A Vision-Guided Autonomous Quadrotor in An Air-Ground Multi-Robot System

Wei Li, Tianguang Zhang, Kolja Kühnlenz

Abstract—In this paper, a vision-guided autonomous quadrotor in an air-ground multi-robot system has been proposed. This quadrotor is equipped with a monocular camera, IMUs and a flight computer, which enables autonomous flights. Two complementary pose/motion estimation methods, respectively marker-based and optical-flow-based, are developed by considering different altitudes in a flight. To achieve smooth take-off, stable tracking and safe landing with respect to a moving ground robot and desired trajectories, appropriate controllers are designed. Additionally, data synchronization and time delay compensation are applied to improve the system performance. Real-time experiments are conducted in both indoor and outdoor environments.

I. INTRODUCTION

There is a growing interest in SLAM using unmanned vehicles, including MAVs (Micro Aerial Vehicles) and ground robots. This is applicable both in civilian and military domains, such as monitoring and surveillance in regions, which can not be reached by satellites and radars. Aerial and ground vehicles have complementary characteristics: aerial vehicles are able to give a complete 3D-sensing of the interesting regions in the environment at adequate altitudes. They can also scan the interesting regions at higher speed compared with ground vehicles. But due to the limited payload, the equipped sensors and batteries have usually poor performances. In contrast, ground vehicles carry more powerful sensors and batteries, which result in more accurate estimation and longer running time. Since they are set to move on the ground, they give only limited field of view and low moving speed in the environment. Aiming at combining the advantages of aerial and ground vehicles, we designed an air-ground multi-robot system, consisting of a quadrotor and a ground robot, for the execution of exploration tasks in unknown environments. Based on [1], we present here our improved results of developing the quadrotor in this system. For a controlled flight, accurate pose and motion information is required. There are various techniques available for pose and motion estimation of the quadrotor using different sensors. A quadrotor in [2] equipped with a laser ranger and IMUs (Inertial Measurement Units) was able to achieve 2D self-localization in a known indoor environment. In [3], a monocular on-board camera was set to face specified moire patterns to get the relative position and orientation of the quadrotor at a low rate. A novel two-camera method was proposed in [4] for the pose estimation of a quadrotor in 6 DoF (Degrees of Freedom). The two cameras were set to see each other and one of them was off-board. In [5] integrations of multi-sensor information for the estimation of self-motion and 3D scene structure were introduced. Considering this idea, on-board pose/motion estimation at high frequency using minimized sum of sensors is possible.

In recent years, a pair of quadrotors were developed for unmanned flight. In [6] an off-board monocular camera and on-board sensors, including a sonar and a laser ranger, were employed for the hovering control of a quadrotor with artificial markers. The $x$, $y$ positions and the heading of the aerial vehicle were obtained by using sensor data from the off-board camera or the on-board laser ranger. The altitude was estimated using the sonar measurement. Simple PID controllers were implemented for the pose control. In [7] and [8] hovering, terrain following, collision avoidance and landing of a quadrotor, which were based on the optical flow captured by an on-board camera in static, textured indoor environment, were presented. Nonlinear controller was applied for the motion control and there was no heading control concerned. A vision-guided quadrotor was developed in [9]. Nonlinear control and proof of the system stability were presented. This aerial vehicle was able to track moving target (a manually pulled cart). A landing on desired target was not achieved and no designed trajectories for the take-off and landing actions were concerned. The computational tasks were partly locally executed. An autonomous quadrotor with a flight computer was presented in [10]. IMUs, an ultrasonic sensor and four infrared sensors were mounted on this aerial

Fig. 1. system overview
vehicle for the on-board estimation of relevant system states. Autonomous flights with collision avoidance were realized in an obstacle free, closed environment.

In this paper, a vision-guided autonomous quadrotor with minimizing sensing is presented. By using only on-board resources (a monocular camera, IMUs and a flight computer) this MAV is able to take off, track, land with respect to a moving ground robot. To achieve this, vision-based 3D pose/motion estimation, based on artificial reference markers or optical flow, is developed. An EKF (Extended Kalman Filter) considering motion in non-inertial frame is employed for the multi-sensor data fusion. To perform desired flight from take-off to landing, well designed trajectories and appropriate controllers are proposed. Data synchronization and time delay compensation are also considered to improve the system reliability. Finally, autonomous flights are conducted for both indoor and outdoor environments.

