > 0$ the distance from P to the optical center of the camera pose o_1 .

$$n^T \hat{m}_1 = d \quad (2)$$

Reformulating (2) and combining it with (1) leads to:

$$\hat{m}_2 = R\hat{m}_1 + t \frac{1}{d} n^T \hat{m}_1 \Leftrightarrow \hat{m}_2 = (R + \frac{1}{d} t n^T) \hat{m}_1 \quad (3)$$

Since the translation t is scaled by the distance d , the homography matrix expresses the relation between both projected points up to a scaling with $\lambda_i m_i = \hat{m}_i, i \in 1, 2$:

$$m_2 \approx H m_1 \quad (4)$$

If there are more than four feature correspondences, whereas no more than 3 points are collinear, the homography matrix can be calculated and the scaling can be determined, see Lemma 5.18 and Algorithm 5.2 in [6].

By calculating a matrix only from the 2D images points it is possible to receive an implicit representation of the 3D motion. The standard method to explicitly calculate the 3D motion is

the decomposition of the homography matrix, see e.g. [9]. The decomposition has a high computational effort consisting of several steps and provides not an unique but four solutions, of which two can be canceled out directly since they do not meet the positive depth constraint of a physically possible solution. Different techniques exist to choose one solution from the two remaining by e.g. estimating the real normal vector [12].

B. Direct Jacobian Control

Even though efficient decomposition methods exist a controller settled in the Cartesian space is still dependent on an abstracted sensor signal. Noise and errors in the measurements will be increased by the decomposition. A more robust and more efficient way is to work directly in the image space, whereas the aim is to control the entries of the homography and therefore the pose of the mobile robot.

The first step is to calculate a Jacobian matrix J to see how changes of the position effect the entries of H .

$$\dot{h} = J\dot{\xi} \quad (5)$$

Whereas h are the entries of H stacked to a vector:

$$h = [h_{11} \ h_{12} \ h_{13} \ h_{21} \ h_{22} \ h_{23} \ h_{31} \ h_{32} \ h_{33}]^T \quad (6)$$

The state vector ξ consists here of the position deviation between current pose and goal pose expressed in the current camera coordinate system. Depending on the state vector ξ the Jacobian could get very complicated, but the goal here is to control a mobile robot. Consequently, the translation can be limited to the ground plane, which is denoted by the x, z plane of the camera, and only rotations around the y axis pointing up in the camera coordinate system are possible. With those restrictions the homography matrix simplifies to:

$$H = \begin{pmatrix} \cos(\alpha) + \frac{t_1 n_1}{d} & \frac{t_1 n_2}{d} & \sin(\alpha) + \frac{t_1 n_3}{d} \\ \frac{t_2 n_1}{d} & \frac{t_2 n_2}{d} + 1 & \frac{t_2 n_3}{d} \\ -\sin(\alpha) + \frac{t_3 n_1}{d} & \frac{t_3 n_2}{d} & \cos(\alpha) + \frac{t_3 n_3}{d} \end{pmatrix} \quad (7)$$

Here n_i are the components of the normal vector n , t_i the entries of the translation vector t , and α is the angle of rotation R .

The control interface $\dot{\xi}$ of the highly maneuverable vehicle consists of the velocities v_x, v_y and the yaw rate $\dot{\psi}$. Those values are described in the vehicle's coordinate system, but, if it is assumed that there is no translational displacement between the coordinate systems, they could easily be transferred to the camera's coordinate system and the following relations hold:

$$\begin{aligned} \dot{t}_1 &= -v_y \\ \dot{t}_3 &= v_x \\ \dot{\alpha} &= \dot{\psi} \end{aligned} \quad (8)$$

Since all other variables of (7) are static and not controllable, their time derivatives are zero. Calculating the time derivative of (7) and using (6) and (8), Equation (5) becomes:

$$\dot{h} = \begin{pmatrix} 0 & -\frac{n_1}{d} & -\sin(\alpha) \\ 0 & -\frac{n_2}{d} & 0 \\ 0 & -\frac{n_3}{d} & \cos(\alpha) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{n_1}{d} & 0 & -\cos(\alpha) \\ \frac{n_2}{d} & 0 & 0 \\ \frac{n_3}{d} & 0 & -\sin(\alpha) \end{pmatrix} [v_x v_y \dot{\psi}]^T \quad (9)$$

This Jacobian matrix allows now a direct control of the entries of the homography matrix, as far as this is possible with the described mobile. Here it is clearly visible that d , as mentioned above, does not have to be known and an up-to-scale control is possible.

