J Intell Robot Syst (2017) 87:169–186
DOI 10.1007/s10846-017-0510-0

Vision Based Output Feedback Control of Micro Aerial Vehicles in Indoor Environments

Hanoch Efraim · Shai Arogeti · Amir Shapiro · Gera Weiss

Received: 24 November 2015 / Accepted: 6 February 2017 / Published online: 1 March 2017

Abstract We present a new image based visual servoing (IBVS) approach for control of micro aerial vehicles (MAVs) in indoor environments. Specifically, we show how a MAV can be stabilized and guided using only corridor lines viewed on a front facing camera and angular velocity measurements. Since the suggested controller does not include explicit attitude feedback it does not require the use of accelerometers which are susceptible to vibrations, nor complex attitude estimation algorithms. The controller also does not require direct velocity measurements which are difficult to obtain in indoor environments. The paper presents the new method, stability analysis, simulations and experiments.

H. Efraim (✉) · S. Arogeti · A. Shapiro
Department of Mechanical Engineering, Ben Gurion University of the Negev, Beersheba, Israel
e-mail: hanoche@post.bgu.ac.il

S. Arogeti
e-mail: arogeti@bgu.ac.il

A. Shapiro
e-mail: ashapiro@bgu.ac.il

H. Efraim
Sami Shamoon College of Engineering, Department of Electrical Engineering, Beer Sheva, Israel

G. Weiss
Department of Computer Science, Ben Gurion University of the Negev, Beersheba, Israel
e-mail: geraw@cs.bgu.ac.il

Keywords MAV · Visual servoing · Indoor flight · Corridor flight

1 Introduction

The increasing popularity of micro aerial vehicles (MAVs) has resulted in many publications and industrial interest in MAV control and state estimation in recent years [5, 20, 21]. Specifically, there is much research on the use of on-board cameras as the main sensing device for MAVs guidance [9, 11, 25]. All the control architectures for MAVs that we are aware of apply some sort of attitude estimation in the feedback loop. There are two main approaches for attitude estimation of MAVs: 1) Using a set of external cameras that track markers that are placed on the MAV [12, 24]; and 2) Using on-board Micro Electro Mechanical System (MEMS) sensors, specifically, rate gyros and accelerometers. The data acquired by these sensors is fused together using filters such as the complementary filter or the Kalman filter [18, 19, 22]. The first method requires expensive and complicated infrastructure, and is not practical for most applications. The second approach relies on the use of MEMS sensors, which are inexpensive and lightweight, but produce relatively inaccurate measurements - especially accelerometers, which are sensitive to vibrations.

Beyond sensitivity to vibrations, there is a fundamental problem with the way accelerometers are used for MAV attitude estimation, as reported in [17].

Specifically, a common assumption in the design of attitude estimation algorithms is that the accelerometers measure the direction of gravity in the vehicle frame of reference. In MAVs this assumption is not valid in some cases, e.g., in near hover flight. Accelerometers provide a vector measurement of all the forces acting on the MAV minus gravity. Usually, there are three forces acting on a MAV - thrust, drag and gravity. While it is true that when the MAV is not accelerating the drag force and thrust are equal to the gravity and then accelerometers measurements provide the direction of gravity, in near hover flight the drag forces are negligible and the accelerometers only measure thrust, which does not contain information about the direction of gravity.

Another source for difficulties in MAV controllers stems from the need for linear velocity feedback required to damp the position subsystem. To the best of our knowledge, all of the publications on MAV control (e.g., [4, 13, 20, 24]) include some method for velocity estimation. In some cases the estimation depends on external infrastructure (either GPS or an external camera system), and in other cases it is based on complex vision algorithms (e.g., [3, 14]), sometimes with the aid of both vision and inertial sensors [26].

In this work we present a novel approach for the control of MAVs in an indoor environment, specifically corridors. The approach is simple to apply and does not require explicit attitude feedback, eliminating the need for accelerometers and complicated estimation algorithms. In addition, the proposed approach does not require linear velocity estimation, which is difficult to obtain accurately without external infrastructure. The controller suggested in this work is based only on visual measurements from a front facing camera (the line images depicted in Fig. 1), and angular velocity sensors. No external input is required.

As will be explained later, our suggested controller relies on a low level angular velocity controller (see Fig. 2). Such a controller can be implemented using a MEMS angular rate sensor. With current MEMS technologies, gyroscope sensors are much less sensitive to vibrations than accelometers. Their main weakness is measurement bias (as described in [23], i.e., they add a small slowly changing offset to the measurement. This bias poses a problem in attitude estimation algorithms since gyro measurements are integrated and the integration of the bias creates drift. However, for the purpose of angular velocity control, the small bias has a negligible influence on flight performance.

It should be emphasized that the novelty of this controller is not in the use of vision for guidance of a MAV, but in the way that it is used to both guide and stabilize simultaneously. For example, in [2] the authors propose to use longitudinal lines as visual cues to control the heading of a quadrotor in an indoor environment. The approach presented here also uses

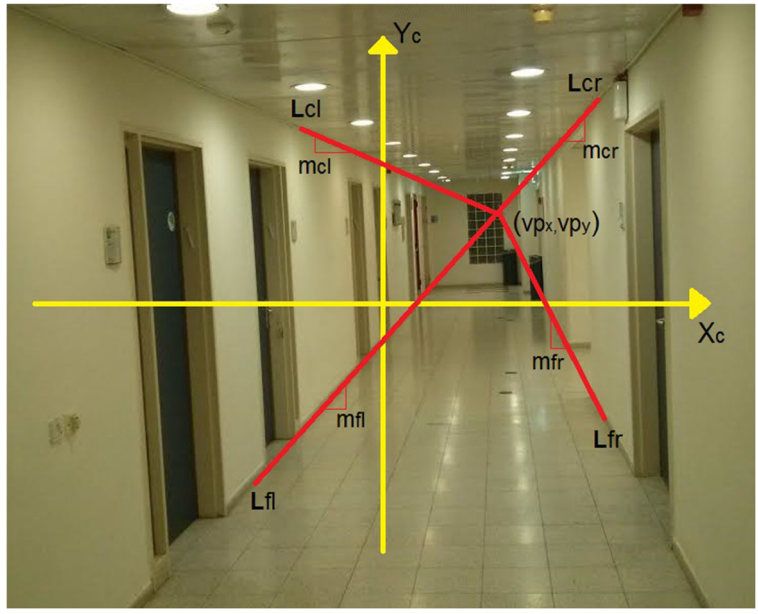

Fig. 1 The *lines marked in red* are the visual measurements of the suggested controller. L_{cl} stands for *line ceiling left*, L_{cr} stands for *line ceiling right*, L_{fl} stands for *line floor left*, L_{fr} stands for *line floor right*. The only additional measurements required are angular velocity measurements which can be obtained using a MEMS gyroscope sensor for a low level angular velocity controller

corridor longitudinal lines but in a way that enables full control of the MAV (and not only its heading), including altitude and lateral position, with reduced use of sensors. Similarly, in [1] the authors show how IBVS can be used with the linear quadratic servo approach to control the position and heading of a quadrotor above electric power lines. As in [2], the approach presented in [1] uses the images of straight power lines, but only for guidance, while the low-level control relies on attitude estimation from inertial sensors.

In addition to the possibility of using our approach as a main control strategy, the minimal usage of sensors makes it a plausible candidate for a backup control strategy that can be used in case of sensor failure or in situations, such as near hover, where the sensors are known to provide poor information (see, e.g., [17]).

This paper describes the continuation of the work in [8]. The controller presented here is a revised controller, with a more linear behavior and a larger area of convergence. Furthermore, while in [8] the concept was introduced, it was validated only by stability analysis of the linearized system model and simulations. There were no experiments included. In this work we present actual experiments performed using a real MAV. The experiments validate: (1) That the controller's area of convergence is large enough in practice; (2) That the dynamic model used (and specifically the assumptions made regarding the low level angular velocity controller) is valid; and (3) That the suggested method is feasible. We also investigate experimentally the effects of controller operating frequency on the performance and stability of the MAV. This is a significant issue since the ability to operate in low frequencies will contribute to size reduction of the MAV by enabling the use of relatively low performance, low weight processors for image processing tasks.

