

Adaptive Model Predictive Control of a Quadrotor

Piwei N. Chikasha* Chioniso Dube**

* *University of the Witwatersrand, Johannesburg, Wits 2050 South Africa (e-mail: 481274.students@wits.ac.za)*

** *University of the Witwatersrand, Johannesburg, Wits 2050 South Africa (e-mail: Chioniso.Kuchwa-Dube@wits.ac.za)*

Abstract: This work explores the application of Adaptive Model Predictive Control (AMPC) to quadrotor altitude control. Model Predictive Control (MPC) is a very powerful method of Advanced Control, utilizing an implicit model to make output predictions which are in turn used in computing control action. Dynamic Matrix Control (DMC) was the first MPC technique to be implemented, and is still very common in process control especially for Chemical plants. Due to changes in operating points, the implicit model in MPC typically becomes insufficient, costing the quality of the controller. This work proposes Adaptation (based on the Recursive Least Squares (RLS) algorithm), for online system identification to take changes in operating points into account when computing the DMC control action. Overall, Adaptive DMC is investigated as to whether Adaptive DMC is capable of improved quadrotor control or tracking.

© 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Model Predictive Control, Dynamic Matrix Control, Adaptive Control, Recursive Least Squares, Forgetting Factor.

1. INTRODUCTION

This research proposes Adaptive Model Predictive Control (AMPC) for altitude control of a quadrotor. Many different controllers have been tested for the control of quadrotors. Whether or not a controller works desirably for quadrotor control normally depends on the system model of the quadrotor (Schreier, 2012) and the environment the quadrotor will operate in.

Some of the aspects that make quadrotor control demanding are that the quadrotor is subject to unknown wind disturbances and possibly, uncertain dynamics influenced by misalignment of motors, weight imbalance etc, and physical limitations in velocity, attitude angles, rotor angular velocity etc. A controller designed for one operation condition of the quadrotor is prone to performance degradation as the environment and/or vehicle dynamics change (Wang et al., 2016).

On the other hand, Adaptive control allows adjustment of control of the quadrotor by automatically updating the system model (through parameter estimation), online or offline, according to the prevailing process dynamics. Work such as that of Landau and team (Landau et al., 2011 and Osa et al., 2001) shows that adaptive control has gained much ground in flight control systems and has become a topic of interest in cases where plant dynamics or disturbances are unknown or varying. Model predictive control (MPC) is an optimal control technique employing an internal model to predict system behaviour in some finite time horizon (Abdolhosseini et al., 2012 and Kim

et al., 2012). Two common types of MPC are Dynamic Matrix Control (DMC) and Generalised Predictive Control (GPC). The differences between the two are rather subtle, but important to note is that DMC, which was developed and introduced by Charles Cutler in the 1970s. (Cutler et al., Guiamba, 2001), uses a step response model, and puts relatively more emphasis on past input data to make predictions, than GPC does (Rossiter, 2003). DMC also, because it is non-parametric (utilising a step response model with 60 data points for example) is less sensitive to noise, although a little more computationally expensive compared to the parametric GPC. For the advantage of noise sensitivity, the type of MPC investigated in this work is DMC, leading to Adaptive Dynamic Matrix Control (ADMC). It should be note though that, methods do exist, to make GPC less sensitive to noise (for example implementing filters). MPC has been successful for quadrotor control (Alexis et al., 2014 and Kunz et al., 2013) achieving high quality performance and on the other hand, adaptive control (Li et al., 2014 and Monte et al., 2013). AMPC, in the form of GPC, has also been found to meet high performance indices for quadrotor trajectory tracking (Bouffard et al., 2012).