The ideal Jacobian would be a diagonal matrix with entries not close to zero, which means that the system is decoupled and well observable. Even though this is not the case in general here, different simplifications of J can apply in dependence of n . It seems that v_x and v_y can be directly calculated from h_{12} and h_{32} , but n_2 is very likely to be zero, since most plane surfaces in reality are perpendicular to the ground plane. Nevertheless, if n_2 is sufficiently large, this should be exploited.

Moreover, if the camera is oriented perpendicular to the target plane the rotation and the translation can be directly read out from the homography matrix, since a normal vector $n \in \{[1 \ 0 \ 0], [0 \ 0 \ 1]\}$ simplifies H even further. This could also be kept in mind when designing artificial target poses.

C. Closing the control loop

Now that the relation between the vehicle's control interface and the homography matrix is known, it can be utilized to set up the control loop, which is depicted in Figure 2. The control goal here is to move the mobile robot exactly to the pose of the goal image, therefore the desired homography matrix equals the identity matrix $H^* = I$, as easily can be seen from (3).

The deviation Δh of the current homography matrix H to

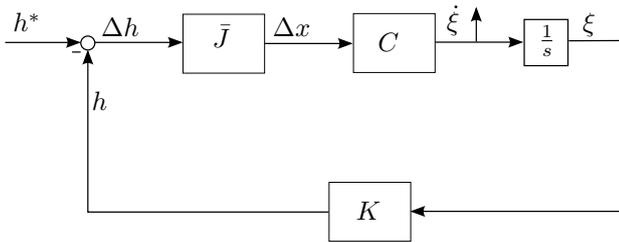


Fig. 2. Control Structure

I is the input to the control part, which is composed by \bar{J} and C and is discussed later in detail. Here the controller C is placed after \bar{J} to control the different motion components (rotation and translations) independently.

The Jacobian J of the preceding Section maps the vehicle's motion to the homography entries, but this control loop requires right the inverse transformation from \dot{h} to \dot{x} . Although

this can be obviously achieved by inverting J , it is not preferable, since J is not quadratic, which requires the calculation of a pseudo-inverse. An analytic calculation of \dot{x} in dependence of \dot{h} is the more accurate and efficient choice.

For the remainder of this paper we will consider the realistic case with $n_1 \neq 0$, $n_2 = 0$, and $n_3 \neq 0$. Obviously, not all equations of (9) are equally suitable, because no information is contained in h_{21} , h_{22} , and h_{23} . As written above h_{12} , h_{32} are also not always a good choice. With the four equally suitable equations h_{11} , h_{13} , h_{31} , and h_{33} there are the following four possibilities to calculate the vehicle's motion demand:

(I.) h_{11} , h_{13} , h_{33} :

$$\dot{v}_x = \frac{d(h_{13}n_1 \sin(\alpha) + h_{33}n_1 \cos(\alpha)) - dn_3(h_{11} \sin(\alpha) - h_{33} \sin(\alpha))}{n_3(n_1 \cos(\alpha) + n_3 \sin(\alpha))}$$

$$\dot{v}_y = \frac{-d(h_{11} \cos(\alpha) + h_{13} \sin(\alpha))}{n_1 \cos(\alpha) + n_3 \sin(\alpha)}$$

$$\dot{\alpha} = \frac{-h_{11}n_3 + h_{13}n_1}{n_1 \cos(\alpha) + n_3 \sin(\alpha)}$$