The remainder of this paper is organized as follows: Section II presents the system overview. Then section III describes the pose and motion estimation based on visual and inertial data. The proposed control design is introduced in section IV. Evaluation of the system performance is given after illustration of the experimental results in section V. Conclusions and future work are provided in section VI.

II. SYSTEM OVERVIEW

The entire system consists of two parts: a quadrotor with a monocular camera, a triaxial accelerometer, three piezogyros and a flight computer, a ground robot with a patterned board and two LED markers. This is illustrated in Fig. 1.

A. Quadrotor

A Hummingbird mini-quadrotor from Ascending Technologies1 with a X-3D scientific controller board is applied in this work. Pairwise control of motors enables not only free rotations about body x-, y- and z-axis, but also linear accelerations, translations in 3D. The pre-installed controller on the X-3D board takes charge of the attitude, yaw rate and thrust control at 1KHz. So there are four available inputs for the quadrotor: \( u_\phi, u_\theta, u_\psi, u_{th} \). They are associated with the system variables using following approximated equations

\[
\begin{align*}
    u_\phi &= c_1 \arcsin(-s_\phi c_\theta) + d_1 \\
    u_\theta &= c_2 \theta + d_2 \\
    u_\psi &= c_3 \psi + d_3 \\
    u_{th} &= c_4 \|F\|/U
\end{align*}
\]

where \( \phi, \theta \) and \( \psi \) are Tait-Bryan angles with respect to the ground (earth), \( c_1, c_2, c_3, c_4, d_1, d_2, d_3 \) are constants, \( F \) is the thrust produced by propellers, \( U \) is the current battery voltage. To simplify expressions, we use here \( s(\cdot) \) and \( c(\cdot) \) instead of \( \sin(\cdot) \) and \( \cos(\cdot) \).

IMUs give the measurement \( \mathbf{a} \) of linear acceleration (including also the gravity acceleration) and the measurement \( \mathbf{r} \) of angular velocity in the body frame. \( \phi, \theta \) are estimated on the X-3D board using the IMUs measurements. These data are produced also at 1KHz. Considering limited payloads of the quadrotor, a light “Firefly MV” (37g) from Point Grey Research Inc.2 is selected as the on-board camera. A fisheye lens with 2.1mm focal length enlarges the field of view of the camera to ensure the tracking of the markers.

To enable the on-board implementation of the closed-loop control, a flight computer (“atom-board” with an Intel Atom CPU), also from Ascending Technologies, is mounted on the quadrotor. With limited weight and dimension (80g, 7.5cm x 5.8cm), it is suited for this MAV. An Ubuntu linux OS is installed for the execution of all computational tasks.

B. Ground Robot

A Pioneer 3-DX Roboter from Mobilerobots Inc.3 is used to approach desired trajectories on the ground (x-y plane). It can move along the body x-axis and rotate about the body z-axis. The motion measurements \( \mathbf{v}_g = [v_g, 0, 0]^T, \mathbf{r}_g = [0, 0, \dot{\Psi}_g]^T \) and the acceleration measurements \( \mathbf{a}_g = [a_g, 0, 0]^T, \dot{\mathbf{r}}_g = [0, 0, \dot{\Psi}_g]^T \) in its body frame are sent to the quadrotor via a wireless UDP connection. Two LED markers are located on the surface of this robot and are tracked by the quadrotor. A flat board with random patterns is installed on the top of the Pioneer roboter for the take-off and landing actions of the quadrotor.

C. Control Loop

In Fig. 2, the entire closed-loop control is illustrated. The aerial vehicle is simplified to be a second-order nonlinear system. It takes the acceleration and rotation of the quadrotor in the body frame, together with the motion and acceleration of the ground robot, as the system inputs. The relevant system outputs are states variables \( \mathbf{p} = [X, Y, Z]^T, \dot{\mathbf{p}} \) and \( \dot{\Psi} \) produced also at 1KHz.

