A tracking error control approach for model predictive position control of a quadrotor with time varying reference.

DENTLER, J., KANNAN, S., OLIVARES MENDEZ, M.A. and VOOS, H.

2016

Proceedings of the 2016 IEEE
International Conference on Robotics and Biomimetics
Qingdao, China, December 3-7, 2016

A tracking error control approach for model predictive position control of a quadrotor with time varying reference *

Jan Dentler, Somasundar Kannan, Miguel Angel Olivares Mendez, Holger Voos[1]

***Abstract*— In mobile robotic applications, a common problem is the following of a given trajectory with a constant velocity. Using standard model predictive control (*MPC*) for tracking of time varying trajectories leads to a constant tracking error. This problem is modelled in this paper as quadrotor position tracking problem. The presented solution is a computationally light-weight target position control (*TPC*), that controls the tracking error of *MPC*s for constantly moving targets. The proposed technique is assessed mathematically in the Laplace domain, in simulation, as well as experimentally on a real quadrotor system.**

I. INTRODUCTION

Within the last decade the importance of unmanned aerial vehicles (*UAV*) has been growing with the availability of low-cost commercial quadrotor solutions. The applications reach from video capturing and telecommunication to transportation. The low costs and ease of applicability of quadrotors makes them particularly suitable for environmental observation tasks. Current developments extend this field of activity even to underwater exploration [1] [2].

The increase of the level of autonomy for *UAV*s is crucial to develop solutions for large scale environmental observation. Accordingly a major research focus of *UAV* systems is the reliability and security. This is particularly challenging for small low-cost systems with limited computational power and payload. To achieve the required security level, prior control approaches of quadrotors have been analysing disturbance rejection e.g. for a backstepping and sliding-mode controller [3] and an integral backstepping controller in [4].

To implement more complex security measures e.g. limiting the operational space of the quadrotor, environmental and system constraints are considered with the help of model predictive control (*MPC*). As for its high computational burden, real-time *MPC* for small low-cost systems is challenging, but is pushed by recent developments of real-time capable *MPC* frameworks like *ACADO* [5], *GRAMPC* [6] or *CGMRES* [7], etc. In previous work the applicability of *CGMRES* [8] has been shown for single quadrotors and even multi *UAV* scenarios [9]. A standard *MPC* controller is minimizing an objective function to track a desired trajectory (state x) under minimization of the energetic effort (control u). The result is a trade-off between state and control tracking. This trade-off typically leads to a constant tracking error for constantly moving targets. For security purposes such an “offset” is not desired.

One strategy for an offset-free *MPC* trajectory control for quadrotors has been presented in [10]. The author is separating the control problem of a quadrotor into a path tracking *MPC* with underlying $H\infty$ attitude control. Offset free tracking is achieved by considering a disturbance error model in the path planning dynamics. The *MPC* control policy minimizes the offset by an integral part. The disadvantage of the proposed method is, that first the optimal controls are computed with *MPC* where the implemented constraints are respected. Afterwards the controls are altered with the integral part, which then might lead to constraint violation. A more detailed analysis on the described reference tracking is given in [11]. To tackle *UAV* tracking errors induced by e.g. wind gusts, [12] is proposing a similar hierarchical *UAV* control scheme considering tracking errors as output errors in the *MPC* scheme. [13] is presenting an offset free linear *MPC* for quadrotors by using disturbance modelling. The advantage of modelling the disturbance to minimize the offset is, that information about the disturbance can be used to optimize the systems behaviour. The drawback of disturbance modelling is the increased computational effort, as the disturbance has to be modelled e.g. augmenting the system dynamics and the disturbance has to be estimated. A summery of offset-free linear *MPC* control strategies is provided in [14], including disturbance model and observer, state disturbance observer and using a velocity form model.

Based on the *NMPC* control concept presented in [8], this work is presenting an alternative offset free control approach for constantly moving targets. The reduction of the offset is achieved by using an overlying tracking position control. The advantages of the proposed approach is the ease of implementation, minor computational effort and intuitive tuning. Furthermore the determined controls are not altered after the optimization process which ensures the respect of constraints by the *MPC*.

Section II is presenting a system model for a quadrotor, the *MPC* and an a simulative result that states the problem of a static tracking error. The proposed target position control approach is given in detail in section III. Section IV is showing experimental results on a real quadrotor (*AR.Drone 2.0*[2]). The work is concluded in section V.

*This work was supported by FNR ”Fonds national de la Recherche” (Luxembourg) through AFR ”Aides à la Formation-Recherche” Ph.D. grant scheme No. 9312118.