2 Problem Formulation

The inertial coordinate system is denoted by A and the body coordinate system which is attached to the MAV is marked as B. The position of the MAV in A is denoted as $P = [P_X\ P_Y\ P_Z]^T$. As commonly used in aviation, the $\hat{X}_B$ axis is directed "forward", the $\hat{Y}_B$ axis is directed to the right, and $\hat{Z}_B$ is directed downwards (see Fig. 3). The coordinates of some point $[X_A\ Y_A\ Z_A]^T$ in the inertial frame A expresed in the body frame B is:

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = [R^T, -R^T P] \begin{bmatrix} X_A \\ Y_A \\ Z_A \\ 1 \end{bmatrix} \tag{1}$$

where R is the rotation matrix. Using the yaw pitch roll convention, R gets the form presented in Eq. 2,

$$R = \begin{bmatrix} R_{11} & R_{21} & R_{31} \\ R_{12} & R_{22} & R_{32} \\ R_{13} & R_{23} & R_{33} \end{bmatrix} = \begin{bmatrix} c(\theta)c(\psi) & c(\psi)s(\theta)s(\phi) - c(\phi)s(\psi) & s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta) \\ c(\theta)s(\psi) & c(\phi)c(\psi) + s(\theta)s(\phi)s(\psi) & c(\phi)s(\theta)s(\psi) - c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \tag{2}$$

where ψ, ϕ and θ are the yaw roll and pitch angles respectively, c represents the cosine function and s represents the sine function.

2.1 MAV Model

This work addresses MAVs with four actuation mechanisms - τ_x, τ_y, τ_z which are torques that can be generated around the three principle axes and T which is the thrust that can be generated along the body $\hat{Z}_B$ axis,in the negative direction (upwards). The accelerations of the MAV can be modeled by:

$$\begin{bmatrix} \dot{V}_X \\ \dot{V}_Y \\ \dot{V}_Z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - R \begin{bmatrix} 0 \\ 0 \\ \frac{T}{m} \end{bmatrix} \tag{3}$$

In order not to limit the suggested controller design to a specific MAV form (quadrotor, hexacopter, helicopter, etc.), the attitude dynamics are not considered

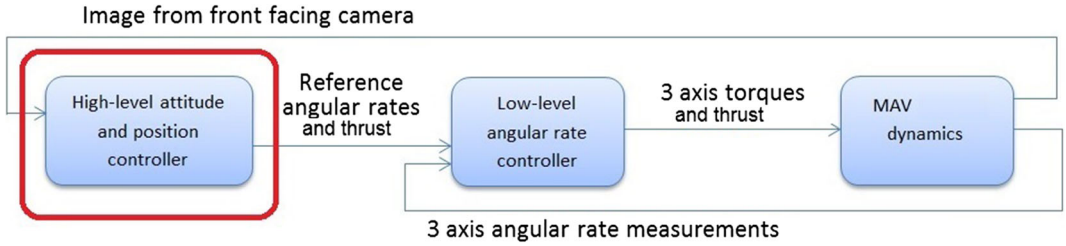

Fig. 2 System block diagram. This work focuses on the high-level controller marked in red. A platform specific lower level angular velocity controller is assumed. The only measurements needed for feedback are 3 axis angular rates and images from a front facing camera

and, instead, an attitude controller is assumed to be implemented according to the specific platform. This low level controller regulates the angular velocities around $\hat{X}_B$, $\hat{Y}_B$ and $\hat{X}_B$ (p, q and r are respectively).

The relations between body angular velocities and the Euler angles is:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4}$$

2.2 Pinhole Camera Model

In this work the MAV's front facing camera is modeled using the pinhole camera model, with the camera optic center located at the origin of B and the optical axis of the camera coinciding with $\hat{X}_B$. The image plane of the camera is defined such that $\hat{X}_c$ (image plane horizontal axis) coincides with $\hat{Y}_B$, and $\hat{Y}_c$ (image plane vertical axis) points "up" - opposite to $\hat{Z}_B$. According to the pinhole camera model, the image of some point $\begin{bmatrix} X_B & Y_B & Z_B \end{bmatrix}^T$ expressed in B is:

$$\begin{aligned} x_i &= f\frac{Y_B}{X_B} \\ y_i &= f\frac{-Z_B}{X_B} \end{aligned} \tag{5}$$

where f is the focal length and x_i and y_i are the image point coordinates.

2.3 Line Image

The controller that will be described later on uses the image lines obtained by the on-board front facing camera when flying in a corridor. In order to investigate the closed loop system stability and behavior, a model of the specific line images is needed. The inertial system A is defined such that $\hat{X}_A$ is the "forward" direction of the corridor. The line $\begin{bmatrix} t & 0 & 0 \end{bmatrix}^T$ with $t \in \mathbb{R}$ is located within the floor plane and in the middle of the corridor. Since the lines of interest are parallel longitudinal lines with known parameters in A, it is convenient to express them in B and then

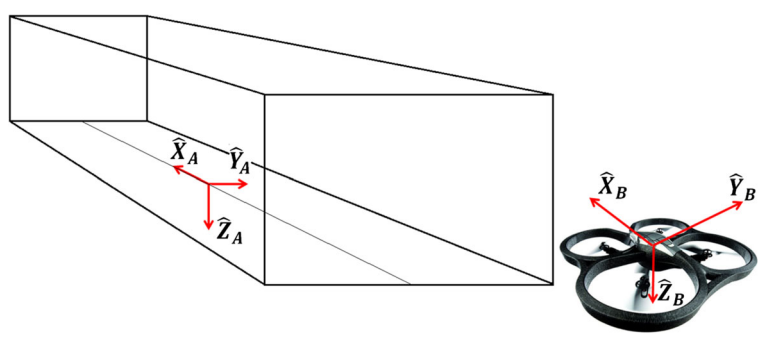

Fig. 3 Inertial and body frames of reference. As commonly used in aviation, the $\hat{X}_B$ axis is directed "forward", the $\hat{Y}_B$ axis is directed to the right, and $\hat{Z}_B$ is directed downwards. The inertial frame A is defined such that $\hat{X}_A$ is the "forward" direction of the corridor. The line $[t, 0, 0]^T$ with $t \in \mathbb{R}$ is located within the floor plane and in the middle of the corridor

calculate the image according to the pinhole camera model (5). These lines are created by the intersection of the side walls and the ceiling - $L_c = \begin{bmatrix} t & a_y & a_z \end{bmatrix}^T$ - or by the intersection of the side walls and the floor - $L_f = \begin{bmatrix} t & a_y & 0 \end{bmatrix}^T$. Here, a_z is the corridor height and a_y is half the corridor width (since the $\hat{X}_A$ axis is located within the floor plane). The lower line, L_f, expressed in B, is given by:

$$L_l = \begin{bmatrix} x_B(t) \\ y_B(t) \\ z_B(t) \end{bmatrix} = [R^T, -R^T P] \begin{bmatrix} t \\ a_y \\ 0 \\ 1 \end{bmatrix} \tag{6}$$

and the image line equations are:

$$x_i = f \frac{R_{21}t + R_{22}a_y - F_2}{R_{11}t + R_{12}a_y - F_1}$$

$$y_i = -f \frac{R_{31}t + R_{32}a_y - F_3}{R_{11}t + R_{12}a_y - F_1}$$

where

$$F_1 = R_{11}P_X + R_{12}P_Y + R_{13}P_Z$$
$$F_2 = R_{21}P_X + R_{22}P_Y + R_{23}P_Z$$
$$F_3 = R_{31}P_X + R_{32}P_Y + R_{33}P_Z$$

The slope of the image line (in the image plane) is given by:

$$m_f = \frac{\delta y_i}{\delta x_i} = \frac{\delta y_i / \delta t}{\delta x_i / \delta t}$$

$$m_f(a_y, R, P) = \frac{F_3R_{11} - F_1R_{31} + a_yR_{12}R_{31} - a_yR_{11}R_{32}}{-F_2R_{11} + F_1R_{21} - a_yR_{12}R_{21} + a_yR_{11}R_{22}} \tag{7}$$

Using the image point of the point $L_f(t = 0)$ and the slope, the equation of the image line itself can be derived:

$$y_i = f \frac{(F_3R_{21} - F_2R_{31} + a_yR_{22}R_{31} - a_yR_{21}R_{32} - F_3R_{11}x_i + F_1R_{31}x_i - a_yR_{12}R_{31}x_i + a_yR_{11}R_{32}x_i)}{(F_2R_{11} - F_1R_{21} + a_yR_{12}R_{21} - a_yR_{11}R_{22})}$$

In the same manner it is possible to calculate the higher line (L_c) image slope (8):

$$m_c(a_y, a_z, R, P) = \frac{F_3R_{11} - F_1R_{31} + a_yR_{12}R_{31} + a_zR_{13}R_{31} - a_yR_{11}R_{32} - a_zR_{11}R_{33}}{-F_2R_{11} + F_1R_{21} - a_yR_{12}R_{21} + a_yR_{11}R_{22} - a_zR_{11}R_{23}} \tag{8}$$

The image of parallel lines intersects at what is called the "vanishing point". Calculating the intersection of the images of any two lines parallel to $\hat{X}_A$ results in:

$$vp_x = \sin(\phi)\tan(\theta) - \cos(\phi)\sec(\theta)\tan(\psi)$$

$$vp_y = -\cos(\phi)\tan(\theta) - \sec(\theta)\sin(\phi)\tan(\psi) \tag{9}$$

The point (vp_x, vp_y) in the image plane, is the intersection of all four lines L_{fl}, L_{fr}, L_{cl} and L_{cr}.