1http://www.ascetec.de
2http://www.ptgrey.com
3http://www.mobilerobots.com
of the quadrotor with respect to the ground robot. They are observable using the vision-based pose/motion estimation. For the marker-based estimation, a state observer, the EKF, is used additionally to estimate the states variables more accurately and more smoothly. The estimated states are pushed forwards to compensate the time delay and then fed into the outer-loop controller. With respect to the desired pose, motion and acceleration, the outer-loop controller produces corresponding commands to the inner-loop controller, which is pre-installed. Then controlled acceleration and rotation of the quadrotor are obtained.

III. POSE/MOTION ESTIMATION

As feedback of the control loop, the pose and motion of the quadrotor can be estimated using detected marker positions or optical flow.

A. Frame Definitions

The key frames involved in this vision-based pose/motion estimation are all illustrated in Fig. 3.

- the inertial frame $S_i$.
- the object frame $S_g$, fixed on the ground robot (the height of the robot is omitted).
- the body frame of the quadrotor $S_q$.
- the camera frame $S_c$, it has the same orientation as $S_q$.
- the image plane $S_i$.

B. Marker-Based Estimation

Only two LED markers are employed as the flight guidance for the quadrotor at adequate altitudes. Using the pinhole camera model, the projections of the markers in $S_i$ can be expressed as follows

$$p_{1/2,i} = \begin{bmatrix} y_{1/2,c} \\ z_{1/2,c} \end{bmatrix},$$

where $p_{1/2,c} = \begin{bmatrix} x_{1/2,c} \\ y_{1/2,c} \\ z_{1/2,c} \end{bmatrix}$ are positions of the two markers in $S_c$. After a series of homogeneous transformations, there is also

$$P_{1/2,c} = CT_q P_{q}^{-1} \begin{bmatrix} p_{1/2} \end{bmatrix},$$

where $T$ is the homogeneous transformation matrix, $p_{1/2} = [x_{1/2}, y_{1/2}, z_{1/2}]^T$ are positions of the two markers in $S_i$. Details for the marker detection are available in [11]. If the Tait-Bryan angles $\Phi$, $\Theta$ and $\Psi$ of the quadrotor in $S_g$ are all known, then we have four equations for 3 unknown variables $X$, $Y$ and $Z$ by combining (2) and (3). Since $\phi$, $\theta$ are estimated from IMUs data and $\Phi = \phi$, $\Theta = \theta$, we can use the gradient descending method iteratively to find the optimal value of $\Psi$, which gives optimized solution of $p$. $\Psi^{-i}$ found in $i$-th iteration will be disturbed, so that small variation of calculated orientation of the markers in $S_i$ can be observed. Then $\Psi^{-i}$ will be pushed in the direction to minimize the error between the calculated orientation and the true value in $S_i$.

Instead of using direct time derivative of estimated pose, the EKF-based multi-sensor data fusion is applied to give more accurate motion estimation. The pose estimation will also be smoothed. For convenience, $S_g$ is selected as the reference frame and is, unlike in other works, non-inertial as fixed on a moving target. The system state vector and the input vector are $x = [p^T, \dot{p}^T, \Phi, \Theta, \Psi]^T$, $u = [a^T, r^T, v_g^T, a_g^T, r_g^T]^T$ respectively. The conventional system dynamic model is written as

$$f(x, u, \omega) = \begin{bmatrix} \dot{p} \\ Qr - Ir_g \end{bmatrix} + \omega,$$

where $\dot{x} = f(x, u, \omega)$, $\omega$ is the process noise, the acceleration vector $\dot{p}$ consists of three parts : $\dot{p}_{\text{trans}} = gR_g a - a_g$ caused by translational terms, $\dot{p}_{\text{rot}} = -gR_l^{-1}R_g p - gR_l^{-1}R_g (2\dot{p} + v_g)$ caused by rotational terms and the gravity acceleration $g = [0, 0, g]^T$. $R$ is the rotation matrix, $Q$, $I$ are the transformation matrices with

$$Q = \begin{bmatrix} 1 & s\Phi \tan \Theta & c\Phi \tan \Theta \\ c\Phi & -s\Phi & 0 \\ s\Phi \sec \Theta & c\Phi \sec \Theta & 0 \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$ 

The system measurement model considering relevant system states is written as

$$h(x, v) = \begin{bmatrix} p \\ \Psi \end{bmatrix} + v,$$

where $v$ is the measurement noise.