[1] Jan Dentler, Dr. Somasundar Kannan, Dr. Miguel Angel Olivares Mendez and Prof. Dr.-Ing Holger Voos are with Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, L-1359 Luxembourg, Luxembourg jan.dentler@uni.lu, somasundar.kannan@uni.lu, miguel.olivaresmendez@uni.lu, holger.voos@uni.lu

[2] Copyright ©2015 Parrot SA. All Rights Reserved.

II. NMPC CONTROLLER

This paper is extending the *NMPC* control presented in [8]. The considered system is an *AR.Drone 2.0*. For the control, the coordinate frames of the quadrotor are chosen according to figure 1.

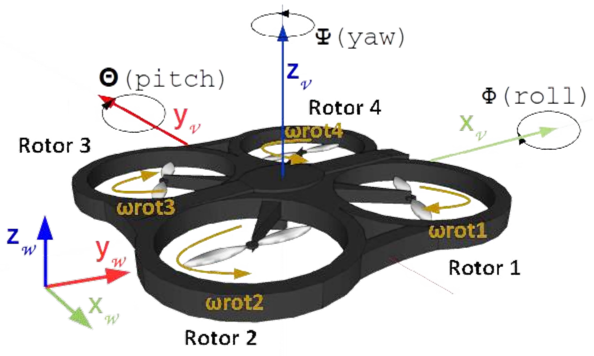


Fig. 1: Coordinate frame definition

The system states are composed by x, y, z, [m], Ψ [rad] in world coordinates $\mathscr{W}$ and forward and sideward velocity $\dot{x}_{\mathscr{V}}$ $[\mathrm{m\,s^{-1}}]$ respective $\dot{y}_{\mathscr{V}}$ in vehicle frame coordinates $\mathscr{V}$

$$\mathbf{x} = [x_{\mathscr{W}}, y_{\mathscr{W}}, z_{\mathscr{W}}, \Psi_{\mathscr{W}}, \dot{x}_{\mathscr{V}}, \dot{y}_{\mathscr{V}}]. \tag{1}$$

The controls of the quadrotor are defined as forward, sideward, upward $[\mathrm{m\,s^{-1}}]$ and yaw-velocity $[\mathrm{rad\,s^{-1}}]$

$$\mathbf{u} = \left[u_f, u_s, u_z, u_\Psi\right]. \tag{2}$$

The resulting system dynamics yield to

$$\begin{bmatrix} \dot{x}_{\mathscr{W}}(t) \\ \dot{y}_{\mathscr{W}}(t) \\ \dot{z}_{\mathscr{W}}(t) \\ \dot{\Psi}_{\mathscr{W}}(t) \\ \ddot{x}_{\mathscr{V}}(t) \\ \ddot{y}_{\mathscr{V}}(t) \end{bmatrix} = \begin{bmatrix} \dot{x}_{\mathscr{V}}(t)\cos(\Psi) - \dot{y}_{\mathscr{V}}\sin(\Psi) \\ \dot{x}_{\mathscr{V}}(t)\sin(\Psi) + \dot{y}_{\mathscr{V}}\cos(\Psi) \\ 1 \cdot u_z(t) \\ 0 \cdot \Psi(t) + 1.6 \cdot u_\Psi(t) \\ -0.5092 \cdot \dot{x}_{\mathscr{V}}(t) + 1.458 \cdot u_f(t) \\ -0.5092 \cdot \dot{y}_{\mathscr{V}}(t) + 1.458 \cdot u_s(t) \end{bmatrix}. \tag{3}$$

The model predictive controller is based on solving the following optimal control problem (*OCP*) (4)-(8) over a receding horizon:

$$\min_{\mathbf{u}} J = \int_{t_0}^{t_f} (\mathbf{x}_{des} - \mathbf{x})^\top \mathbf{Q} (\mathbf{x}_{des} - \mathbf{x}) + \mathbf{u}^\top \mathbf{R} \mathbf{u} \; \mathrm{d}\tau \tag{4}$$

s.t.