(II.) h_{11} , h_{13} , h_{31} :

$$\dot{v}_x = \frac{d(h_{31}n_3 \sin(\alpha) - h_{11}n_3 \cos(\alpha)) + dn_1(h_{13} \cos(\alpha) + h_{31} \cos(\alpha))}{n_1(n_1 \cos(\alpha) + n_3 \sin(\alpha))}$$

$$\dot{v}_y = \frac{-d(h_{11} \cos(\alpha) + h_{13} \sin(\alpha))}{n_1 \cos(\alpha) + n_3 \sin(\alpha)}$$

$$\dot{\alpha} = \frac{-h_{11}n_3 + h_{13}n_1}{n_1 \cos(\alpha) + n_3 \sin(\alpha)}$$

(III.) h_{11} , h_{31} , h_{33} :

$$\dot{v}_x = \frac{d(h_{33} \cos(\alpha) - h_{31} \sin(\alpha))}{n_3 \cos(\alpha) - n_1 \sin(\alpha)}$$

$$\dot{v}_y = \frac{d(h_{31}n_3 \sin(\alpha) - h_{11}n_3 \cos(\alpha)) + dn_1(h_{11} \sin(\alpha) - h_{33} \sin(\alpha))}{n_1(n_3 \cos(\alpha) - n_1 \sin(\alpha))}$$

$$\dot{\alpha} = \frac{-h_{31}n_3 + h_{33}n_1}{n_3 \cos(\alpha) - n_1 \sin(\alpha)}$$

(IV.) h_{13} , h_{31} , h_{33} :

$$\dot{v}_x = \frac{d(h_{33} \cos(\alpha) - h_{31} \sin(\alpha))}{n_3 \cos(\alpha) - n_1 \sin(\alpha)}$$

$$\dot{v}_y = \frac{d(h_{13}n_1 \sin(\alpha) + h_{33}n_1 \cos(\alpha)) - dn_3(h_{13} \cos(\alpha) + h_{31} \cos(\alpha))}{n_3(n_3 \cos(\alpha) - n_1 \sin(\alpha))}$$

$$\dot{\alpha} = \frac{-h_{31}n_3 + h_{33}n_1}{n_3 \cos(\alpha) - n_1 \sin(\alpha)}$$

(10)

The existence of four different calculation possibilities has two advantages. First if $n_3 \cos(\alpha) - n_1 \sin(\alpha) = 0$ the formulas with the other denominator can be used and vice versa. Second the calculation of v_x , v_y , and $\dot{\alpha}$ by different formulas and by using all four homography entries increases the robustness.

Moreover, Equation (10) shows clearly that even with an unknown distance d from the goal pose to the feature plane P the direction of the translation is computable up to scale, whereas the rotation, as expected, is completely unaffected. The mapping K of the current states x to the homography

matrix H can be deduced directly from (7) and has to be examined for the stability evaluation in the next chapter.

III. STABILITY EVALUATION

The first choice in the design of the controller is whether to establish one gain c for all three state variables or to set different gains. Since the controller C is placed in the control loop after \bar{J} , the state variables are decoupled and the gains can be individually adapted. The velocities v_x, v_y have similar characteristics, so it is sufficient to apply the same gain c_1 for their control, while an extra gain c_2 is used for the control of the orientation.

For the evaluation of the stability the results from the preceding Chapter are utilized to calculate the control inputs by only three terms of the Homography Jacobian (9) - here solution (IV.) from (10), namely h_{13} , h_{31} , and h_{33} . First the control loop is cut at \dot{x} in order to compute the system matrix A with $\dot{x} = Ax$. The non-linear function block K maps the state variables to the entries of the homography matrix. As the redundant Jacobian is reduced, K also reduces to:

$$\begin{aligned} h_{13} &= -\frac{y \cdot n_3}{d} + \sin(\alpha) \\ h_{31} &= \frac{x \cdot n_1}{d} - \sin(\alpha) \\ h_{33} &= \frac{x \cdot n_3}{d} + \cos(\alpha) \end{aligned} \quad (11)$$