A common form of ADMC is one in which adaptation is used to calculate parameters of the DMC controller such as the prediction or control horizon, depending on the estimated real-time system dynamics. In the work of Klopot, this is achieved by using linear spline interpolation, for some nonlinear hydraulic plant (Klopot et al., 2015). High quality system performance is achieved for process control by updating the parameters; suppression and scaling factor, of the DMC (Posada et al., 2008). Another realisation of ADMC is to update the DMC model

* This work is based on the research supported in part by the National Research Foundation of South Africa for the grant, Unique Grant No. 107025.

as the system moves from one operating point to another. This has been investigated for a distillation column (Maiti et al., 1995) and also for a turbine system (Lee et al, 2003). Another realisation is to have several models of the system, for different points, then, based on system measurements, use interpolation to find the best model for a particular operating point. This has been successfully implemented for process control, (Chen et al., 2009 and Demircan et al., 1999). Other forms of ADMC have been implemented and are in most ways, extensions of the above stated. Extra caution must however be taken with ADMC because a variable prediction model typically results in poor system performance, especially for low order systems, (Demircan et al., 1999) due to the switching. It is therefore common practice to set up ADMC such that all models continue to calculate the dynamic matrix although only the activated model is considered for calculating control action at each sample time. This in turn, however, raises computational cost.

Since the majority of the work done on ADMC does not take computational expense and implementation (especially for such systems as multi-rotors) into account, this work looks into ADMC from theory to implementation. As such, with the intention to improve controller efficiency, this work proposes the use of a single step response model for different disturbance influence, hence limiting switching. This follows several simulations to analyse how model parameters are affected by disturbance. This is supported by a Supervisor which monitors system behaviour and then decides when to switch to or from the disturbance step response model. The expected profits of using a supervisor over interpolation are code simplicity (hence more efficiency) and, since a RLS estimator already exists, which estimates model parameters in real time, the RLS output can conveniently and readily be used by the supervisor in decision making. The mathematical model is presented in Section 2 while Section 3 explores the derivation and implementation of ADMC and Section 4 presents simulation results. Section 5 concludes.

2. MATHEMATICAL MODEL

2.1 Analytical Model

AMPC is based on an internal model, which is updated in real-time according to adaptive laws and is used to predict system output. The quadrotor is modelled analytically, based on dynamic and kinematic equations (Newton-Euler formalisation) as in (Balas, 2007, Ahmed et al., 2015 and Abaunza et al., 2016). This considers the quadrotor as the moving (body) frame ($x_B - y_B - z_B$) and the earth as the inertial frame ($x_E - y_E - z_E$) as summarised by Fig. 1 and for this work, according to the parameters given in table 1.

Table 1. Model parameters

Parameter	Unit	Symbol	Value
Mass	kg	m	0.55
Rotor wheelbase	m	l	0.23
Drag factor	–	d	$7.5 * 10^{-3}$
Thrust factor	–	b	$3.13 * 10^{-3}$

The notation ω_i denotes angular velocity of the i th motor and τ_{Mi} denotes torque produced by the i th motor. The

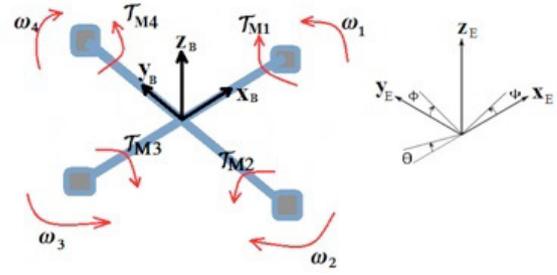


Fig. 1. Quadrotor inertial and body frames

equation (1) and equation (2) represent the resultant analytical model in terms of linear and angular accelerations respectively:

$$\begin{cases} \ddot{x} = -(c\psi s\theta c\phi + s\psi s\phi) \frac{u_1}{m} \\ \ddot{y} = -(s\psi s\theta c\phi - c\psi s\phi) \frac{u_1}{m} \\ \ddot{z} = -(c\theta c\phi) \frac{u_1}{m} + g \end{cases} \quad (1)$$

and

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & \dot{\phi} c\phi t\theta + \dot{\theta} \frac{s\phi}{c^2\theta} & \dot{\phi} s\phi c\theta + \dot{\theta} \frac{c\phi}{c^2\theta} \\ 0 & -\dot{\phi} s\phi & -\dot{\phi} c\phi \\ 0 & \dot{\phi} \frac{c\phi}{c\theta} + \dot{\phi} s\phi \frac{t\theta}{c\theta} & -\dot{\phi} \frac{s\phi}{c\theta} + \dot{\theta} c\phi \frac{t\theta}{c\theta} \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \cdot \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \quad (2)$$