3 Control

The suggested controller stabilizes the MAV at the center of the corridor facing forward. It regulates roll, pitch and yaw angles (ϕ, θ, ψ), vertical position (P_Z) and lateral position (P_Y). The system block diagram is presented in Fig. 2. Usually, in the standard cascaded controller for position and attitude of a MAV, some sort of measurement or estimation of the position and attitude is required (see Fig. 4). The controller presented here uses a similar structure; however, since there are no available position and attitude measurements, different quantities which are based on the available visual cues were selected. These quantities were selected such that they approximate the actual position and attitude of the MAV in a relatively large flight envelope. The six visual cues needed are extracted from the four line images created by the intersection of the ceiling, the walls and the floor: the slopes of the four lines, and the vanishing point (depicted in Fig. 1). Instead of lateral position, P_y, the controller regulates the quantity $\tilde{P}_y$ which will be defined as $\tilde{P}_y \triangleq \text{atan}(m_{cl}) + \text{atan}(m_{cr}) - \text{atan}(m_{fl}) - \text{atan}(m_{fr})$. In a similar manner, instead of vertical position, P_z, the controller regulates the

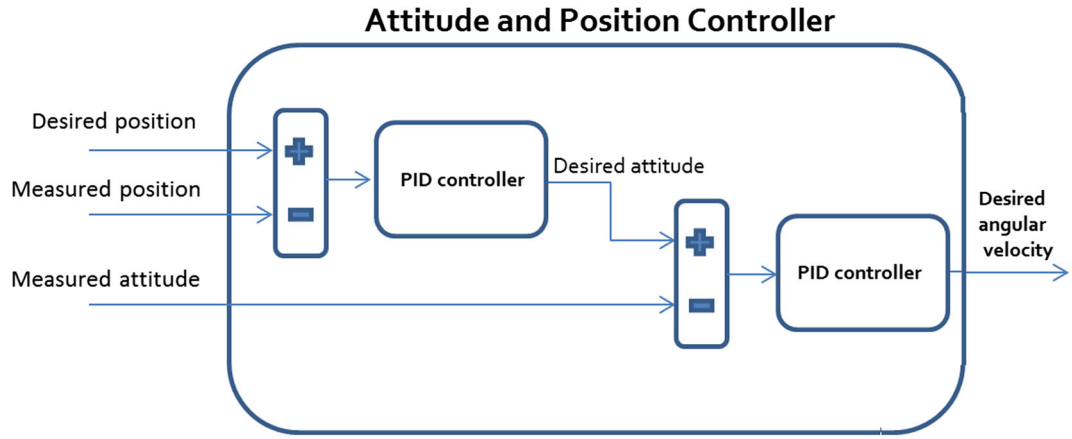

Fig. 4 Standard structure of an attitude and position controller. The cascaded structure generates a desired attitude as a function of the error in position. In order to calculate these errors, a measurement of attitude and position is required. Then, these errors are regulated using some sort of a controller, usually a PID controller

quantity $\tilde{P}_z \triangleq (\text{atan}(m_{fl}) - \text{atan}(m_{fr}) + \text{atan}(m_{cl}) - \text{atan}(m_{cr})$; instead of roll angle, ϕ, the controller regulates $\tilde{\phi} \triangleq \text{atan}(m_{fl}) + \text{atan}(m_{fr}) + \text{atan}(m_{cl}) + \text{atan}(m_{cr})$; instead of the pitch angle the controller regulates vp_y and instead of the yaw angle the controller regulates vp_x. There are other possible quantities that will stabilize the system, some of them were presented in our previous work [8]. These specific quantities were selected such that the controller stabilizes each degree of freedom in a relatively independent manner, while the effect on the other state variables is reduced. For example, T_r, which can be considered the vertical controller, is not sensitive to the yaw angle and the position along the lateral axis (Y_A). This quality is important during the take off maneuver, where initial pitch and roll angles are close to zero, but the yaw angle and the position P_Y can vary. A major drawback to the quantities selected in [8] (and the reason they are not suitable for implementation in practice) is their nonlinear nature. As the MAV position deviates from the equilibrium point these quantities may reach very large values in some configurations. This is the result of their being directly proportional to the slopes of the lines in the image or to the intersection of the lines with the image horizontal and vertical axes. This is not the case with the quantities suggested here - $\tilde{P}_y$, $\tilde{P}_z$, $\tilde{\phi}$, vp_x and vp_y are bounded, since they are proportional to the angles of the lines relative to image axes and to the vanishing point of the lines. The quantities suggested here increase the area of convergence (relative to the quantities selected in [8]) and produce a practical controller with smooth and predictable system behavior.

The signals v_d and h_d are used to damp the system (the D element in a PID controller). They are generated by filtering $\tilde{P}_z$ and $\tilde{P}_y$, respectively, through a High Pass Filter (HPF).

$$\begin{aligned} \dot{v}_{ds} &= f_a\, v_{ds} + f_b\, \tilde{P}_z \\ v_d &= f_c\, v_{ds} + f_d\, \tilde{P}_z \\ \dot{h}_{ds} &= f_a\, h_{ds} + f_b\, \tilde{P}_y \\ h_d &= f_c\, h_{ds} + f_d\, \tilde{P}_y \end{aligned} \tag{10}$$

Here f_a, f_b, f_c and f_d are the filter coefficients, and v_{ds} and h_{ds} are the state variables.

The suggested control law is as follows:

$$\begin{aligned} p_r &= k_p\tilde{\phi} + k_\phi\, \tilde{P}_y + k_{\phi d} h_d \\ q_r &= k_\theta (vp_y) \\ r_r &= k_\psi (vp_x) \\ T_r &= k_z\, \tilde{P}_z + k_{zd} v_d + mg \end{aligned} \tag{11}$$

where p_r, q_r and r_r are the desired angular velocities (the inputs to the low level angular velocity controller) and $k_p, k_\phi, k_{\phi d}, k_\theta, k_\psi, k_z$ and k_{zd} are the control gains.

3.1 Closed Loop System Model

Most MAVs make an impression due to their agility, which is the result of high power energy sources combined with relatively low mass. This quality makes it

reasonable to assume that the lower level attitude controller can be designed to be very fast, with very small tracking delays that are small enough to be neglected. This means that, practically, $p{=}p_r$, $q{=}q_r$, $r{=}r_r$ and $T{=}T_r$.

Under this approximation, the complete closed loop system (including the filtered signals) is described by

$$\begin{aligned}
\dot{P}_Y &= V_Y \\
\dot{V}_Y &= -R_{32}\frac{T_r}{m} \\
\dot{P}_Z &= V_Z \\
\dot{V}_Z &= g - R_{33}\frac{T_r}{m} \\
\dot{\phi} &= p_r + \sin(\phi)\tan(\theta)q_r + \cos(\phi)\tan(\theta)r_r \\
\dot{\theta} &= \cos(\phi)q_r - \sin(\phi)r_r \\
\dot{\psi} &= \frac{\sin(\phi)}{\cos(\theta)}q_r + \frac{\cos(\phi)}{\cos(\theta)}r_r \\
\dot{v}_{ds} &= f_a\, v_{ds} + f_b\, \tilde{P}_z \\
\dot{h}_{ds} &= f_a\, h_{ds} + f_b\, \tilde{P}_y
\end{aligned} \tag{12}$$

Here, a_y and a_z are positive numbers (a_z is the corridor height and a_y is half the corridor width). The state vector of the system is defined as $\zeta = (P_Y\ V_Y\ P_Z\ V_Z\ \phi\ \theta\ \psi\ m_{frs}\ m_{fls}\ m_{cls})^T$. This system has an equilibrium point at $P_z = -a_z/2$ with all other state variables equal to zero. Using the linearization of the system (13) and Lyapunov stability theory [15], it will be shown that with the right choice of control gains the system is locally stable.

$$\dot{\zeta} = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{16a_y(k_z+f_dk_{zd})}{m(4a_y^2+a_z^2)} & 0 & 0 & 0 & 0 & -\frac{f_ck_{zd}}{m} & 0 \\
-\frac{8a_z(k_\phi+f_dk_{\phi d})}{4ay^2+az^2} & 0 & 0 & 0 & 4k_p & 0 & 0 & 0 & f_ck_{\phi d} \\
0 & 0 & 0 & 0 & 0 & -k_\theta & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -k_\psi & 0 & 0 \\
0 & 0 & -\frac{16a_yf_b}{4a_y^2+a_z^2} & 0 & 0 & 0 & 0 & f_a & 0 \\
-\frac{8a_yf_b}{4a_y^2+a_z^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & f_a
\end{bmatrix} \zeta \tag{13}$$

3.2 Guidance

Guidance of the MAV is very simple, but has some limitations. By adding an offset value to each of the four regulated quantities $\tilde{P}_y$, $\tilde{P}_z$, vp_x and vp_y in the control law (11) , the state of the MAV can be manipulated. The ability to manipulate the attitude angles in order to achieve forward flight is most important. According to the notation in this work, when the MAV camera is facing forward, acceleration along the corridor (X_A direction) is manipulated by the pitch angle (θ). As mentioned before, and can be seen in Eq. 13, around the equilibrium point, the closed loop dynamics of the pitch angle is approximately that of an autonomous first order system:

$$\dot{\theta} = -k_\theta\theta \tag{14}$$

Changing q_r by adding an offset value, will make it possible for the pitch angle to be manipulated. The revised controller is

$$q_r = k_\theta(vp_y + \theta_{OS}) \tag{15}$$

and the dynamics around the equilibrium point would be

$$\dot{\theta} = -k_\theta(\theta - \theta_{OS}) \tag{16}$$

This is a stable transfer function in which the pitch tracks θ_{OS}. It should be emphasized that this method of controlling pitch directly also does not use any kind of accelerometer measurements.