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{x}_{\mathscr{V}}(t)\cos(\Psi) - \dot{y}_{\mathscr{V}}\sin(\Psi) \\ \dot{x}_{\mathscr{V}}(t)\sin(\Psi) + \dot{y}_{\mathscr{V}}\cos(\Psi) \\ 1 \cdot u_z(t) \\ 0 \cdot \Psi(t) + 1.6 \cdot u_\Psi(t) \\ -0.5092 \cdot \dot{x}_{\mathscr{V}}(t) + 1.458 \cdot u_f(t) \\ -0.5092 \cdot \dot{y}_{\mathscr{V}}(t) + 1.458 \cdot u_s(t) \end{bmatrix} \tag{5}$$

$$c \le (u - \bar{u})^2 - (u_{max} - \bar{u})^2 : \forall u : u_{max} = 1 \vee \bar{u} = 0 \tag{6}$$

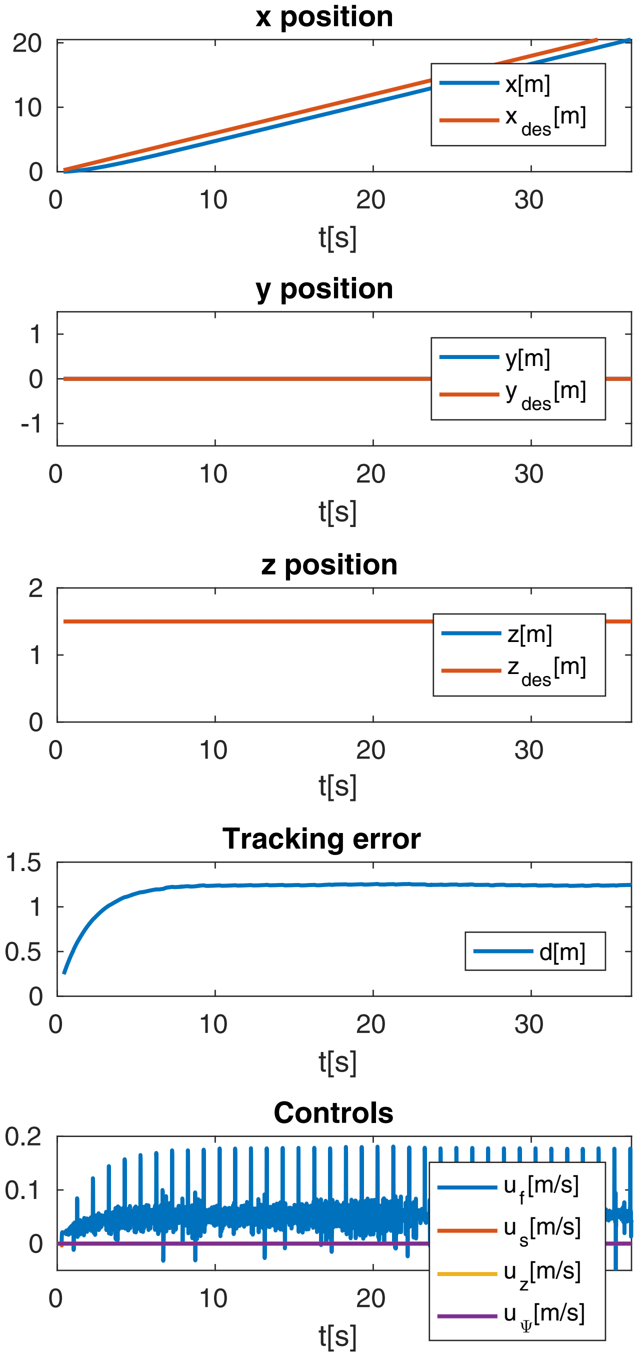
$$\mathbf{x}(0) = \left[0, 0, 0, 0, 0, 0\right] \tag{7}$$

$$\mathbf{Q} = \mathscr{D}\{[1, 1, 8, 3, 1.5, 1.5]\}, \quad \mathbf{R} = \mathscr{D}\{[1.5, 1.5, 3.0, 3.1]\} \tag{8}$$

The optimal controls are determined to minimize a given performance index J (4). Cost function J includes state and control reference tracking. Stabilizing solver parameters are given in [8]. The penalty matrices Q and R are chosen to limit the drone movement to the xy-plane by highly penalizing an alteration from z_{xdes}. The velocity and control tracking are chosen empirically to smoothly stabilize the system in static target positions. If the target position is moving with a constant velocity, the trade-off between control position and velocity tracking leads to a constant tracking error. In Fig.2 the *NMPC* approach is used to track a target

$$\mathbf{x}_{des}(t) = \begin{bmatrix} 0.2\mathrm{m\,s^{-1}} \cdot t & 0 & 1.5 & 0 & 0 \end{bmatrix}. \tag{9}$$