where u_1 is total thrust force, g is acceleration due to gravity, s , c and t are trigonometric functions sine, cosine and tangent respectively. ϕ , θ and ψ are the roll, pitch and yaw attitude angles of the quadrotor (Euler angles), respectively and x , y and z represent 3 the dimensional position of the quadrotor, all expressed in the reference (stationary frame) i.e. Earth frame. The roll, pitch and yaw rotational angular velocities are denoted as p , q and r for roll, pitch and yaw respectively and expressed in the quadrotor frame.

2.2 System Identification

System identification is necessary for this work because this results in a relatively reliable model, representing accurate system dynamics. The identified ARX model is used to determine the regressors for parameter estimation. This parameter estimation, as stated earlier, is based on the Recursive Least Squares (RLS) method, a method well known for convenient implementation with the discrete-time ARX model type (Landau et al., 2011). The analytical model available is therefore run in MATLAB for step input and the resultant input-output data is collected and stored. Prior to this simulation, the model is stabilised by PID control using Simulink PID Tuner, because DMC works when the internal step response model is stable. Using the System Identification tool in MATLAB, the collected

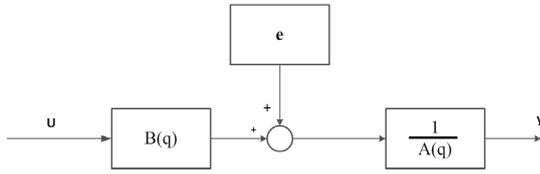


Fig. 2. ARX model structure

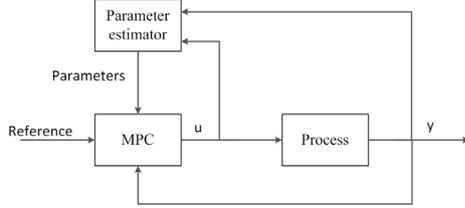


Fig. 3. Direct adaptive control scheme

data is processed for ARX model identification and the identified ARX model is optimised using the Response Optimisation tool to minimise error. The discrete-time ARX model, is of the form:

$$A(z)y(t) = B(z)u(t) + e(t) \quad (3)$$

where $A(z)$ and $B(z)$ are backward shift operator (z^{-1}) polynomials (considering zero input delay for simplification) as follows:

$$A(z) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{n_a}z^{-n_a} \quad (4)$$

$$B(z) = b_1z^{-1} + b_2z^{-2} + \dots + b_{n_b}z^{-n_b} \quad (5)$$

with the backward shift operator such that $z^{-n}u(k) = u(k - n)$ (Bhuvaneshwari et al., 2012 and Andersson et al., 1998). The ARX model can be presented by such an illustration as Fig. 2 where e represents random zero-mean noise.

The system identification and optimisation processes produce a second order ARX model:

$$A(z) = 1 - 1.811z^{-1} + 0.8147z^{-2} \quad (6)$$

$$B(z) = 0.0044z^{-1} \quad (7)$$

from the recorded input-output data. This model is successfully verified by simulating ramp reference input then comparing the ARX response against that of the initial analytical model.

3. AMPC DERIVATION AND IMPLEMENTATION

3.1 Adaptive mechanism

Fig. 3 shows the general direct adaptive control scheme.