In the same manner, adding an offset to $\tilde{P}_z$ will cause the MAV to change its altitude, and adding an offset to $\tilde{P}_y$ will cause the MAV to change its lateral

position. The main limitation is that since the dimensions of the corridor are unknown, it is not possible to set an exact numeric position. This guidance method may be especially useful combined with a remote human operator, who will guide the MAV through the structure by manipulating these offsets.

4 Simulation

In order to calculate initial control gains and evaluate the performance and stability of the closed loop system, a Simulink model was constructed according to Eq. 12. Using this model, a Matlab based visualization of the take-off process was developed (Fig. 10).

4.1 Control Gains

It is important to note that the parameters in Eq. 12 are all design parameters, except for the corridor width and height (a_y and a_z). This means that the control gains presented here are adequate for any MAV with a fast enough low level angular velocity controller. Also, as can be seen in Eq. 13, the dynamics of the MAV are not dependent on the focal length f of the camera. This is a very important feature, since it means that any camera with a wide enough field of view will be suitable, and there is no need to estimate the exact focal length.

From Eq. 13, it can be observed that in the linear closed loop system both ψ and θ are independent of the other state variables. Any positive value for the control gains k_θ and k_ψ will result in convergence of these state variables.

The HPF is used to achieve an approximation of the derivation. It is designed so its pole frequency is high enough compared to the position subsystem bandwidth. We chose the filter coefficients to be: $f_a = -10$, $f_b = 8$, $f_c = -12.5$, and $f_d = 10$.

All of the control gains were set using the graphical interface of the "compensator design" tool in the Simulink control toolbox. The chosen control gains are: $k_p = -1.1$, $k_\phi = 0.35$, $k_{\phi d}$=0.4, $k_z = -0.8$, $k_{zd} = -0.9$, $k_\theta = 3$ and $k_\psi = 6$. Using these gains, the resulting closed loop poles in a standard corridor 2[m] wide and 2.5[m] high ($a_y = 1, a_z = 2.5$) are all stable:
$-11, -8.3, -6, -3, -1.43, -0.96 \pm 1.8i, -0.83 \pm 0.9i$. Further stability analysis will follow in the next section.

As mentioned before, the corridor width and height are the only unknown parameters. Figure 5 depicts the value of the "slowest" converging pole, the pole with the largest real value, as a function of corridor

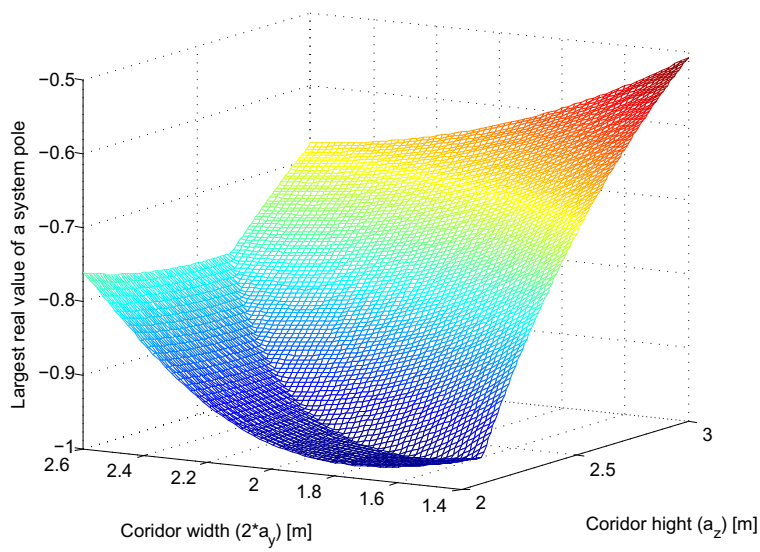

Fig. 5 The real value of the slowest converging pole as a function of corridor width and height when using the presented control gains. It is clear that for any reasonably sized corridor the closed loop system is stable

size. From this figure it is clear that the suggested controller is locally stable for any reasonably sized corridor (width between 1.4 to 2.6 meters, height between 2 to 3 meters); therefore there is no need for prior knowledge of the exact corridor size.

4.2 Stability Margins and Natural Response

The design goal was to achieve fast convergence, with no less than 40 degrees of phase margin and no less than 8 dB gain margin for each feedback loop. Figure 6 shows the Nichols plots for the vertical open loops at k_z and k_{zd}, and Fig. 7 shows the Nichols plots for the horizontal open loops at k_ϕ, $k_{\phi d}$ and k_p. All stability margins are satisfactory. The time response for a non zero initial condition of the nonlinear closed loop system is presented in Fig. 8.

4.3 Area of Convergence

Before takeoff, while the MAV is on the floor, both pitch and roll angles are very close to zero. It is reasonable to assume that due to the relatively small size of the corridor the roll and pitch angles will remain small during the flight (especially the roll angle). However, the altitude before takeoff is far from the equilibrium point, and so may be the case for the yaw angle and the lateral position. Figures 9 and 10 shows that the choice of regulated quantities presented here results in a very large area of convergence. The initial position of the MAV in Fig. 9 is $P_y = -0.5[m]$ and the initial yaw angle is $\psi = 35[deg]$.

5 Experiments

Two experimental systems were set up, each with different goals: 1) An experimental system that used emulated images, which purpose was to show the flight performance of the suggested method in case ideal image acquisition and processing was available, and 2) An experimental system that used real images, which goal was to demonstrate the feasibility of the method and to explore the technical difficulties that may be encountered when using computer vision to control MAVs indoors.

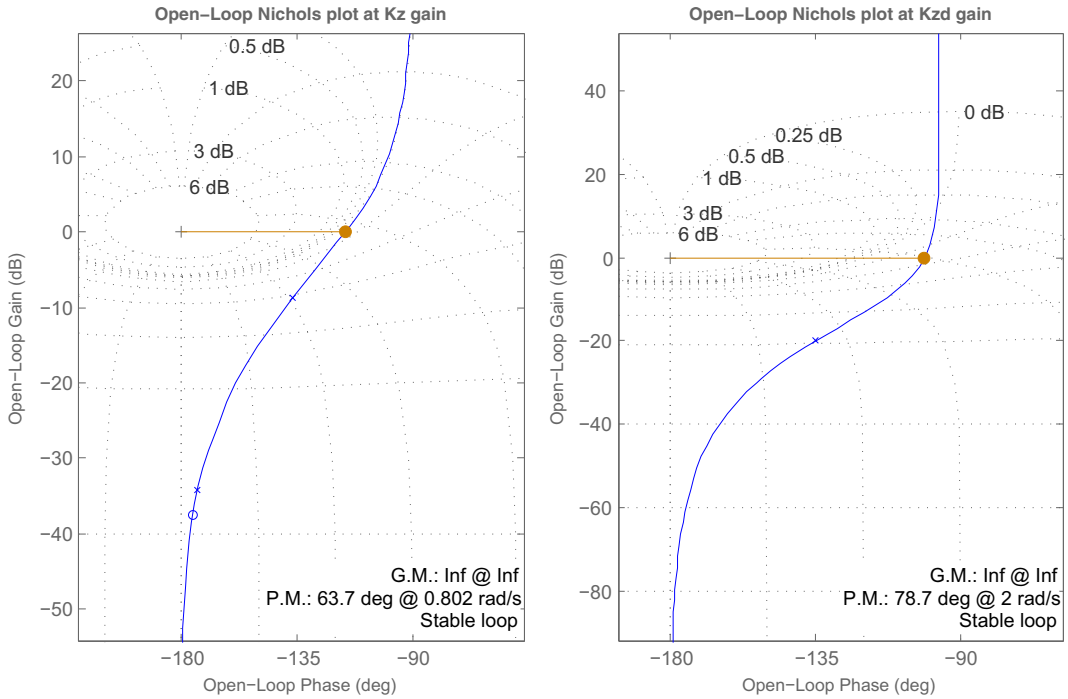

Fig. 6 Nichols plots for the vertical feedback loops as generated by the Simulink control design tool. The stability margins are satisfactory. The curves are smooth, and there is no risk that a small change in phase or amplitude will significantly change the margins

