Event driven intelligent PID controllers with applications to motion control

Jing Wang, Hugues Mounier, Arben Cela, Silviu-Iulian Niculescu

Event driven intelligent PID controllers with applications to motion control

Jing Wang *,*** **Hugues Mounier** * **Arben Cela** **
Silviu-Iulian Niculescu *

* *Laboratoire des Signaux et Systèmes (L2S), CNRS, Supélec, Université Paris Sud 11, Supélec, 3, rue Joliot Curie, 91192 Gif sur Yvette Cedex (e-mails: jing.wang@lss.supelec.fr, hugues.mounier@lss.supelec.fr, silviu.niculescu@lss.supelec.fr)*
** *UPE, ESIEE Paris, Embedded System Department, 2 Bd Blaise Pascal, 93162 Noisy Le Grand Cedex(e-mail: celaa@esiee.fr)*
*** *Institut Polytechnique des Sciences avancées, 7-9 rue Maurice Grandcoing 94200 Ivry-sur-Seine (e-mail: wang@ipsa.fr)*

Abstract: A novel type of reduced complexity controller is proposed. It is the combination of model free control and event triggered control. The robustness of model free control, especially for badly known dynamics, is added to the event based scheme. The performances of the proposed method are illustrated in two motion controls, vehicular longitudinal control and quadrotor control. Comparisons with existing control schemes are also proposed.

Keywords: Intelligent PID, event driven control, model free control, reduced complexity controllers

1. INTRODUCTION

The trend to complex embedded control systems brings out a lot of new challenges. On one hand, the embedded character demands reduced complexity controllers. On the other hand, the complexity of the controlled systems enforces robustness of the proposed control schemes. Many constraints have to be taken into account, especially in distributed systems (see Murray et al. [2003]). Low computational cost control schemes which are able to deal with nonlinear systems with robustness are needed.

Model free control has been proven to be a simple but very efficient nonlinear feedback technique for the unknown or partially known dynamics (see Fliess et al. [2009], Choi et al. [2009]). We shall here use so-called *intelligent* PID (or *i*-PID). While retaining the PID reduced computational cost, it is able to cope with general types of nonlinearities. A precise relationship between *i*-PID and PIDs is given in d'Andréa-Novel et al. [2010]. It particularly emphasizes the ease of tuning of *i*-PID gains and gives a clearcut explanation of the performance of usual PIDs.

Contrarily to the time triggered control scheme which the control signals are sent to the actuator board every fixed sampling time, in the event based scheme, the control signals are sent only upon the triggering of an event (see Årzén [1999]). A typical event is that the tracking error goes beyond a specified limit. This type of scheme allows to go beyond the traditional Shannon sampling limit while still achieving asymptotic stability. We here propose an event based scheme for intelligent PID. The two techniques quoted above enable the efficiency and reduced complexity of the controller.

In the first section, the general setting of model free control and intelligent PID (i-PID) controllers are recalled. Then, event driven i-PID controllers are introduced. The simulations on the simplified models of longitudinal dynamics of a car and aerodynamics of quadrotor are then given.

2. MODEL FREE CONTROL

2.1 General setting

Model free control is a quite recent and very efficient technique for unknown and partially known systems (see Fliess et al. [2009]). The input-output behavior of the system is approximatively governed within its operating range by a partially known or totally unknown finite-dimensional ordinary linear or non-linear differential equation. For the sake of simplicity, the input and output are assumed to be mono-variable. The system is described implicitly as

$$E(y, \dot{y}, \ldots, y^{(a)}, u, \dot{u}, \ldots, u^{(b)}) = 0 \tag{1}$$

where $E : \mathbf{R}^{a+1} \times \mathbf{R}^{b+1} \to \mathbf{R}$ is a sufficient smooth function of its arguments. Assume that for integer ν, $0 < \nu \leqslant \iota$, $\partial E/\partial y^{(\nu)} \not\equiv 0$. The implicit function theorem (see Krantz et al. [2002]) allows to express $y^{(\nu)}$ locally

$$y^{(\nu)} = \mathfrak{E}(t, y, \dot{y}, \ldots, y^{(\nu-1)}, y^{(\nu+1)}, \ldots, y^{(\iota)}, u, \dot{u}, \ldots, u^{(\kappa)})$$

with the function $\mathfrak{E} : \mathbf{R} \times \mathbf{R}^{\iota} \times \mathbf{R}^{\kappa+1} \to \mathbf{R}$.

Replace (1) by the following phenomenological model which is only valid in a very short time interval.

$$y^{(\nu)} = F + \alpha u \tag{2}$$

where

- $\alpha \in \mathbb{R}$ is a *non-physical* constant parameter, which is chosen by the engineer in such a way that F and αu are of the same magnitude.

- The derivation order ν is also an engineer's choice.
- F is determined thanks to the knowledge of u, α, and of the estimate of $y^{(\nu)}$.