After linearization of the system dynamic and measurement models, we can use the EKF, which is described in [12][13], to fuse the multi-rate (visual data @15Hz and inertial data @100Hz) multi-sensor data @100Hz together. As described in [14], the criterion, minimizing the residual prediction error for the external ground truth, is employed by tuning the EKF parameters to obtain the optimal performance of the data fusion.
C. Optical-Flow-Based Estimation

For the flight near the surface of the ground robot, not all markers can be detected by the camera. In this case, another complementary method optical-flow-based estimation is considered. The optical flow can be computed using the pyramidal Lucas-Kanade algorithm[15]. Based on geometry and the image Jacobian matrix, not only the linear velocity, but also the angular velocity of the quadrotor are available. This computation is very complex and omitted here. After integration of the motion over time, the pose is also obtained. The data refresh rate is 15Hz. To improve the quality of the estimation, a simple outlier canceling algorithm is designed for the on-board implementation. Optical flow is first detected at all 36 feature points and used to estimate the motion of the quadrotor, then the estimated motion in $S_p$ is projected into $S_i$ to calculate the corresponding movement of every feature point. The calculated values are compared with the detected movements in $S_i$. The relative errors are employed to indicate the 12 “bad” measurements that should be canceled.

Note that, there are always drifts due to long time integration with out any corrections. The quadrotor is guided by optical flow just for several seconds in the take-off and landing actions, therefore the optical-flow-based pose/motion estimation is still accurate. No pose information can be directly obtained using this method, the EKF is not applied here for this reason.

D. Data Synchronization and Time Delay Compensation

Since multi-sensor data are employed to achieve accurate pose/motion estimation, data synchronization is required. For the marker-based estimation, the multi-sensor data have to be synchronized twice : 1. Before using detected marker positions and estimated $\Phi$, $\Theta$ for the pose estimation, $\Phi$, $\Theta$ must be shifted along the time axis to match the results of the image processing, which take much more time for data transmission and processing. 2. After the vision-based pose estimation, the estimated pose is pushed forward using the motion estimation and synchronized with the IMUs measurements. Then the EKF can be correctly updated. For the optical-flow-based estimation, it is necessary to shift $\Phi$, $\Theta$ along the time axis to match the detected optical flow before carrying out the motion estimation.

Sensor data from the ground robot have a low refresh rate and it is very difficult to achieve an accurate interpolation, therefore no synchronization is carried out for these data. However this has only slight influence on the system performance, because the ground robot does not give any sudden movements.

As discussed, there is non-negligible, varying data transmission and processing time in the control loop. This can lead to undesired oscillation or instability. To improve the system reliability, a second-order time function is applied as the state predictor to compensated the time delay. The system state $x_i$ is assumed to be $x_i = k_{i,0} + k_{i,1} t + k_{i,2} t^2$, where $k_{i,0}$, $k_{i,1}$ and $k_{i,2}$ are corresponding coefficients and are estimated using last $n$ recorded states. Using this second-order time function, the pose and motion estimations are pushed forward and fed into the controllers as feedbacks. This structure of the predictor has an advantage : filtering of the states estimations.

IV. FLIGHT CONTROL

In this section, appropriate controllers are designed to enable a complete flight of the quadrotor from take-off to landing with respect to the moving target.

A. Take-Off

Smooth take-off means zero velocity along the $Z$-axis for both beginning and end of this action. A cubic spline is selected as the take-off trajectory to fulfill this requirement. It can be written as

$$Z_d = Z_{d,0} + 3 (Z_{d,T} - Z_{d,0}) \frac{t^2}{T^2} - 2 (Z_{d,T} - Z_{d,0}) \frac{t^3}{T^3},$$  (6)

where $T$ is the take-off time, $Z_{d,0}$ and $Z_{d,T}$ are desired start and end positions.

Strong ground effect, described in [16], is observed in flights near the surface of the ground robot and make the control of the flight difficult. To deal with the model inaccuracy caused by the ground effect, a sliding mode controller based on [17] is designed to enable the desired flying behavior. The sliding mode must exist for the switching surface $s_1$ and $s_1$ should be reachable from any initial states, then the control output $\bar{Z}$ is written as

$$\bar{Z} = \bar{Z}_d + \alpha \dot{\chi} + \beta \chi + M_1 \text{sign}(s_1) + M_2 s_1,$$  (7)

where $\alpha$, $\beta$, $M_1$ and $M_2$ are positive constants, $\chi = Z_d - Z$,

$$s_1 = \dot{\chi} + \alpha \dot{\chi} + \beta \int_0^t \chi dt.$$  (8)

For the control of $X$, $Y$ and $\Psi$ in the take-off, the same controller, which is designed for the tracking flight and will be introduced later, is applied.