The signal u represents control action and y represents the system output. Using these two signals, adaptive laws estimate parameters of the system model (within the parameter estimator block) in real-time, the model for which control is designed by updating the controller parameters. In this way, the reason adaptivity is needed is that adaptivity tracks changes in system dynamics (resulting from wind disturbance or payload attachment) and allows the controller to adjust for the changes. The

RLS principle is used based on the Regressor Form as in (8) (Landau et al., 2012 and Osa et al., 2001) where $\varphi^T = [\varphi_1, \varphi_2 \dots \varphi_n]^T$ - regressor functions - and $\theta = [\theta_1, \theta_2 \dots \theta_n]$ - model parameters to be estimated.

$$y(i) = \varphi^T(i)\theta \quad (8)$$

The RLS parameter estimation algorithm can be summarized by the equations (9), (10) and (11), (Landau et al., 2012 and Guo et al., 1993):

$$\hat{\theta}(t+1) = \hat{\theta}(t) + P(t+1)\varphi(t)\varepsilon(t+1) \quad (9)$$

where

$$P(t+1) = P(t) - \frac{P(t)\varphi(t)\varphi^T(t)P(t)}{1 + \varphi^T(t)P(t)\varphi(t)} \quad (10)$$

and

$$\varepsilon(t+1) = y(t+1) - \hat{\theta}^T(t)\varphi(t) \quad (11)$$

The solution for the estimated parameters comes from minimising a cost function, which is expressed in terms of the Forgetting Factor (λ) (Guo et al., 1993):

$$V(\hat{\theta}, k) = \frac{1}{2} \sum_{i=1}^k \lambda^{k-i} [y(k) - \varphi^T(k)\hat{\theta}(k)]^2 \quad (12)$$

The idea is that when λ is close to 1, the result is good convergence and small variances of the estimates; while when λ is closer to zero, the result is good tracking (quick forgetting of old data). The parameters of adaptive control that are tuned are the Forgetting Factor and Starting (Initial) Values. $P(0)$ represents some estimate of the covariance matrix of the initial parameters. Small $P(0)$ results in small changes of the estimated parameters. On the other hand, large $P(0)$ results in quick a rapid move from initial parameters to the estimates (Soderstrom et al., 1989). Typically, for the practical case:

$$\hat{\theta}(0) = 0 \quad (13)$$

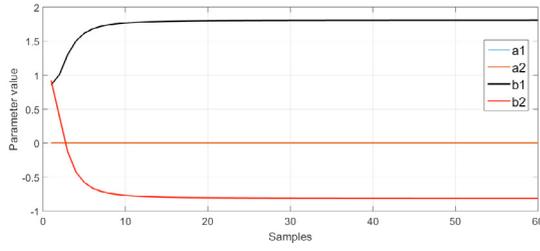
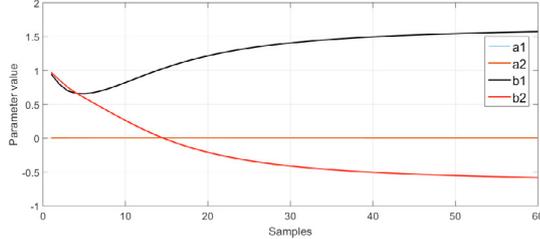
and

$$P(0) = \rho \mathbf{I} \quad (14)$$

where ρ is a constant and \mathbf{I} is a unitary identity matrix. Simulations are run to determine suitable parameters for tuning the estimator. The RLS estimator requires persistent excitation, i.e. input signal capable of exposing the characteristics of the system, in this case, a pulse width Modulated signal. After several simulations, the estimator is tuned to $\lambda = 0.97$, $\rho = 1e^6$ and initial estimates = 1 for the poles (a_1 and a_2 parameters). This setting brings about satisfactory convergence as shown in Fig. 4 when a step altitude increase of 1 meter is input at hover.

To quickly demonstrate the effect of the tuning parameters, the simulation with $\rho = 1e^4$ is run, which is expected to give slower convergence, as is the case shown in Fig. 5.

The behaviour of the parameters of the model is analysed for varying dynamics i.e. with different disturbances. It is observed that, while the poles of the system vary in magnitude as system dynamics change, this variation is consistent even for low magnitude disturbance input. One reason for this is the low order of the system. This

Fig. 4. RLS estimation $\lambda = 0.97$, $\rho = 1e6$ Fig. 5. RLS estimation $\lambda = 0.97$, $\rho = 1e4$

makes the use of a Supervisor for adaptation profitable as explained in Section 4.