In real mobile robotic applications, target positions changes are often triggered by discrete events. The corresponding estimation of the velocity is therefore not trivial. For this


x position

y position

z position

Tracking error

Controls

Fig. 2: Simulation of *NMPC* tracking of constantly moving target

reason, the velocity states are not adjusted according to the position change in this scenario. This leads to an exacerbation of the problem of a static tracking error for the moving position. Therefore, errors in the velocity tracking are represented in the shown example. For the standard *MPC* with the *OCP* (4)-(8), Fig.2 is showing the tracking error expressed via Euclidean distance

$$d = \left\| \begin{matrix} e_x \\ e_y \\ e_z \end{matrix} \right\| = \left\| \begin{matrix} x_{des} - x \\ y_{des} - y \\ z_{des} - z \end{matrix} \right\|. \tag{10}$$

The plot shows that d is converging towards a constant value which represents a constant tracking error.

III. Target position control

The close loop behaviour of the *NMPC* can be described as a PT_1 element. In the Laplace domain, this is equal to system function

$$x(t) = (1 - e^{-\frac{t}{T}}) x_{des}(t) \longleftrightarrow G(s) = \frac{1}{1 + Ts} \tag{11}$$

In the following, variables of the Laplace domain are marked with $\hat{}$. Under use of the step $\varepsilon(t)$, the input signal of constant velocity v_0 (ramp) can be described as

$$x_{des}(t) = v_0 \cdot t \cdot \varepsilon(t) \longleftrightarrow U(s) = \frac{v_0}{s^2}. \tag{12}$$

This leads to a

$$U(s) \cdot G(s) = \frac{\frac{1}{T}}{(\frac{1}{T} + s)s^2} \tag{13}$$

The inverse Laplace transformation of (13) results in the system response

$$x(t) = v_0 T \left(e^{-\frac{t}{T}} + \frac{t}{T} - 1 \right) \varepsilon(t) = v_0 t + v_0 T \left(e^{-\frac{t}{T}} - 1 \right) \tag{14}$$

This system response is illustrated in Fig.3 for arbitrarily chosen $T = 2\text{s}$ and $v_0 = 0.2\text{m}\,\text{s}^{-1}$. As it shows the same behaviour as the the simulation of the x-position in Fig.2, As T and v_0 are chosen arbitrarily, this does not prove the consistency of the model, but justifies the modelling approach.

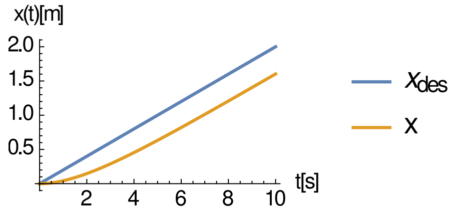


Fig. 3: System response for $T = 2\text{s}$, $v_0 = 0.2\text{m}\,\text{s}^{-1}$

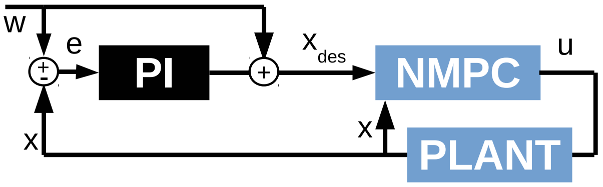


Fig. 4: Target position control structure

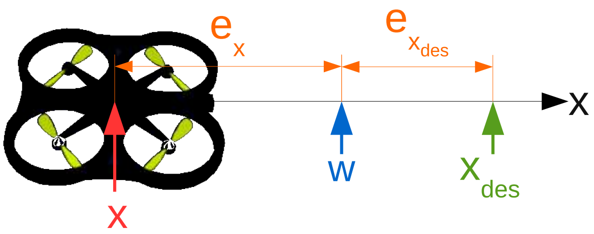


Fig. 5: Target position control idea

To get rid of this static tracking error, the basic concept is an adaptation of the target position $\mathbf{x}_{des}$. For this reason the system is extended with an outer control loop according to Fig.4 with the new input w and error $e = w - x$. Where the trajectory reference signal (9) is connected to the new input w. Fig.4 illustrates the idea for a displacement in x-direction. Instead of the actual target w, an altered target x_{des} is given to the *NMPC*. This alteration e_{xdes} is increasing with the tracking error e_x The resulting system response is higher, which leads to a convergence towards w. A linear interpolation of the target position (P-part) and an integral (I-part) leads to the control law

$$\begin{align} x_{des}(t) &= w(t) + K_P \cdot e(t) + K_I \cdot \int_0^t e(t)\,dt \tag{15} \\ \longleftrightarrow \hat{x}_{des} &= \hat{w} + K_P \cdot \hat{e} + K_I \cdot \frac{\hat{e}}{s} \tag{16} \end{align}$$

the system control can be composed:

$$\begin{align} \hat{x} = \frac{1}{1 + Ts} \left(\hat{w} + K_P \cdot (\hat{w} - \hat{x}) + K_I \cdot \frac{(\hat{w} - \hat{x})}{s} \right) \tag{17} \\ \tag{18} \end{align}$$

which yields to

$$\frac{\hat{x}}{\hat{w}} = \frac{K_I + (1 + K_P)s}{K_I + (1 + k_p)s + s^2 T} \tag{19}$$