B. Tracking

As part of the complete flight, stable tracking is also to be achieved. Considering the desired pose, velocity and acceleration, a backstepping controller is preferred for the flight control. An augmented backstepping controller with additional integral component in [18] is able to give reliable performance for systems under non-negligible disturbances (wind, etc.). This integral backstepping controller is used for the control of $X$, $Y$, $Z$, $\Psi$ and the $Z$ control is taken as example to give a simple description. The control output is written as

$$\bar{Z} = \bar{Z}_d + \gamma_1 \chi_1 + \gamma_2 \chi_2 - \alpha_1 \beta_1 \zeta_1 + \beta_2 \zeta_2,$$  (9)

where $\gamma_1 = 1 + \beta_1 - \alpha_1^2$, $\gamma_2 = \alpha_1 + \alpha_2$, $\alpha_1$, $\alpha_2$, $\beta_1$ and $\beta_2$ are positive constants, $\chi_1 = Z_d - Z$, $\zeta_1 = \int_0^t \chi_1 dt$, $\chi_2 = \dot{\chi} + \alpha_1 \chi + \beta_1 \zeta_1$, $\zeta_2 = \int_0^t \chi_2 dt$. By applying the La Salle’s principle, all subsystems are proved to be asymptotically stable.
C. Landing

In contrast to the take-off and tracking, a speed limitation along the $Z$-axis is required for the safe landing. An exponential curve with the speed $\dot{Z}_d$ and acceleration $\ddot{Z}_d$ is selected as the landing trajectory, where a positive constant $\lambda$ is the corresponding damping factor and

$$
\begin{aligned}
\dot{Z}_d &= -\lambda Z \\
\ddot{Z}_d &= \lambda^2 Z .
\end{aligned}
$$

Compared with the take-off, Another sling mode controller is proposed here to overcome strong ground effect and ensure the safe landing with speed control. The control output is written as

$$
\ddot{Z} = (-c \lambda + \lambda^2) Z - c \dot{Z} - M \text{sign}(s_2),
$$

where $c$ and $M$ are positive constants, the switching surface $s_2$ is defined as

$$
s_2 = \lambda Z + \dot{Z}.
$$

To ensure the reachability of the switching surface and the existence of the sliding mode, the conditions $c > \lambda$ and $M > |\dot{\delta}|_{\text{max}}/m$ must be fulfilled, where $|\dot{\delta}|_{\text{max}}$ is the maximal absolute value of the unmodelled force, $m$ is the mass of the quadrotor.

The tracking controller is used here for the $X$, $Y$ and $\Psi$ control.

D. Command Generation

All computed control outputs are defined in $S_g$, which is non-inertial, but the quadrotor is driven by the air flows in $S_I$ (or the world frame, which is quasi-inertial), therefore a transformation of the control outputs into an inertial frame is necessary:

$$
\ddot{p}_I = I_R \ddot{q} v_\phi + I_R \theta a_\theta + \ddot{R}_g p + 2 I_R \dot{R}_g p + I_R \ddot{p},
$$

where $p_I = [x, y, z]^T$ and is the position of the quadrotor defined in $S_I$. Considering the dynamic model, there is

$$
\ddot{p}_I = I_R \ddot{q} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \|F\|/m \\ g & 0 & 0 \end{bmatrix},
$$

then we have the relations between $\ddot{p}_I$ and $\dot{p}_I$, $\dot{\phi}$, $\dot{\theta}$, $\|F\|$. Using (1) the corresponding commands $u_\phi$, $u_\theta$, $u_{th}$ are generated. Similarly the yaw command $u_\psi$ is also available.

The refresh rate of the command generation is set to be identical with the refresh rate of the estimation results.

V. EXPERIMENTAL RESULTS

For the evaluation of the system performance, real-time experiments were carried out in both indoor and outdoor environments. Due to the limitation of the tracking system, which provides the pose information as ground truth merely for indoor experiments, therefore only an indoor flight is quantitatively evaluated.