3.2 Model Predictive Control - DMC

DMC utilises two main components, i.e. the model and the optimisation solver to minimise such a cost function as (15), (k is the sampling instant).

$$J = \sum_{i=1}^p (y^{ref}(k+i) - \hat{y}(k+i))^2 + r \sum_{i=0}^{c-1} \Delta u(k+i)^2 \quad (15)$$

- y^{ref} is the reference trajectory.
- y^{sp} is the set-point.
- $\hat{y}(k)$ is the predicted output at instant k .
- $\Delta u(k)$ is the control move at instant k and $\Delta u(k) = u(k) - u(k-1)$.
- p is the prediction horizon, c is the control horizon.
- r is a weighting factor.

Signal y^{ref} , which is typically an exponential rise, is the reference trajectory to move from measured output to the set-point. The expression for y^{ref} can be manipulated to tune the rate at which the output approaches the set-point, by introducing the scaling factor α (Lopez-Guede et al., 2013 and Rani et al., 2014):

$$y^{ref}(t+k) = \alpha y^{ref}(t+k-1) + (1-\alpha)y^{sp}(t+k) \quad (16)$$

α is a value between zero and one and technically (from the expression of y^{ref}), should result in aggressive response when closer to zero and sluggish response when closer to one. The DMC Prediction of output can be written as (17)

$$\underbrace{\hat{y}(k+1)}_{\text{=prediction}} = \underbrace{\mathbf{T}\mathbf{f}(k)}_{\text{=past}} + \underbrace{\mathbf{s}^d \Delta d(k)}_{\text{=present}} + \underbrace{\mathbf{G}\Delta \mathbf{u}(k)}_{\text{=future}} \quad (17)$$

where \mathbf{G} is the dynamic matrix, and is derived from a step model of the system due to step input (s^u) and (17) is in fact summary to (19) and also, (17) simplifies to (18) (Lopez-Guede et al., 2013 and Rani et al., 2014).

$$\hat{y}(k+1) = \mathbf{y}^p(k+1) + \mathbf{G}\Delta \mathbf{u}(k) \quad (18)$$

$$\begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+p|k) \end{bmatrix} = \begin{bmatrix} f(k+1|k) \\ f(k+2|k) \\ \vdots \\ f(k+p|k) \end{bmatrix} + \begin{bmatrix} s^u(1) & 0 & \cdots & 0 \\ s^u(2) & s^u(1) & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ s^u(p) & s^u(p-1) & \cdots & s^u(1) \end{bmatrix} \cdot \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+p-1|k) \end{bmatrix} + \begin{bmatrix} s^d(1) \\ s^d(2) \\ \vdots \\ s^d(p) \end{bmatrix} \cdot \Delta d(k|k) + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \cdot (y(k) - f(k|k)) \end{bmatrix} \quad (19)$$

The term \mathbf{y}^p where $(\mathbf{y}^p(k+1) = [y^p(k+1), y^p(k+2), \dots, y^p(k+p)]^T)$ represents the past free response term and the present feed-forward and feedback terms. $\Delta \mathbf{u}$ denotes change in control action. The term $\mathbf{G}\Delta \mathbf{u}$ where \mathbf{G} is of $p \times c$ dimension represents the part due to the future control moves. From here, the sequence of computed control action is derived from (20)

$$\Delta \mathbf{u} = \underbrace{[\mathbf{G}^T \mathbf{G} + r\mathbf{I}]^{-1} \mathbf{G}^T}_{\mathbf{H}} \underbrace{(\mathbf{y}^{sp} - \mathbf{y}^p)}_{\mathbf{e}} = \mathbf{H}\mathbf{e} \quad (20)$$

The control horizon c is selected and held constant early in the controller design while tuning other controller settings. Keeping c small means fewer variables need to be calculated at each control interval. This in turn, allows faster computation. Typically, tuning the model predictive controller is not standardised but rather more intuitive.