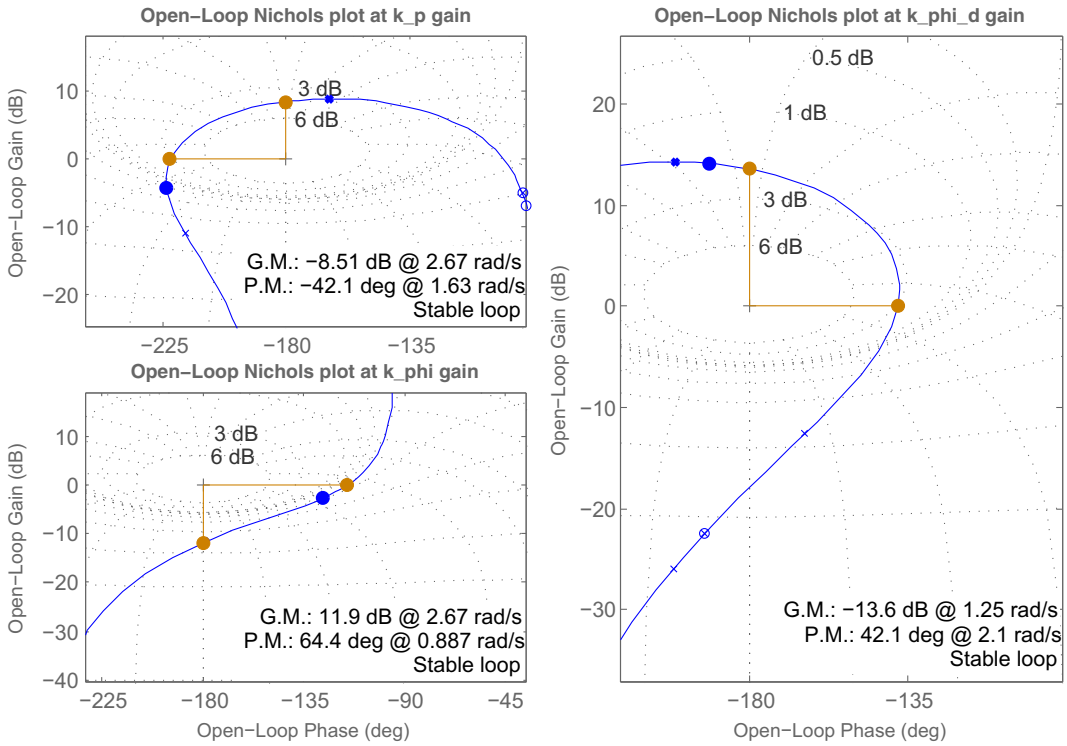

Fig. 7 Nichols plots for the horizontal feedback loops as generated by the Simulink control design tool. The stability margins are satisfactory. The curves are smooth, and there is no risk that a small change in phase or amplitude will significantly change the margins

In the first experimental system, the image that would have been captured on the front facing camera was emulated using an external set of cameras that calculated the MAV pose. Using simple transformations (see Eqs. 7, 8 and 9) the corridor image line parameters were calculated and wirelessly sent to the MAV. In the second experimental system, the front facing camera of the MAV was used to take images of the actual corridor. Since the MAV used is equipped with a relatively weak processor, and since the contribution of this work is not in image processing, the corridor lines were marked using black adhesive tape. An image processing algorithm was developed to identify the lines and the line parameters were used as input to the controller.

Due to its durability and availability, the MAV chosen for the experiments is the commercially available AR.drone 1 from parrot [6]. Figure 11 shows the AR.drone during flight experiments.

It is a widely available, low cost MAV that we could program easily without additional hardware. The drone is equipped with an ARM9 processor running a Linux operating system, a range of sensors including accelerometers, gyros, altimeter, and two cameras - one facing forward and another directed down. The

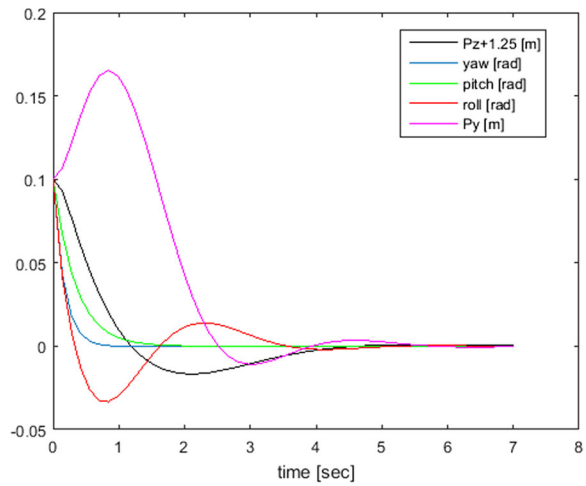

Fig. 8 Simulated time response of the closed loop non linear system around the equilibrium point

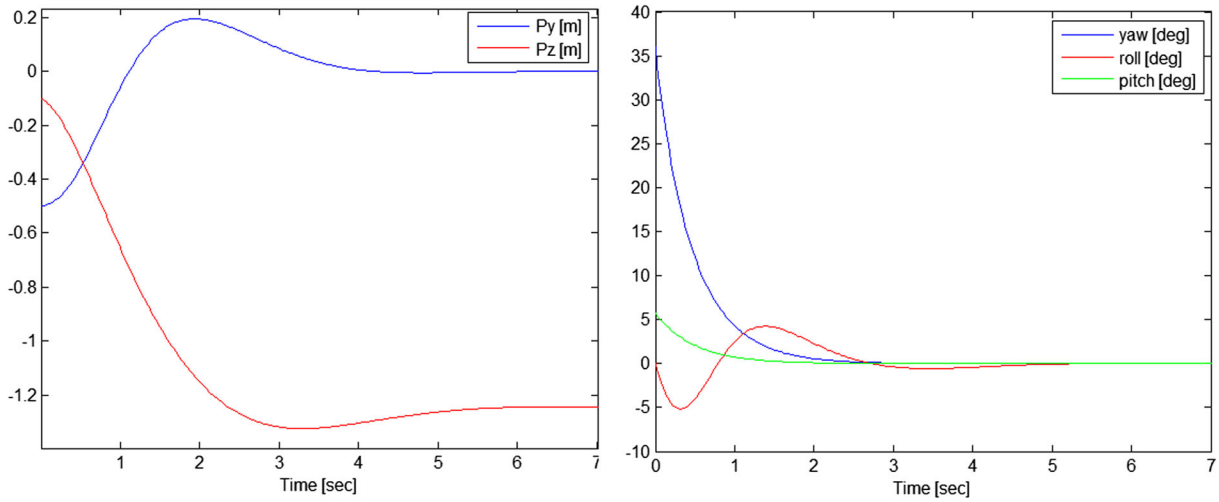

Fig. 9 Simulation of a take off and hover maneuver. On the left - lateral and vertical position. On the right - roll, pitch and yaw angles. The closed loop system converges even though the initial condition is far from the equilibrium point

default firmware includes a wireless interface that supports the manipulation of the roll angle, the pitch angle, the yaw rate, and the vertical velocity. Since the controller presented here requires a low level angular rate controller for all three axes, it was not possible to use the original firmware for this experiment and replacement firmware was developed. Details on how it can be done are available at https://blog.perquin.com/blog/ar-drone-program-elf-replacement/. The new firmware includes a low level angular rate controller (see Fig. 2) implemented at 200 Hz. The controller is composed of three separate PI controllers, one for each axis. The controller uses angular rate measurements from the on-board three axis MEMS gyro sensor to calculate angular velocity error and updates the electric motor control signals using the I2C communication protocol.