Those equations are linearized and the linearized mapping \bar{K} can be written as:

$$\begin{bmatrix} \Delta h_{13} \\ \Delta h_{31} \\ \Delta h_{33} \end{bmatrix} = \begin{pmatrix} 0 & -\frac{n_3}{d} & \cos(\alpha_0) \\ \frac{n_1}{d} & 0 & -\cos(\alpha_0) \\ \frac{n_3}{d} & 0 & -\sin(\alpha_0) \end{pmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \alpha \end{bmatrix} \quad (12)$$

From this point the linearized model will be used and the Δ will not be written in front of the terms. The linearization is around the current rotational angle α_0 . With the linearized \bar{K} and the reduced Jacobi \bar{J} the state matrix $A = C \cdot \bar{J} \cdot \bar{K}$ can be calculated:

$$\dot{\xi} = \begin{pmatrix} c_1 & 0 & -c_1 \frac{d \sin(\alpha_0 - \alpha_s)}{(n_3 \cos(\alpha_s) - n_1 \sin(\alpha_s))} \\ 0 & c_1 & -c_1 \frac{d n_1 \sin(\alpha_0 - \alpha_s)}{n_3 (n_3 \cos(\alpha_s) - n_1 \sin(\alpha_s))} \\ 0 & 0 & c_2 \frac{n_3 \cos(\alpha_0) - n_1 \sin(\alpha_0)}{n_3 \cos(\alpha_s) - n_1 \sin(\alpha_s)} \end{pmatrix} \xi \quad (13)$$

The reason why two different α appear in Equation (13) is that α_s , which is the α from \bar{J} - see (10), is only an estimated or measured value of α , while α_0 is the real physical value that does not have to be known as it is implicitly used in \bar{K} . If α_0 is perfectly observable, thus $\alpha_s = \alpha_0$, the state matrix A becomes diagonal and identical to C .

Nevertheless, measuring this Cartesian value would be against the principle of an only image based feedback. For the controller design the fixed value $\alpha_s = 0$ is chosen to establish an exacter control the nearer the vehicles comes to the target orientation and to ensure a correct positioning.

The design choice is applied $A \xrightarrow{\alpha_s=0} \hat{A}$ and the characteristic

polynomial p of the system is calculated:

$$\begin{aligned} p &= \det(sI - \hat{A}) \\ &= (s - c_1)^2 \cdot (s - c_2(-\frac{n_1}{n_3} \cdot \sin(\alpha) + \cos(\alpha))) \end{aligned} \quad (14)$$

It can easily be seen that c_1 has to be chosen smaller than zero for a stable double root, whereas c_2 has to be examined more closely.

Theorem 1: While regarding the visibility constraint $d > 0$, the gain c_2 must be chosen smaller than zero, as the following expression holds:

$$-\frac{n_1}{n_3} \cdot \sin(\alpha) + \cos(\alpha) > 0 \quad (15)$$

Proof 1: Let γ be the angle between the optical center o_1 of the camera at the goal position and the normal n vector to the plane P - see Figure 3. The angle γ is limited to the following value range:

$$\gamma \in]0, \pi[\quad (16)$$

For $\gamma = 0$ and $\gamma = \pi$ the normal vector n lies in the image plane and for $\gamma > \pi$ or $\gamma < 0$ the homography condition $d > 0$ is hurt.

Due to the same restrictions the angle α , which denotes the rotation around $y_c^* = y_c$ between the current camera pose and the goal pose, must fulfill:

$$0 < \gamma - \alpha < \pi \quad (17)$$

Additionally, as n is expressed in the target camera's coordinate system, the following relation holds:

$$\begin{aligned} n_1 &= \cos\gamma \\ n_3 &= \sin\gamma \end{aligned} \quad (18)$$

Multiplying Equation (15) with n_1 and inserting (18) yields:

$$\sin(\gamma) \cdot \cos(\alpha) - \cos(\gamma) \cdot \sin(\alpha) > 0 \quad (19)$$

Using the sine subtraction theorem on (19) provides:

$$\sin(\gamma - \alpha) > 0 \quad (20)$$

The sine is greater zero in the range described in (17) and therefore Equation (15) is true for the visibility condition $d > 0$. This is a very interesting result, because it means that strong errors of n can be handled as long as the system is far enough from the boundaries of Condition (17).