For the input signal of

$$w(t) = v_0 \cdot t \longleftrightarrow \frac{v_0}{s^2}. \tag{20}$$

the system response results in

$$\hat{x} = \frac{v_0(K_I + K_P s + s)}{s^2(K_I + s(K_P + sT + 1))} \tag{21}$$

which represents the time domain signal

$$\begin{align} x(t) &= \frac{v\left(\left(T - Te^{\frac{rt}{T}}\right) e^{-\frac{t(K_P + r + 1)}{2T}} + rt\right)}{r} \tag{22} \\ \text{with } r &= \sqrt{(K_P + 1)^2 - 4K_I T}. \tag{23} \end{align}$$

This signal shows the desired convergence, as shown in Fig.3 for the chosen parametrization. Furthermore Fig.3 shows, that higher K_P leads to smaller tracking errors, but convergence for $K_P < \infty$ can be only achieved with an integral part $K_I > 0$. Due to the underlying *MPC* and the three dimensional problem, where x, y, z are coupled, the controller

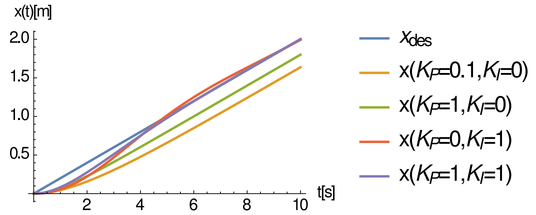


Fig. 6: System responses for different parametrization for $T = 2$s and $v_0 = 0.2 \mathrm{m\,s}^{-1}$

cannot be parametrized by the typical setting of poles. One reason is, that the integral part leads to overshooting in the quadrotor trajectory for direction changes. Accordingly the integral part K_I is chosen to be very small, while the convergence is accelerated by using high proportional gain K_P. Due to the stability issues of the internal *MPC* controller, K_P cannot be chosen arbitrarily high. For the considered quadrotor/*MPC* combination of section II, the parameters have been empirically chosen for the x-channel to $K_p = 2$ and $K_I = 0.001$. The simulation with the target position controller is shown in 7. As expected, the tracking error d is converging to zero. The next step is the transfer to the real system.

IV. Experimental validation

The experiment is consisting of an *AR.Drone 2.0* controlled by the inner *MPC* control of section II and the outer target position control of section III. The position of the quadrotor is measured with a motion capture system, which limits the experimental space. For this reason, the desired trajectory is chosen as square with a side length of $a = 1.5$m in an altitude of $z_{des} = 1.5$m. The target position is changed with a constant velocity of $v_0 = 0.2 \mathrm{m\,s}^{-1}$. Fig.8 is visualizing the resulting trajectory without the proposed target position controller on the left side. It can be seen, that the desired square trajectory is not reached, but altered to an ellipsoid. The corresponding system trajectories are shown in Fig.9, where x and y position show a delay towards x_{des} and y_{des} respectively. The high resulting tracking error with $d \approx 2$m contains alterations that are caused by the direction changes in the square corners.

To reduce the tracking error d, the proposed target position controller is implemented. As the prediction model (3) shows asymmetric model parameters for x and y channel, the target position controller is adapted to each channel individually. The parameters for the x-channel are empirically chosen to $K_{P,x} = 2$, $K_{I,x} = 0.001$ and for the y-channel $K_{P,y} = 3$, $K_{I,y} = 0.001$. The resulting trajectory with target position control in Fig.8 (right) is showing a square movement as desired. In the corners of the square, the integral part leads to overshooting. After the corner, the quadrotor converges towards the square edges. The corresponding trajectory shows a much lower tracking error d, as given in Fig.10. This is caused by the smaller delay in x and y positions. At each corner the tracking error is increased due to the direction change of the target trajectory. After the corner, it is converging against zero as desired. The experiment is validating the desired target position controller for constantly moving targets.