5.1 Experiments Using Simulated Front Camera View

The position and attitude of the MAV was determined using an external camera system and markers placed on the MAV. Using this information, the view from the front facing camera was emulated, according to the pinhole camera model, as described in Eqs. 7, 8 and 9. Then, the six parameters $m_{fl}, m_{fr}, m_{cl}, m_{cr}, vp_x$ and vp_y were sent wirelessly to the MAV. The emulated corridor in the following experiments was set to be 2 meters wide and 2 meters high. The external camera system used was the Optitrack system (see

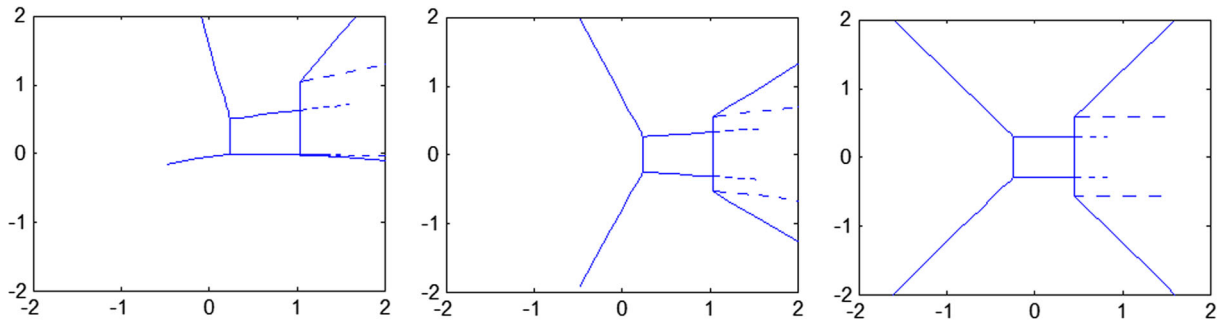

Fig. 10 Simulated view of the front facing camera. The scenery includes a 2[m] wide and 2.5 [m] high corridor with a 90 degrees right turn at it's end. On the *left* - before take off, the MAV is on the ground, located 0.5 [m] from the left wall, facing 20 degrees to the left. In the *middle* - end of take off phase. On the *right* - the MAV is in steady state at the center of the corridor

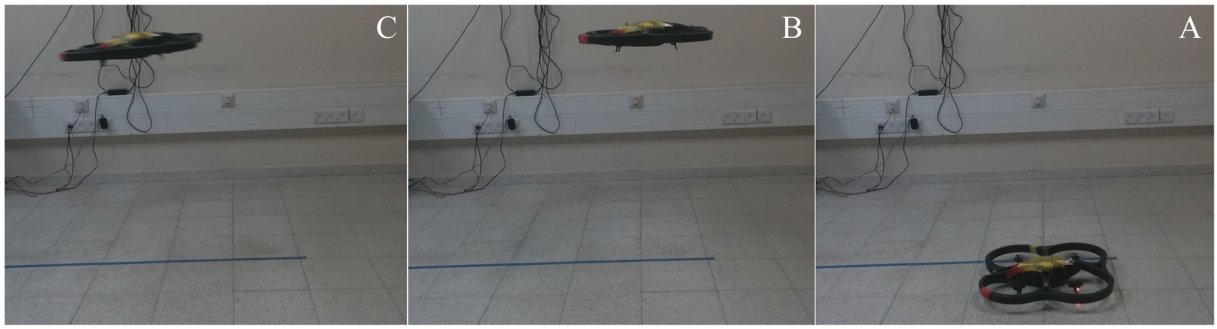

Fig. 11 The AR.drone during forward flight experiment in Ben Gurion University robotics lab. The Optitrack camera system is used to simulate a virtual corridor and data from a simulated image is sent using a wireless interface to the MAV. *Lines* marking the edges of the virtual corridor are visible on the floor (These lines are only used to visually inspect the MAV performance during experiments). From *right* to *left*: A - MAV on ground, B - MAV hovering after take off, C - Forward flight

http://www.naturalpoint.com/optitrack/) working at one hundred frames per second.

Area of Convergence - Take Off and Hover Maneuver To investigate the controller's area of convergence we performed a take off and hover maneuver, starting with the MAV on the ground. The take off maneuver exposes the controller to relatively extreme initial conditions since the yaw angle, P_z and P_y are all simultaneously far from the equilibrium point. In this case, with a nonlinear system, it is useful to use limiters which limit maximum control signal values. However, we found that the controller area of convergence is large enough that limiters were not required. The results are depicted in Fig. 12. The plot on the left depicts the position and the plot on the right depicts the roll and yaw angles. The initial yaw was $\psi(0) = 19[deg]$ and the initial lateral position was $P_y(0) = 0.65[m]$, which is about the maximum allocation possible in a two meters wide corridor for a standard quadcopter MAV. As demonstrated in this experiment, the convergence is fast and the system is stable at the equilibrium point.

Forward a Flight Figure 13 shows the position and pitch angle of the MAV during a forward flight maneuver. In this experiment, after take-off, θ_{OS} was set to $2.5[deg]$ (see Eq. 15) for $2.7[sec]$. As expected,

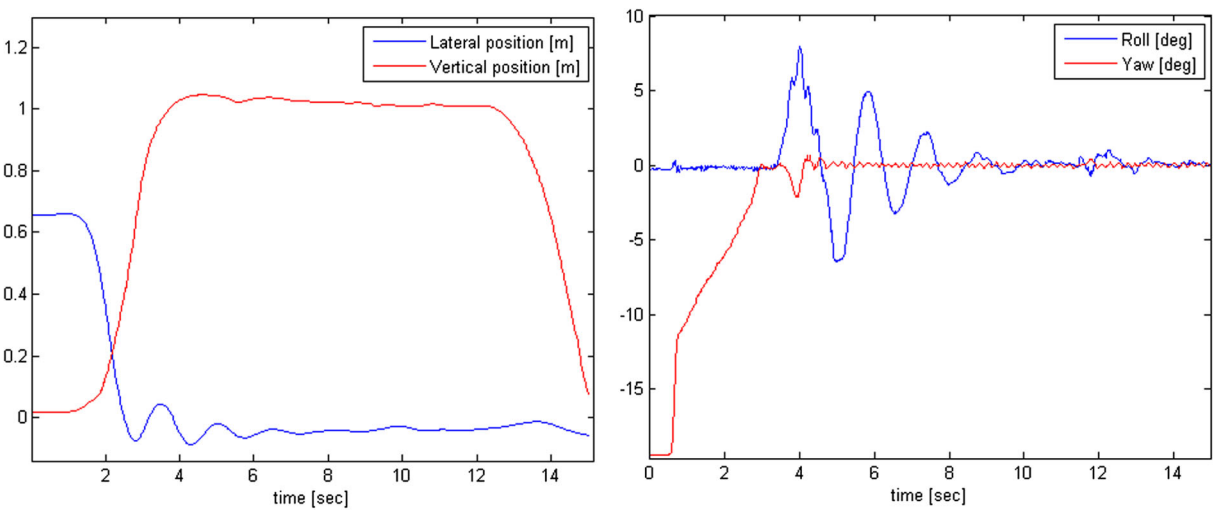

Fig. 12 Take off and hover experiment. On the *left* - vertical and lateral position of the MAV as measured by the Optitrack system during the experiment. On the *right* - roll and yaw angles. The initial yaw, $\psi(0) = 19[deg]$ and the initial lateral position, $P_y(0) = 0.65[m]$

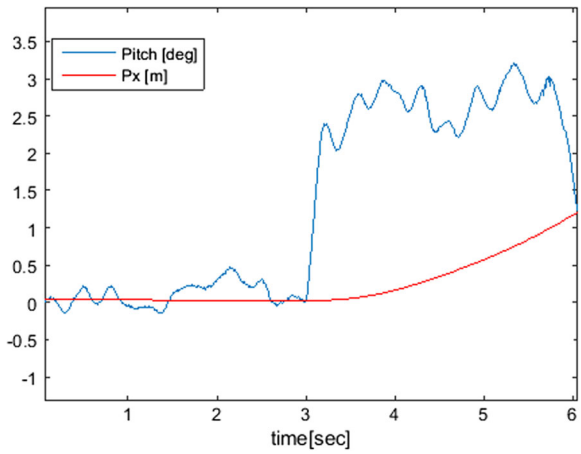

Fig. 13 P_x and pitch angle θ as measured by the Optitrack system during forward flight experiment

the MAV accelerates forward along the P_x axis and forward flight is achieved.

Controller Frequency of Operation The frequency at which the controller calculates control signals is a critical parameter. The controller bandwidth needs to be significantly higher than the process being regulated. This is commonly a weak point in vision based controllers, since image processing is a computationally demanding task that limits the controller bandwidth. The purpose of this experiment is to investigate the performance of the controller at different operating frequencies and to validate that it produces good results at lower operating rates that allow enough time for the image processing to take place.

The image processing required for the suggested controller is line detection. In [10] the authors suggest a Hough transform based method for extracting lines from images in real time. It is shown that the proposed approach can achieve up to 200 frames per second for images with 512×512 pixel resolution using an AMD Athlon processor running at 2.21 GHz. Even though the experimental results presented in previous sections used a 100Hz controller, it is interesting to investigate lower controller frequencies that will allow the use of a lower performance processor. This can contribute to size reduction of the MAV, since lower performance processors are usually smaller and require less power to operate.