IV. RESULTS

In order to verify the theory from the preceding Chapters, a test scenario, depicted in Figure 4, is set up in simulation. For an easier understanding a world coordinate system x_W, y_W is introduced and the camera pose is expressed in the world coordinate system by its $[x, y, \psi]$ tuple.

Different starting poses and goal poses were tested, but here only one scenario is presented, for which the goal pose is set to $[2, 2, 0]$, the initial pose is $[-4, -4, \frac{\pi}{4}]$, and the chosen plain equation P is $x + y - 10 = 0$. Those conditions lead to a normal vector $n = [-\sqrt{0.5}, 0, \sqrt{0.5}]$ and distance $d = \sqrt{18}$.

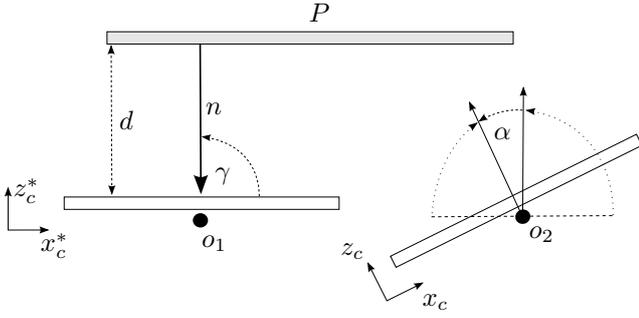


Fig. 3. Visibility Restriction

The camera model, which projects the 3D features $p_i \in P$ to 2D features m_i in the image space, uses real calibration parameters for focal length, pixel scale factors, and principal point. The focus of the simulation was laid on the sensor side and hence simple vehicle dynamics are implemented in accordance with the control interface described in the preceding Chapters. Nevertheless, actuator saturations are set up to limit v_x, v_y and $\dot{\psi}$.

In this simulation environment noise can be added to the 2D feature points as a multiplicative value or as a pixel offset range. Additionally, the normal n and distance d could be multiplied with a noise term.

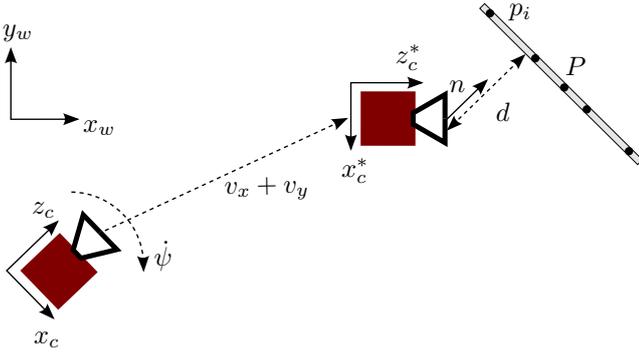


Fig. 4. Simulated Scenario

A. Robustness against Errors of n

This first experiment should test the robustness of the normal vector n , which has to be estimated once for the goal pose, against errors. To add noise the normal vector is multiplied with a random value ν_n between 0.6 and 1.4. Hereby, n_1 and n_3 have to be multiplied with different random values or otherwise the disturbance will cancel out - compare Equation (15). The goal pose is reached when the Euclidean distance from current system state to the goal state is below a set limit $l_e = 0.05$. If this condition is not met after a time limit, the positioning task is considered as failed.

The simulation with the erroneous n is run 1000 times with the result that the convergence rate is 100%. This is no surprise, since for this scenario with $\gamma = \frac{3\pi}{4}$ and $\alpha = \frac{\pi}{4}$ the distances to the boundaries according to Equation (16) and (17) are large.

If either γ or α is close to the boundaries ($< 1^\circ$) and a strong error ($\approx 2\|n\|$) is added to n , the simulation gets unstable.

B. Necessity of Measuring α

The next experiment evaluates the influence of α in the controller term \bar{J} to the system's behavior. The scenario is run first with an ideal system with no noise and $\alpha_s = \alpha_0$, which means a perfect measurement of the current angle, and then α_s is set to zero as proposed before.

With no noise both controllers have the same performance, i.e. the goal pose was reached after the same number of time steps and there is no positioning error, which is expected according to Chapter III.