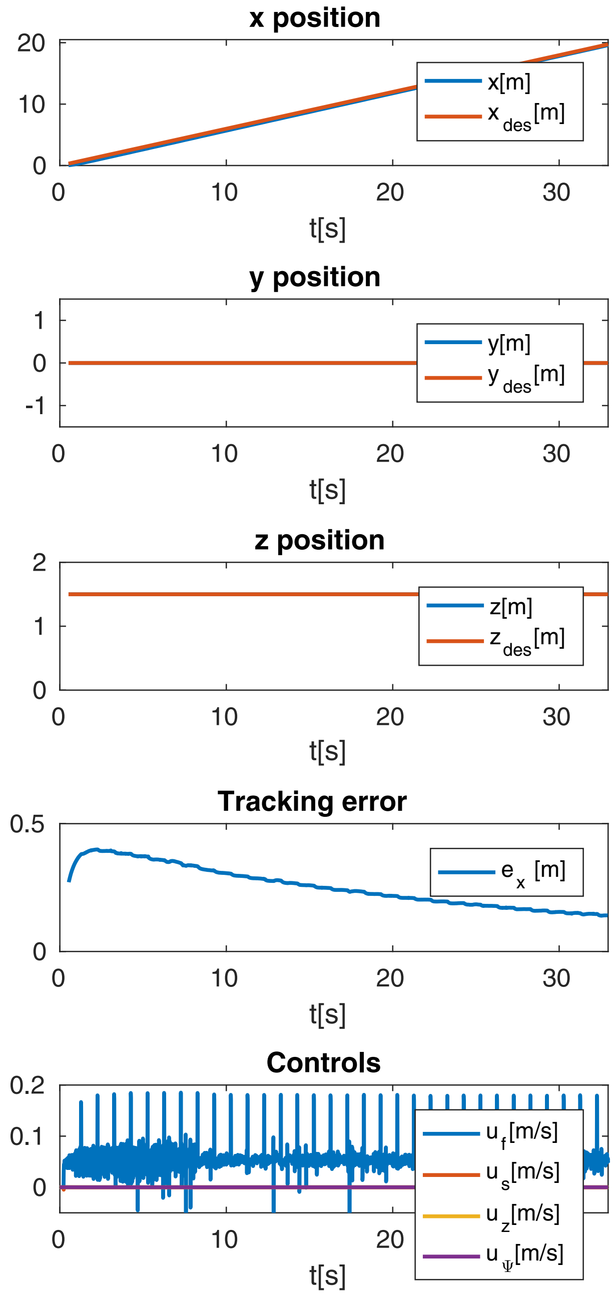
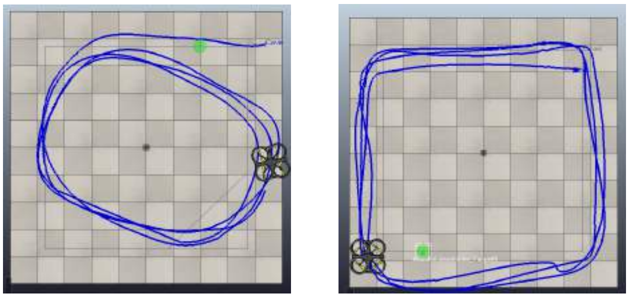


Fig. 7: Simulation with adaptive target control



(a) Without *TPC*

(b) With *TPC*

Fig. 8: Visualization of the experimental data of tracking of a target, moving in a square with a constant velocity