4. SIMULATION RESULTS

It was seen in Section 3.1 that the model parameters change in magnitude with varying system dynamics. ADMC is therefore realised by monitoring parameter values using the supervisor and then switching models based on value of the parameters (poles in this case, as the system zeros remain relatively unchanged) when disturbance acts on the system. This is illustrated by Fig. 6

Simulations are run to investigate the quality of control provided by ADMC (with a 0.025 seconds sampling time) leading to a conclusion as to how much ADMC improves quadrotor control. Two aspects are tested, reference tracking and disturbance rejection. The ADMC is compared

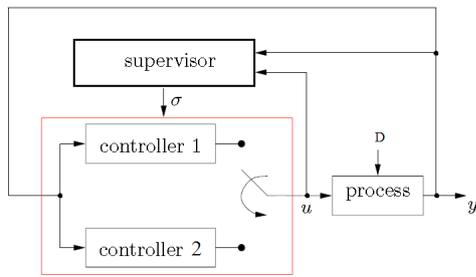


Fig. 6. General supervisory scheme

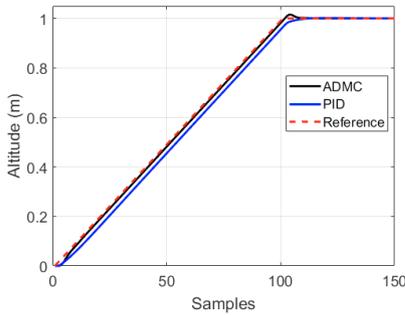


Fig. 7. PID and ADCM response - ramp input

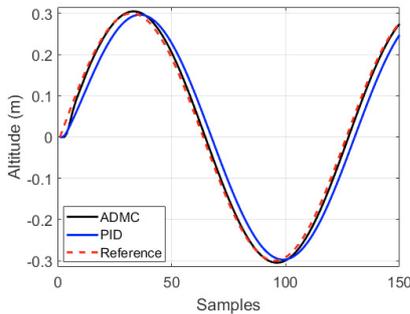


Fig. 8. PID and ADCM response - sinusoidal input

against the PID controller introduced earlier and a DMC only controller. The parameters used for the initial tuning of the predictive controller are $p = 5$ and $c = 5$. Prior simulations show that a shorter prediction horizon, in this case, leads to better performance, which is why p is set to a rather low value of 5. This is not entirely unusual, Lopez and team make a solid conclusion in their work that, in reality, a low and intermediate value of p results in a better performance of DMC (Lopez-Guede et al., 2013).

The first investigation is with respect to reference tracking without disturbance (therefore with deactivated adaptation in ADCM) and is summarised by Fig. 7 and Fig. 8 for two different reference trajectories.

It can be concluded that ADCM archives superior reference tracking, especially even as the reference trajectory becomes more complex. As for disturbance rejection, the three controllers i.e. PID, DMC and ADCM are investigated for step disturbance, which can be modelled as a change in sensor output or as a force, as in this work. While at hover, disturbance is applied to the system as illustrated by Fig. 9.