Now a noise term $\nu_f \in [-1, 1]$ in pixels is added to the features points and also n is multiplied by a noise factor $\nu_n \in [0.8, 1.2]$. After that both controllers are used in respective 1000 simulation runs. The results are listed in Table I, which shows how often the task was successfully finished or not. Additionally $\Delta\bar{x}$, $\Delta\bar{y}$ and $\Delta\bar{\psi}$ are presented, which are the mean value over the respective deviances of x , y and ψ from the goal values. The mean deviances indicate that even when the task was not successful, the goal pose was almost reached.

| | Target reached | Target not reached | $\Delta\bar{x}$ | $\Delta\bar{y}$ | $\Delta\bar{\psi}$ |
|-----------------------|----------------|--------------------|-----------------|-----------------|--------------------|
| $\alpha_s = \alpha_0$ | 996 | 4 | 0.037 | 0.017 | 0.023 |
| $\alpha_s = 0$ | 996 | 4 | 0.024 | 0.024 | 0.031 |

TABLE I
COMPARISON OF $\alpha_s = 0$ AND $\alpha_s = \alpha_0$

This experiment shows that there is no clear advantage in measuring or estimating α_0 and the choice $\alpha_s = 0$ is valid.

C. Direct Control vs. Decomposition

In the final experiment the direct homography based controller C_D must compete with the a homography based control C_H , where the homography matrix is decomposed in order to receive the rotational and translational values. A decomposition method can be found in [6]. Both controllers are simulated concurrently in the same simulation environment, which means that both receive the same initial homography matrix and during the runs the feature points are equally disturbed. The winner is the controller that converges first or comes closest to the target pose, if none was able to meet the convergence limit l_e . In the presence of no noise of the homography matrix both controllers behave absolutely equally.

Therefore, different noise levels are added to the feature

| | C_D wins | C_D closest | C_H wins | C_H closest |
|---------------------------|------------|---------------|------------|---------------|
| $\nu_f \in [-0.1, 0.1]$ | 673 | 0 | 327 | 0 |
| $\nu_f \in [-0.2, 0.25]$ | 770 | 0 | 230 | 0 |
| $\nu_f \in [-0.5, 0.5]$ | 537 | 0 | 463 | 0 |
| $\nu_f \in [-0.75, 0.75]$ | 663 | 0 | 337 | 0 |
| $\nu_f \in [-1, 1]$ | 663 | 0 | 337 | 0 |
| $\nu_f \in [-2, 2]$ | 520 | 0 | 480 | 0 |

TABLE II
COMPETITION RESULTS OF C_D VS. C_H

detection and the simulation runs 1000 times for each noise level. The error level of n is kept in the range of 10% and $\alpha_s = 0$. The results are presented in Table II. The target is always reached, as a well condition feature set is used for this simulation, which means that the features distribute itself over the whole image plane. Simulations with different feature sets have shown that the direct homography control can profit more from a well distributed feature set than the decomposition does.

Moreover Table II shows that the direct homography control performs slightly better then the control with decomposition. Generally, the performance for the position control of x and y is similar, but a significant difference can be observed for the angle control.

One simulation run with $\nu_f \in [-0.75, 0.75]$ is shown in detail in Figure 5. The the above plot depicts the simulated motion in the world coordinate system from the start position to the goal and the second plot of Figure 5 displays the orientation over the simulation time, whereas a strong jitter can be seen in the signal generated by the homography decomposition control.