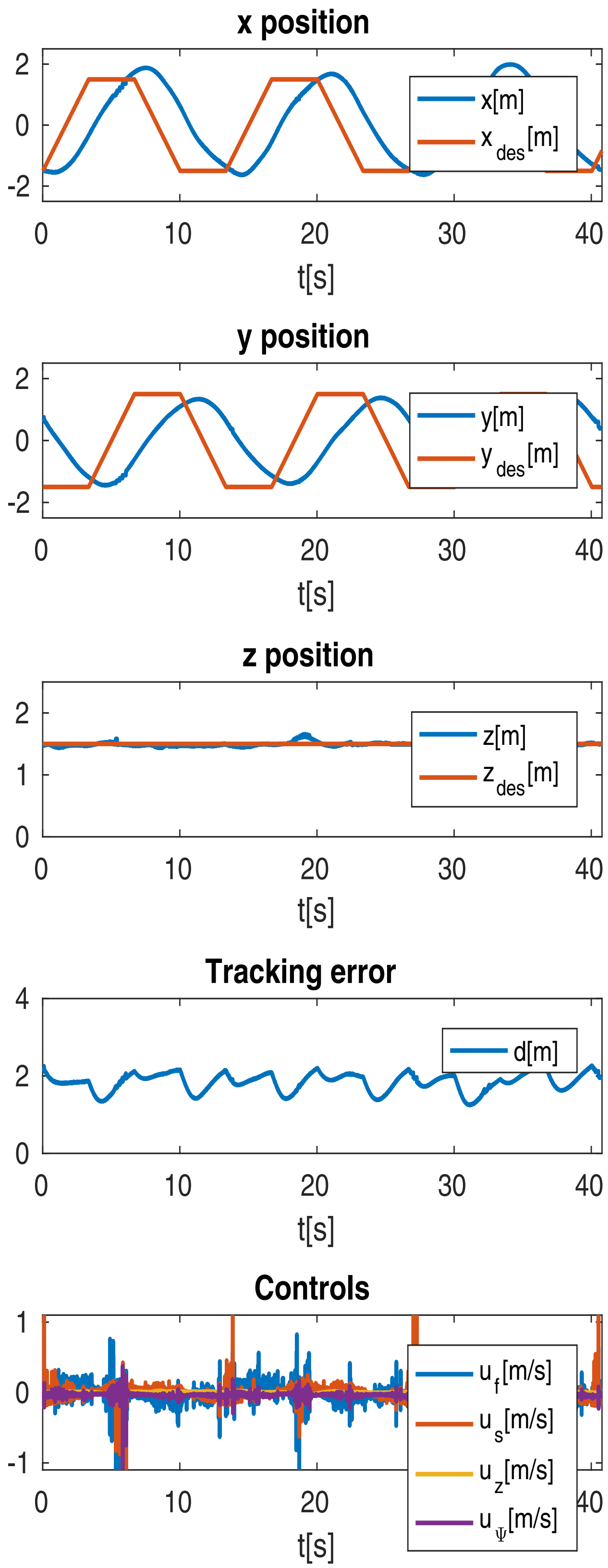


Fig. 9: Real AR.Drone 2 square tracking without target position control

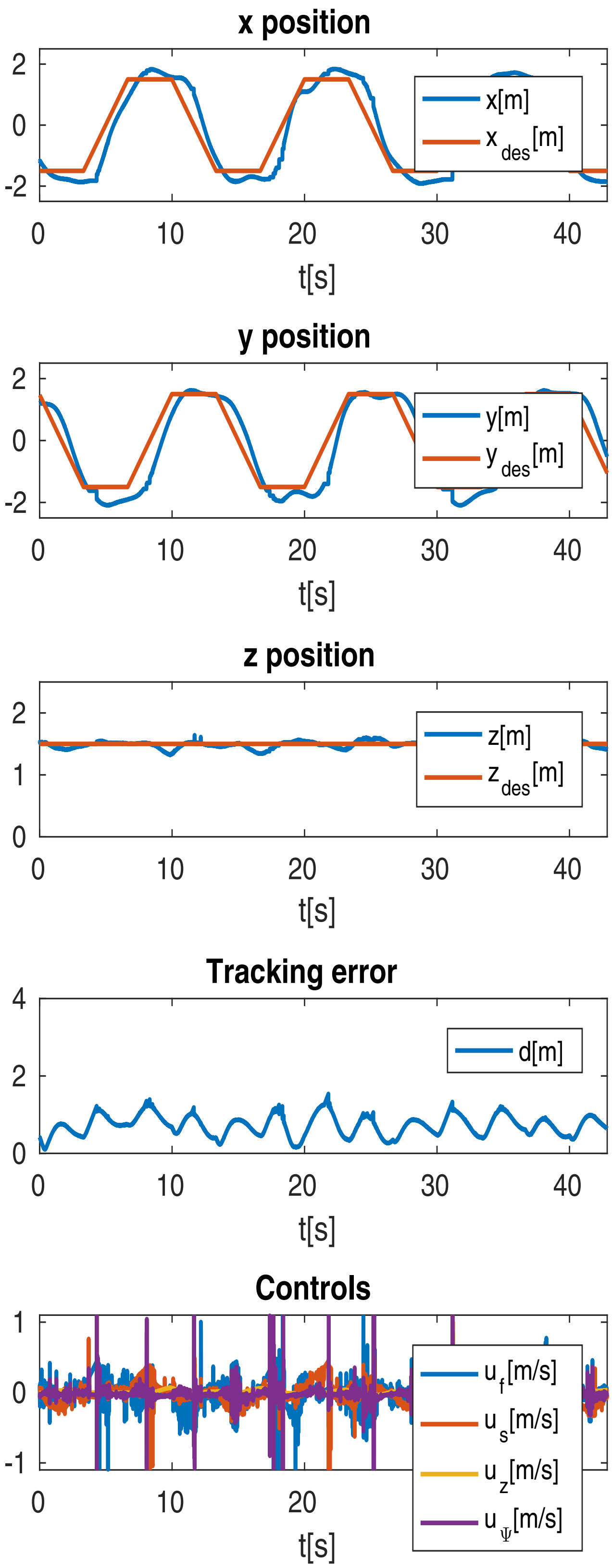


Fig. 10: Real AR.Drone 2 square tracking with target position control

V. CONCLUSIONS

This paper is presenting a target position control, which reduces the tracking error of *MPC* controlled systems with constantly moving targets. The representation of this problem within this work is a quadrotor tracking scenario. To solve this problem, the closed-loop system has been considered as PT_1 system. By means of Laplace transformation it has been shown, that this modelling approach for the closed loop system is justified for the considered problem. The constantly moving reference position leads to a convergence of the tracking error towards a constant value. The expected behaviour is validated in a simulation. To reduce the tracking error, a target position controller is added as outer loop to the *MPC* controlled system. It is based on feeding the *MPC* with targets that appear to be further away than actually given. This distance is related via a *PI*-controller with the tracking error. The proposed control approach has been validated by means of Laplace transformation and simulative results. Additionally, the controller tuning regarding the shown scenario has been discussed. The last contribution of the paper is the experimental comparison with and without target position control. The result shows that the proposed approach leads to a better trajectory tracking, indicated by a reduced tracking error and the resulting trajectory form. The presented controller requires low computational effort in comparison to other approaches that include e.g. disturbance model estimation. Furthermore the parameter tuning is intuitive, does not require *MPC* internal changes and does not affect the constraint handling of the *MPC*.

In future work, this simple control approach can be extended to more complex controllers, e.g. an overlying *MPC* to provide better handling of complex trajectories.

REFERENCES

[1] G. T. U. R. Facility, "Underwater quadrotor flight test," https://www.youtube.com/watch?v=U7vl7uqwN4I, 2015-11-13, accessed: 2016-10-14.