In this experiment a take off and hover maneuver was performed at different controller frequencies. The MAV was initially placed on the ground with a significant lateral offset of 0.45 meters. Figure 14 depicts the results. The results at lower controller frequencies are almost identical to the results at 100 Hz, with no significant decrease in performance. It is common to assume that MAVs require relatively high controller rates - usually higher than 50 Hz - and in some cases as high as 500–1000 Hz with variable pitch propellers. In [7] the authors report an improvement in hovering performance when increaing the controller update rate from 200 to 500 Hz. We explain the surprisingly

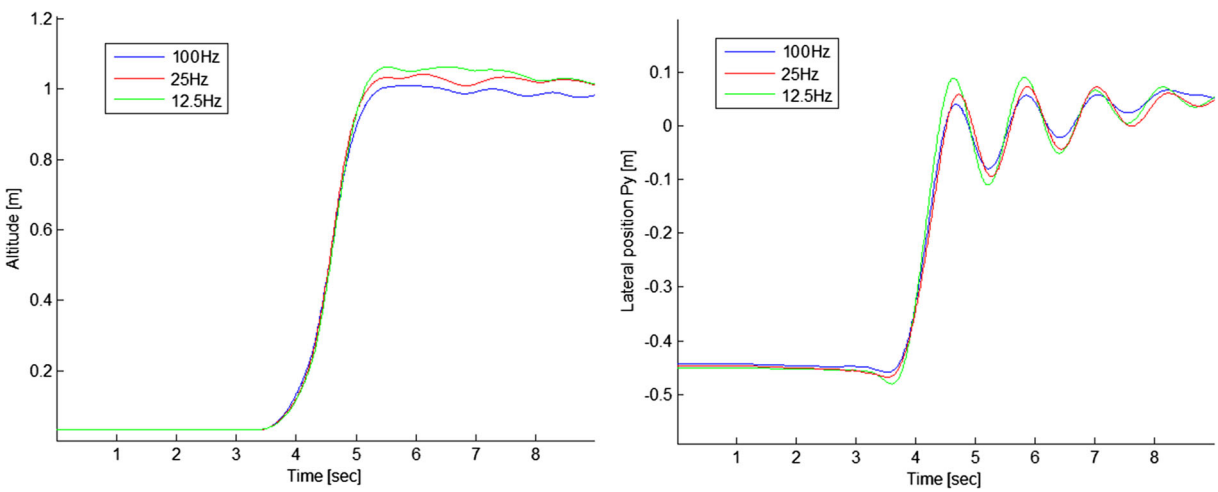

Fig. 14 The MAV lateral position, P_y, and the altitude, $-P_z$, as measured by the Optitrack system. This experiment presents a comparison between take off and hover maneuvers performed at different controller frequencies

good performance at lower controller rates in this experiment by the fact that the lower level angular rate controller operates independently at 200 Hz, and maintains a damped attitude subsystem, while the position subsystem, which is not as agile as the attitude subsystem, does not require high control rate.

We conclude that it is possible to perform image processing at the required rate with large margins for other vision related or otherwise required algorithms.

Manipulating MAV Altitude In this experiment altitude manipulation is demonstrated. As described in Section 3.2, it is possible to manipulate the MAV position in the corridor by revising the control law and adding offsets. By adding an offset to $\tilde{P}_z$, the altitude of the MAV is manipulated. The control law (see Eq. 11) for the throttle becomes:

$$T_r = k_z(\tilde{P}_z + P_{zOS}) + k_{zd} v_d + mg \tag{17}$$

Here P_{zOS} is the offset added. In order to calculate the offset required for a specific altitude we consider the following - around the equilibrium point, where $\phi = 0, \psi = 0, \theta = 0$ and $P_y = 0$ in a corridor 2 meters wide and 2 meters high, we get that $\tilde{P}_z = 4(1 + \text{atan}(P_z))$. For a desired vertical position P_{zd} the required offset is $P_{zOS} = -4(1 + \text{atan}(P_{zd}))$.

Figure 15 presents the experimental result. At $t = 10\,[sec]$ the offset P_{zOS} is updated from zero to

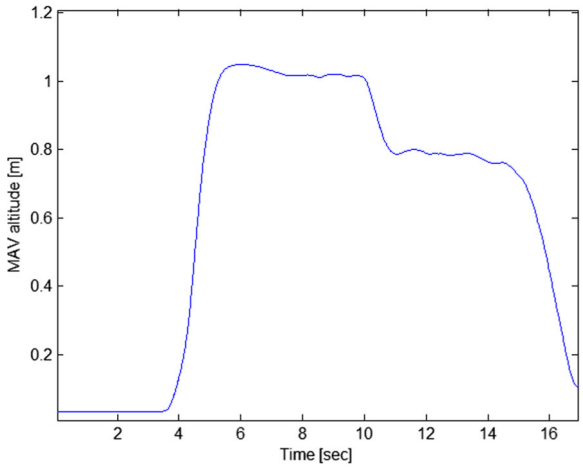

Fig. 15 MAV altitude during altitude manipulation experiment. At $t = 10\,[sec]$ the offset P_{zOS} is updated and the MAV altitude changes accordingly. At $t = 14[sec]$ the landing maneuver is initiated and the experiment is terminated

$P_{zOS} = -4(1 + \text{atan}(-0.8)) = 0.8$. The MAV deviates from the center of the corridor to a lower altitude $P_z = -0.8[m]$ (altitude of 0.8[m] above the floor). The transition is fast, smooth, and the MAV is stable at the new altitude. At $t = 14[sec]$ the landing maneuver is initiated and the experiment is terminated.

5.2 Experiments Using Real Images

The purpose of this experiment is to demonstrate the feasibility of the method and to explore the technical difficulties that may be encountered when using computer vision to control a MAV indoors. In this experiment the front facing camera of the AR.drone MAV was used to capture images of the corridor. Each image contains 640X480 pixels. It is possible to read colored images in YUV 4:2:0 format, but we chose to only use the Y component and treat it as a grayscale image. The images were captured and processed at approximately $15[Hz]$.

Image Processing The task of extracting lines from an image is usually composed of edge detection followed by a Hough transform. Both of these processes are relatively computationally expensive and require significant time. We found that it is impractical to implement these methods on the Ar.drone processor in real time. As mentioned before, since the contribution of this work is not in image processing, we chose to mark the corridor lines with a dark adhesive tape and developed an algorithm that simultaneously detects pixels on the line edges, and the lines themselves. The algorithm only examines the pixels on a rectangle located at the edges of the image. If the detection fails, the algorithm is repeated, checking pixels along a smaller rectangle. This significantly shortens the duration of the process. A circle of 24 pixels surrounding each candidate pixel is examined, as depicted in Fig. 16. If the pixels on the circle can be divided into 12 consecutive pixels whose value is below a certain threshold, followed by 12 pixels whose value is above another threshold, the candidate pixel is detected as a line edge. Once a pixel is detected as a line edge, the pixels surrounding it along a larger circle (with a radius of about 8 pixels) are examined. If another pixel is detected, the algorithm recursively continues to scan for edges along the direction determined by

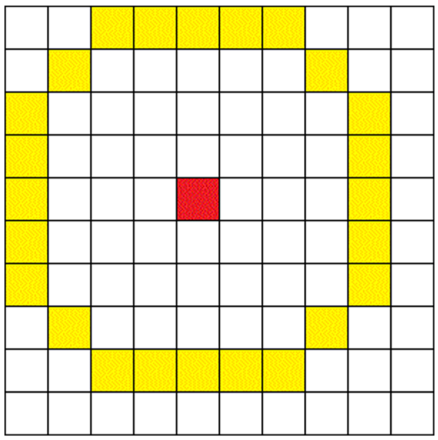

Fig. 16 In this figure each square represents a pixel in the image. The red pixel is the candidate pixel. The pixels in *yellow* represent the pixels that are examined to determine if the candidate pixel is a line edge pixel. If the pixels on the circle can be divided into 12 consecutive pixels whose value is below a certain threshold, followed by 12 pixels whose value is above another threshold, the candidate pixel is detected as a line edge

the already detected edges. At the end of this process, the four longest lines are detected as candidates to be the corridor lines. In order to validate the lines, another test is carried out. The intersection of each two lines is calculated and, if the distance between the resulting points is lower than some threshold, the lines' parameters are used as input to the controller. Figure 17 shows an image of the corridor, taken during flight. The four pairs of white points along the corridor lines mark the beginning and end of each detected line.