In the indoor experiment, the ground robot approached a quasi-rectangle trajectory (1.6m x 1.4m) on the ground with a linear velocity of 0.1m/s for the translation and an angular velocity of 12°/s for the rotation. The quadrotor took off from the surface of the ground robot, guided by the optical flow for $Z > -0.50$ m and by the markers for $Z \leq -0.50$ m, than tracked the movement of the ground robot, at the end landed safely on the surface of the moving object, guided by the markers for $Z \leq -0.40$ m and by the optical flow for $Z > -0.40$ m. The desired values of $X$, $Y$ and $\Psi$ were always set to zero. For $Z$, a cubic spline from -0.16 m (considering the height of the landing gear) to -1.00 m in 10 seconds was selected in the take-off action, an exponential curve with a damping factor of 0.3 was the desired trajectory in the landing action, the desired tracking altitude of the quadrotor was 1.00 m.

The plots of relevant system states $x$, $y$, $z$ and $\dot{w}$ observed in $S_I$ during the flight are presented in Fig. 4. Black solid lines indicate the desired trajectory. The actual values and estimations of the states are plotted in green and red solid lines respectively. We can see that the ground robot was still moving along the $x$-axis during the landing of the quadrotor.

Fig. 5. estimation errors of the complete flight
errors are 0.039m, 0.036m, 0.014m, 2.6° respectively and plotted in red dashed lines. $x$, $y$, $z$ and $\psi$ were accurately estimated. The markers used for the tracking system were covered by the propellers occasionally, this led to some extreme values in the plots, for example, $E_x$ at $t = 95$s.

In Fig. 6, blue solid lines indicate the control errors. The corresponding RMS control errors are 0.057m, 0.054m, 0.022m, 2.9° respectively and plotted in red dashed lines, therefore a desired autonomous indoor flight was obtained. The motion estimation was also indirectly proved to be accurate. There were no frequent switchings between the two complementary estimation methods, the system stability was still warranted.

A similar outdoor flight was also carried out. The ground robot approached a quasi-rectangle trajectory (3.0m x 2.4m) on the ground with a linear velocity of 0.3m/s for the translation and an angular velocity of 18°/s for the rotation. The same take-off and landing trajectories as in the indoor experiment were applied. In real-time experiments, autonomous flights were successfully conducted even in the wind.

VI. CONCLUSIONS AND FUTURE WORK

This paper presented a vision-guided autonomous quadrotor developed in an air-ground multi-robot system. Using only on-board resources: a monocular camera, IMUs and a flight computer, accurate pose/motion estimation was obtained. In contrast to other related works, the system states were defined in a non-inertial frame and the corresponding multi-sensor data fusion was proposed. Appropriate controllers were designed and enabled a complete flight, including smooth take-off, stable tracking and safe landing on the moving ground robot with respect to the desired trajectories, in spite of unavoidable model inaccuracies and disturbances (ground effect, wind and etc.). Data synchronization and time delay compensation in the closed loop were discussed and applied to improve the system reliability. With real-time experiments in both indoor and outdoor environments, the system performance was validated.

For the future work, cooperations between the aerial and ground robots are considered.

VII. ACKNOWLEDGMENTS

This work is supported in part within the DFG excellence initiative research cluster Cognition for Technical Systems-CoTeSys, see also www.cotesys.org, the BMBF funded Bernstein Center for Computational Neuroscience Munich, see also www.bccn-munich.de, and the Institute for Advanced Study (IAS), Technische Universität München, see also www.tum-ias.de.

REFERENCES

[10] James F. Roberts, Timothy S. Stirling, Jean-Christophe Zufferey and Dario Floreano, Quadrotor using minimal sensing for autonomous indoor flight, EMAV, 2007
[14] Pieter Abbeel, Adam Coates, Michael Montemerlo, Andrew Y. Ng and Sebastian Thrun, Discriminative training of kalman filters. in Proceedings of Robotics : Science and Systems, Cambridge, USA, June 2005
[18] Yaolong Tan, Jie Chang, Jun He and Hualin Tan, Advanced nonlinear control strategy for motion control systems, in Proceedings of Power Electronics and Motion Control Conference (IPEMC), vol. 1, pp. 116-121, 2000