[2] P. L. J. Drews, A. A. Neto, and M. F. M. Campos, "Hybrid unmanned aerial underwater vehicle: Modeling and simulation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4637–4642.

[3] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp. 2247–2252.

[4] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2007, pp. 153–158.

[5] D. Ariens, M. Diehl, H. Joachim, B. Houska, Ferreau, F. Logist, R. Quirynen, and M. Vukov, "ACADO Toolkit User's Manual," 2015, accessed: 2015-09-11.

[6] K. Graichen and B. Käpernick, "A real-time gradient method for nonlinear model predictive control," 2012.

[7] "A continuation/gmres method for fast computation of nonlinear receding horizon control," *Automatica*, vol. 40, no. 4, pp. 563 – 574, 2004.

[8] J. Dentler, S. Kannan, M. A. Olivares-Mendez, and H. Voos, "A real-time model predictive position control with collision avoidance for commercial low-cost quadrotors," in *2016 IEEE Multi-Conference on Systems and Control (MSC), Buenos Aires*, September 2016.

[9] J. Dentler, S. Kannan, M. A. Olivares-Mendez, and H. Voos, "A modularization approach for nonlinear model predictive control of distributed fast systems," in *2016 24th Mediterranean Conference on Control and Automation (MED)*, 2016, pp. 292–297.

[10] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "{MPC} with nonlinear control for path tracking of a quad-rotor helicopter," *{IFAC} Proceedings Volumes*, vol. 41, no. 2, pp. 8564 – 8569, 2008, 17th {IFAC} World Congress. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016403277

[11] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/nonlinear control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29 – 39, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109809004798

[12] K. Alexis, G. Nikolakopoulos, and A. Tzes, "Experimental model predictive attitude tracking control of a quadrotor helicopter subject to wind-gusts," in *Control Automation (MED), 2010 18th Mediterranean Conference on*, 2010, pp. 1461–1466.

[13] R. V. Lopes, P. H. D. R. Quemel, A. Santana, A. Borges, and Y. Ishihara, "Model predictive control applied to tracking and attitude stabilization of a vtol quadrotor aircraft," 2011.

[14] G. Pannocchia, "Offset-free tracking mpc: A tutorial review and comparison of different formulations," in *Control Conference (ECC), 2015 European*, 2015, pp. 527–532.