Technical Difficulties and Results During the experiments we encountered two main difficulties, both of which stem from the limitations of the camera and the driver used to access the images. 1) When operating indoors, with relatively low lighting, it is important to use a short exposure time when taking a picture, as otherwise the motion of the MAV will cause the image to be blurred. Usually this is done with increased sensor sensitivity to compensate for the shorter exposure. The camera on the AR.drone does not provide means to manually set the exposure time and sensitivity of the sensor. During flight, and especially while the MAV is in motion, some of the frames were corrupted, and lines could not be identified using our algorithm. Figure 18 shows an image blurred from the motion of the MAV. In this image the detected lines are shorter and the upper right line, L_{cr}, was not identified at all. During the flight experiments, when correct image lines were not detected, all control signals were updated to be 90 % of their previous value, accept for the thrust command, which was updated to be 98 % of its previous value. This cause the control signals

Fig. 17 Image of the corridor taken during flight. The four pairs of white points along the corridor lines mark the beginning and end of each detected line

Fig. 18 An example of a corrupted image of the corridor taken during flight. The image is blurred due to a combination of MAV motion, low lighting and relatively long exposure. The corridor lines are not detected accurately. This is one of the difficulties that needs to be overcome when using low quality cameras while operating indoors. During the flight experiments, when correct image lines were not detected, all control signals were updated to be 90 % of their previous value, accept for the thrust command, which was updated to be 98 % of its previous value. This causes the control signals to fade to zero smoothly, maintaining smoother flight characteristics over corrupted frames

to fade to zero smoothly, maintaining smoother flight characteristics over corrupted frames. 2) We found that there is a significant delay from the moment the image was taken up to the moment where the image was available for the image processing algorithm. It is difficult to calculate the exact delay, however it took approximately 30 milliseconds from the moment the command to capture an image was given to the moment where the image processing was initiated. This delay significantly reduces the system stability margins, and limits the gain of the derivatives, specifically $k_{\phi d}$. This may cause a relatively undamped behavior. Ideally, for these purposes, a camera with lower delays should be used, with higher control gains, in order to achieve a more damped system.

Despite these technical difficulties, consistent stable flights were achieved during the experiments. A video demonstrating the performance of one of the flights is a available at http://www.youtube.com/playlist?list=PLnhKrsOYTPlLlqhl0tsSM0UKW6qc95U7r. This demonstration proves the feasibility of the proposed method in practice, even when using a relatively low cost, low performance MAV with a slow processor and a low quality camera.

6 Conclusions

A new approach for the guidance and control of a MAV was presented. The main advantages of the controller presented compared with existing methods are: 1) It does not require the use of accelerometers or an external system to measure attitude. To the best of our knowledge this is the only work published describing MAV control that does not use accelerometers or an external method for attitude estimation (e.g. [12, 18, 19, 22, 24]). 2) The presented controller both stabilizes and guides the MAV. Most visual servoing papers dealing with MAVs only address the guidance aspect and use a separate controller to stabilize the MAV, usually using accelerometers and complex vision algorithms (e.g. [1, 2, 16]). 3) To the best of our knowledge, all of the publications on MAV

control (e.g., [4, 13, 20, 24]) include some method for velocity estimation. This is the only method we are aware of that does not require explicit pose or velocity estimation (which can be difficult to achieve in indoor environments).

The controller architecture and gains presented are suitable for any MAV with its own specific low level angular velocity controller. This approach can be used as the primary method or as a backup in case of sensor malfunction. The feasibility of this method was demonstrated in experiments using a commercially available MAV. Future work includes the expansion of the controller to include more maneuvers such as cornering, flight through staircases and obstacle avoidance to create a fully autonomous MAV which is capable of executing complete tasks in an indoor environment.

Acknowledgments This research was supported by the ISRAEL SCIENCE FOUNDATION and by the Helmsley Charitable Trust through the Agricultural, Biological and Cognitive Robotics Center of the Ben-Gurion University of the Negev.

References

1. Araar, O., Aouf, N.: Visual servoing of a quadrotor uav for autonomous power lines inspection. In: MED2014, pp. 1418–1424
2. Bills, C., Chen, J., Saxena, A.: Autonomous mav flight in indoor environments using single image perspective cues. In: International conference on Robotics and automation (ICRA), pp. 5776–5783 (2011)
3. Blosch, M., Scaramuzza, D., Weiss, S., Siegwart, R.: Vision Based Mav Navigation in Unknown and Unstructured Environments. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 2128. 2 (2010)
4. Bouabdallah, S., Siegwart, R.: Full control of a quadrotor. In: Intelligent robots and systems, pp. 153–158 (2007)
5. Bouabdallah, S., Siegwart, R.: Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. International Conference on Robotics and Automation Barcelona Spain (2005)
6. Bristeau, P., Callou, F., Vissiere, D., Petit, N.: The Navigation and Control Technology inside the Ar. Drone Micro Uav. In: 18Th IFAC World Congress, Vol. 18, pp. 1477–1484 (2011)
7. Mark Cutler, N., Kemal, U., Michini, B., How, J.P.: Comparison of fixed and variable pitch actuators for agile quadrotors. AIAA Paper (2011-6406) (2011)
8. Efraim, H., Arogeti, S., Shapiro, A., Weiss, G.: Output feedback control of micro aerial vehicle in indoor environment. MED, pp. 728–734 (2015)
9. Carrillo, G., et al.: Stabilization and trajectory tracking of a quad-rotor using vision. J. Intell. Robot. Syst. **61**(1–4), 103–118 (2011)
10. Fernandes, L.A.F., Oliveira, M.M.: Real-time line detection through an improved hough transform voting scheme. Pattern Recogn. **41**(1), 299–314 (2008)
11. Guenard, N., Hamel, T., Mahony, R.: A practical visual servo control for an unmanned aerial vehicle. IEEE Trans. Robot. **24**(2), 331–340 (2008)
12. Hehn, M., D'Andrea, R.: A Flying Inverted Pendulum. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 763770. 2 (2011)
13. Hoffmann, G., Waslander, S., Tomlin, C.: Quadrotor Helicopter Trajectory Tracking Control. In: AIAA Guidance, Navigation and Control Conference and Exhibit, pp. 1–14 (2008)
14. Sturm, J., Engel, J., Cremers, D.: Camera-Based Navigation of a Low-Cost Quadrocopter. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 28152821. 2 (2012)
15. Khalil, H.K.: Nonlinear systems. Prentice Hall (2002)
16. Klein, G., Murray, D.: Parallel Tracking and Mapping for Small Ar Workspaces. In: 6Th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2007, pp. 225234. 2 (2007)
17. Leishman, R., Macdonald, J., Beard, R., McLain, T.: Quadrotors & accelerometers. IEEE Control Systems Magazine (2014)
18. Liu, C., Zhou, Z., Fu, X.: Attitude determination for mavs using a kalman filter. Tsinghua Sci. Technol. **13**, 593–597 (2008)
19. Euston, R., Mahony, J., Kim, M., Coote, P., Hamel, T.: A Complementary Filter for Attitude Estimation of a Fixed-Wing Uav. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008
20. Mellinger, D., Michael, N., Kumar, V.: Trajectory generation and control for precise aggressive maneuvers with quadrotors The International Journal of Robotics Research (2012)
21. Pounds, P., Mahony, R., Corke, P.: Modelling and Control of a Quad-Rotor Robot. In: Proceedings Australasian Conference on Robotics and Automation 2006. Australian Robotics and Automation Association Inc. (2006)
22. Cha, S.-H., Mahony, R., Hamel, T.: A Coupled Estimation and Control Analysis for Attitude Stabilisation of Mini Aerial Vehicles. In: Proceedings of the 19Th Australiasian Conference on Robotics and Automation (2006)
23. Hamel, T., Mahony, R., Pflimlin, J.-M.: Complementary filter design on the special orthogonal group so (3). In: 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference, CDC-ECC05, pp. 14771484. IEEE. (2005)
24. Schöllig, A., Augugliaro, F., D'Andrea, R.: A platform for dance performances with multiple quadrocopters. Improving Tracking Performance by Learning from Past Data, 147 (2012)

25. Tournier, G.P., Valenti, M., How, J.P., Feron, E.: Estimation and Control of a Quadrotor Vehicle Using Monocular Vision and Moire Patterns. In: AIAA Guidance, Navigation and Control Conference and Exhibit, pp. 21–24. Citeseer (2006)
26. Weiss, S., Achtelik, M., Lynen, S., Chli, M., Siegwart, R.: Real-Time Onboard Visual-Inertial State Estimation and Self-Calibration of Mavs in Unknown Environments. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 957–964. 2 (2012)

Hanoch Efraim is Phd student in the Department of Mechanical Engineering, Ben-Gurion University of the Negev, Israel. His research interests include robotics, special MAV structures and UAV guidance and control.

Shai Arogeti is a Senior Lecturer in the Department of Mechanical Engineering, Ben-Gurion University of the Negev, Israel. His research interests include nonlinear control, robotics, automotive control and fault diagnosis.

Amir Shapiro received the B.Sc., M.Sc., and Ph.D. degrees in Mechanical Engineering from the Technion, Israel Institute of Technology, Haifa, in 1997, 2000, and 2004 respectively. Currently he is an Associate Professor and director of the robotics and control laboratory at the Department of Mechanical Engineering of Ben-Gurion University of the Negev, Beer-Sheva, Israel. On 2005-2006 he was a post doctoral fellow at the Robotics Institute of Carnegie Mellon University, Pittsburgh, PA. His interests include locomotion of multi-limbed mechanisms in unstructured complex environments, motion planning algorithms for multi-limbed robots, robot grasping-design, control, and stability analysis, climbing robots, multi-copter control, and agriculture robotics.

Gera Weiss is a faculty at the department of computer science at the Ben Gurion University of the Negev. His research interests include software engineering, formal methods, and control-theory.