The here proposed method itself has demonstrated its ro-

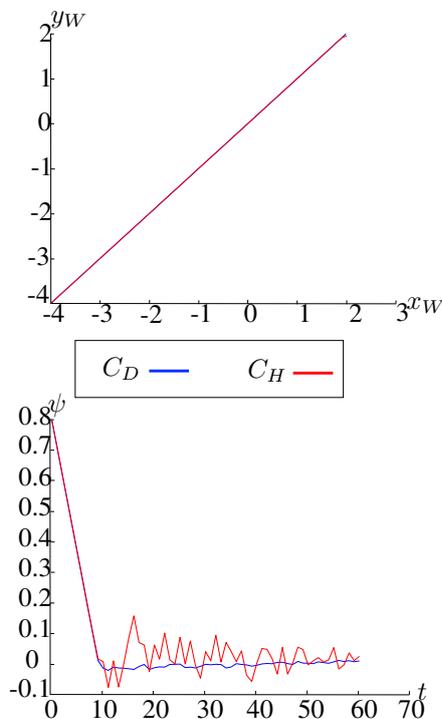


Fig. 5. State variables for a Simulated Run with $\nu_f \in [-0.75, 0.75]$

bustness against uncertainties, therefore the last bigger source of errors is the homography matrix. Consequently, a robust feature detection and homography matrix calculation must be considered when building a system as described here.

V. CONCLUSION

This paper presented a novel approach for vision based control using homographies for navigation. Due to exploiting

the dynamics of a highly maneuverable vehicle a decomposition of the homography matrix is never necessary, which leads to a computationally more efficient and robust controller. The feedback for the 3D motion control consists purely of measurements in the 2D image space. No Cartesian values have to be measured or estimated during the approach to the target, whereas only an image at the target pose and an estimation of the normal vector to the feature plane at the goal position are necessary.

If the vehicle is controlled by a standard P controller, the convergence is asymptotic. A better solution is to use a non-linear controller until the robot is located in certain vicinity to the target in order to achieve an almost exponential convergence. In the future the algorithm should demonstrate its performance on real hardware to handle for e.g. homing or platooning tasks. So far an estimation of the normal vector is necessary and therefore the goal pose must be known a priori. An online target generation would be an interesting research topic and would make the approach feasible for the navigation in unknown environments.

From a systems theory's point of view the integration of a more sophisticated vehicle dynamics model could be interesting. At this point mainly robustness against sensor errors is taken into account, whereas non-perfect behavior of the vehicle's dynamics is not considered explicitly.

REFERENCES

- [1] S. Hutchinson, G.D. Hager, and P.I. Corke. A tutorial on visual servo control. *Robotics and Automation, IEEE Transactions on*, 12(5):651–670, October 1996.
- [2] A. Cherubini, F. Spindler, and F. Chaumette. A redundancy-based approach for visual navigation with collision avoidance. In *IEEE Symp. on Computational Intelligence in Vehicles and Transportation Systems, CIVTS'11*, Paris, April 2011.
- [3] F. Chaumette and S. Hutchinson. Visual servo control. i. basic approaches. *Robotics Automation Magazine, IEEE*, 13(4):82–90, 2006.
- [4] F. Chaumette. Image moments: a general and useful set of features for visual servoing. *Robotics, IEEE Transactions on*, 20(4):713–723, Aug 2004.
- [5] C.J. Taylor and J.P. Ostrowski. Robust vision-based pose control. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 3, pages 2734–2740 vol.3, 2000.
- [6] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.
- [7] E. Malis and F. Chaumette. Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. *Robotics and Automation, IEEE Transactions on*, 18(2):176–186, Apr 2002.
- [8] S. Benhimane and E. Malis. Homography-based 2d visual tracking and servoing. *Int. J. Rob. Res.*, 26(7):661–676, July 2007.
- [9] S. Benhimane, E. Malis, P. Rives, and J.R. Azinheira. Vision-based control for car platooning using homography decomposition. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2161 – 2166, april 2005.
- [10] G. Lopez-Nicolas, N.R. Gans, S. Bhattacharya, C. Sagues, J.J. Guerrero, and S. Hutchinson. Homography-based control scheme for mobile robots with nonholonomic and field-of-view constraints. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(4):1115–1127, Aug 2010.
- [11] J. Brembeck, L. M. Ho, A. Schaub, C. Satzger, and G. Hirzinger. Romo - the robotic electric vehicle. In *22nd International Symposium on Dynamics of Vehicle on Roads and Tracks. IAVSD*, 2011.
- [12] E. Malis and M. Vargas. Deeper understanding of the homography decomposition for vision-based control. Research Report RR-6303, INRIA, 2007.