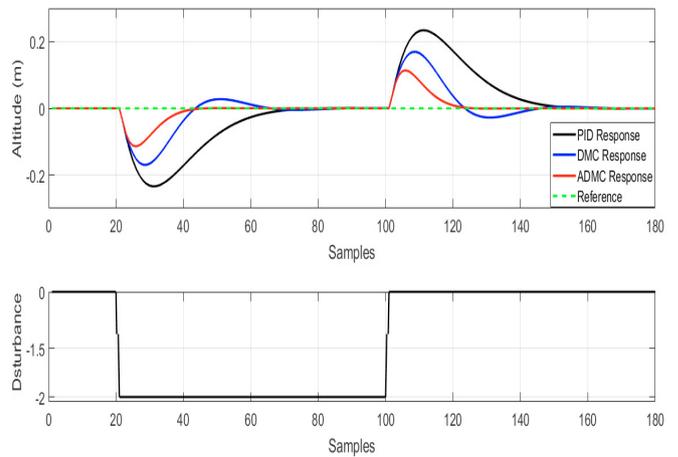


Fig. 9. PID, DMC and ADCM response - step disturbance

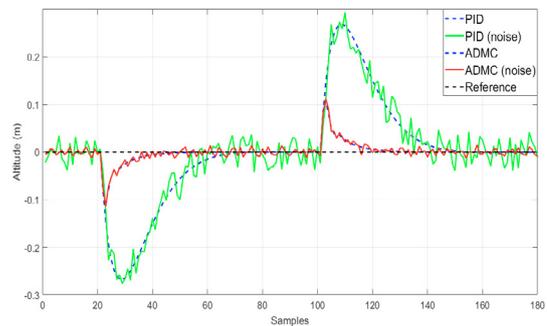


Fig. 10. PID and ADCM noise response

The DMC controller is more rapid in disturbance compensation than PID, but not as rapid as the ADCM controller. Overall, as expected, ADCM provides more rapid and more precise reference tracking and disturbance rejection. Considering a more practical case, with measurement noise in sensor data, ADCM outperforms the PID again, as expected. Simulation shows that the PID response is affected more by noise than ADCM, as shown in Fig. 10 where vibrations occur with greater amplitude for PID. One reason for this is that the derivative action of the PID gains high influence when the measured output fluctuates rapidly as this translates to a steep slope. This problem however, can be resolved by the use of a low-pass filter. Another desirable feature of ADCM is that once set up, ADCM is easy to tune.

Implementation of ADCM however, cannot be done without considering sampling time of the controller. Effects of sampling time depend on the capacity of the computer used. Investigations carried out in this work (using an i5 processor and 4 Gigabytes of RAM) with 0.1 and 0.0001 seconds sampling times reveal that too large a sampling interval prevents adequate capture of the plant dynamics leading to instability, while too small a sampling interval fails to allow for the full computation of the algorithm, again leading to poor performance.

5. CONCLUSION

ADCM is capable of providing the expected improved quadrotor control. In this context, ADCM has been vali-

dated with respect to reference tracking and disturbance rejection, where in both cases, ADMC surpasses the classical PID controller. Tuning the ADMC is quite an intuitive process, one which demands extensive simulation. When tuned well however, the ADMC controller satisfies the two main objectives i.e. adaptation to varying plant dynamics and optimal reference tracking. While DMC is the oldest form of MPC, DMC remains a powerful control tool and further research into implementing or extending DMC for quadrotor control is worthwhile.

REFERENCES

- Abaunza H., Castillo P., Lozano R., and Victorino A. (2016). Quadrotor aerial manipulator based on dual quaternions. *International Conference on Unmanned Aircraft Systems*, 125–247.
- Abdolhosseini M., Zhang Y.M. and Rabbath C.A. (2012). An Efficient Model Predictive Control Scheme for an Unmanned Quadrotor Helicopter. *J Intell Robot Syst*, volume 70, 27–38.
- Ahmed S.F., Kushsairy K, Bakar M.I.A., Hazry D. and Joyo M.K. (2015). Attitude stabilization of Quadrotor (UAV) system using Fuzzy PID controller (an experimental test). *2nd International Conference on Computing Technology and Information Management*, 99–104.
- Alexis K., Nikolakopoulos G. and Tzes A. (2014). On Trajectory Tracking Model Predictive Control of an Unmanned Quadrotor Helicopter subject to aerodynamic disturbances. *Asian Journal of Control* volume 16, no. 1, 209–224.
- Andersson L., Jonsson U., Johansson K.H. and Bengtsson J. (1998). A manual for system identification. Laboratory Exercises in System Identification. *Department of Automatic Control, Lund Institute of Technology*, volume 118.
- Balas C. (2007). *Modelling and Linear Control of a Quadrotor*. MSc Thesis, Cranfield University, UK.
- Bhuvaneshwari N.S., Praveena R. and Divya R. (2012). System identification and modeling for interacting and non-interacting tank systems using intelligent techniques. Easwari Engineering College, Chennai, Tamilnadu, India.
- Bouffard P., Aswani A. and Tomlin C. (2012). *Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results*. *IEEE International Conference on Robotics and Automation*, 79–284.
- Chen X., Li S., Zhai J. and Li Q. (2009). Expert System Based Adaptive Dynamic Matrix Control for Ball Mill Grinding Circuit. *Expert Syst. Appl.*, volume 36, 716–723.
- Cutler C.R., Ramaker B.L. (1979). Dynamic Matrix Control - A Computer Control Algorithm. *American Institute of Chemical Engineers (AIChE) National Meeting*.
- Demircan M. Camurdan M.C. and Postlethwaite B.E. (1999). On-Line Learning Fuzzy Relational Model Based Dynamic Matrix Control of an Openloop Unstable Process. *Chemical Engineering Research and Design*, volume 36, no. 5, 421–428.
- Guiamba I.R.F. (2001). *Adaptive dynamic matrix control for a multivariable training plant*. MSc Thesis
- Guo L., Ljung L., and Priouret P. (1993). Performance analysis of the Forgetting Factor RLS algorithm. *International Journal of Adaptive Control and Signal processing*, volume 7, 525–537.
- Kim B.A., Lee S.H., Lee Y.O. and Chung C.C. (2012). Comparative study of approximate, proximate, and fast model predictive control with applications to autonomous vehicles. *12th International Conference on Control, Automation and Systems*, 479–484.
- Klopot T. and Skupin P. (2015). Adaptive dynamic matrix control with interpolated parameters. *20th International Conference on Methods and Models in Automation and Robotics (MMAR)*, 683–688.
- Kunz K., Huck S.M. and Summers T.H. (2013). Fast Model Predictive Control of miniature helicopters. *European Control Conference*, 1377–1382.
- Landau I., Lozano D., Karimi A. and MSaad M. (2011). *Adaptive Control*. Springer, London.
- Lee K.Y. and Moon U.C. (2003). A boiler-turbine system control using a fuzzy auto-regressive moving average (FARMA) model. *IEEE T. Energy Convers.*, volume 18, 142–148.
- Li S., Li B. and Geng Q. (2014). Adaptive sliding mode control for quadrotor helicopters. *33rd Chinese Control Conference*, 71–76.
- Lopez-Guede J., Fernandez-Gauna B., Grana M. and Oterino F. (2013) Model Predictive Control, Dynamic Matrix Control, Prediction Horizon. *International Journal of Control Science and Engineering*.
- Maiti S.N. and Saraf D.N. and Kapoor N. (1995). Adaptive dynamic matrix control of a distillation column with closed-loop online identification. *Journal of Process Control*, volume 5, 315–327.
- Monte P.D. and Lohmann B. (2013). Position trajectory tracking of a quadrotor helicopter based on L1 adaptive control. *European Control Conference*, volume 18, 3346–3353.
- Osa Y., Mabuchi T. and Uchikado S. (2001). Synthesis of discrete time adaptive flight control system using nonlinear model matching. *IEEE International Symposium on Industrial Electronics*, volume 1, 58–63.
- Posada J. and Sanjuan M. (2008). Intelligent adaptive dynamic matrix control. *Proceedings of the 17th World Congress, The International Federation of Automatic Control*, Seoul, Korea.
- Rani L., Sivakumar D. and Rathikarani D. (2017) Dynamic Matrix Controller Design for Performance Study of an Interacting Coupled Tank MIMO Process. *AIP Conference Proceedings*, volume 1859.
- Rossiter J. A. (2003). *Model-Based Predictive Control: A Practical Approach*. CRC Press.
- Schreier M. (2012) Modeling and adaptive control of a quadrotor. *IEEE International Conference on Mechatronics and Automation*, 383–390.
- Soderstrom T. and Stoica P. (1989). *System Identification*. Prentice Hall International.
- Wang C., Song B., Huang P. and Tang C. (2016). Trajectory Tracking Control for Quadrotor Robot Subject to Payload Variation and Wind Gust Disturbance. *J Intell Robot Syst*, volume 83, no. 2, 315–333.