An estimate of F is obtained as follows:

$$\hat{F} = \hat{y}^{(\nu)} - \alpha\tilde{u} \tag{3}$$

where $\hat{y}^{(\nu)}$ is an estimate of the ν^{th} derivative of the measure y which is assumed available, and $\tilde{u}$ is an approximate value of u, in order to avoid algebraic loops in the controllers. Among the existing possibilities, $\tilde{u}$ can be chosen as a past value of the control variable u. The resulting controller is then

$$u = \frac{1}{\alpha}\left(y_r^{(\nu)} - \hat{F} + \Lambda(\mathbf{e}^{\langle -\xi,\zeta\rangle})\right)$$

where

- y_r is a reference trajectory which is selected as in flatness-based control (see Fliess et al. [1995]).
- $e = y_r - y$ is the tracking error.
- $\mathbf{e}^{\langle -\xi,\zeta\rangle} = (\int^{\xi} e, \int^{\xi-1} e, \ldots, e, \dot{e}, e^{(\zeta)})$, $\xi, \zeta \in [0, \nu]$, $\int^k$ is the k iterated integral, and Λ is an appropriate function $\mathbf{R}^{\xi+\zeta+1} \to \mathbf{R}$ such that the closed loop error dynamics

  $$e^{(\nu)} = \Lambda(\mathbf{e}^{\langle -\xi,\zeta\rangle})$$

  is asymptotically stable.

Remarks 2.1. a) The derivation order ν is not necessarily equal to the derivation order a of y in Equation (1).

b) The derivation order ν, is often taken equal to 1 or 2, yielding so called intelligent PIDs or i-PID (see next subsection).

c) A system may be partially unknown. It is straightforward to adapt the previous method.

d) The estimate in (3) can be obtained for example through a simple first order filtering as

$$\mathscr{L}(\hat{y}) = \frac{s}{1 + T_f s}\mathscr{L}(y)$$

typically, $1/T_f$ ranges from 8 to 20, and $\mathscr{L}$ denotes the transformation to the operational domain.

It can also be given by efficient algebraic techniques (see Mboup et al. [2009]) yielding for example the following estimate for the first derivative

$$\hat{\dot{y}} = \frac{-3!}{T^3}\int_0^T (T - 2\tau)y(\tau)d\tau$$

with T an integration window size which order of magnitude is 20 times the sampling time in a time triggered setting.

2.2 Intelligent PIDs

The desired behavior is obtained by implementing, for instance $\nu = 2$, the *intelligent PID controller* (*i-PID*) is

$$u = -\frac{\hat{F}}{\alpha} + \frac{\ddot{y}_r}{\alpha} + K_P e + K_I \int e + K_D \dot{e} \tag{4}$$

where K_P, K_I, K_D are the usual tuning gains.

Let us consider the following special cases:

- If $\nu = 2$, we may also employ an *intelligent PD controller* (*i-PD*)

  $$u = -\frac{\hat{F}}{\alpha} + \frac{\ddot{y}_r}{\alpha} + K_P e + K_D \dot{e} \tag{5}$$

- If $\nu = 1$, we restrict ourselves to an *intelligent PI controller* (*i-PI*)

  $$u = -\frac{\hat{F}}{\alpha} + \frac{\dot{y}_r}{\alpha} + K_P e + K_I \int e \tag{6}$$

  or even to an *intelligent P controller* (*i-P*)

  $$u = -\frac{\hat{F}}{\alpha} + \frac{\dot{y}_r}{\alpha} + K_P e \tag{7}$$

Remarks 2.2. a) If $\nu = 2$ (resp. 1), plugging Equations (4) or (5) (resp. (6) or (7)) in Equation (2) yields the control of a pure double (resp. simple) integrator. This is why tuning the gains of our intelligent controllers is quite straightforward.

b) It should be emphasized, if $\nu = 2$ (resp. 1), that Equation (5) (resp. (7)) is mathematically sufficient for ensuring stability around the reference trajectory. The integral term $K_I \int e$ in Equation (4) (resp. (6)) is however adding some well known robustness properties.

3. EVENT DRIVEN MODEL FREE CONTROL

The basic Årzén's event based controller consists of two parts: a time triggered event detector τ_{ed} and an event triggered PID controller τ_{ec}. See Årzén [1999]. The latter computes the control signal to be delivered to the actuator board. The former τ_{ed} runs at a fixed sampling period h_{ed}, and upon fulfillment of a certain event triggering law L_{et}, sends events to τ_{ec}. Upon reception of the event, τ_{ec} computes the control signal and sends it to the actuator board.

Examples of event triggering laws L_{et} are:

- Error threshold law:

  $$|e(t_k)| > e_{lim} \tag{8}$$

  where $e = y_r - y$ is the tracking error, t_k is the current discrete sensing time by τ_{ed}, and e_{lim} is a fixed limit.
- Error difference threshold

  $$|e(t_k) - e(t_{k-1})| > e_{lim} \tag{9}$$

- ISS based law:

  $$e(t_k) = \sigma\frac{a}{b}\,|y(t_k)| \tag{10}$$

  assuming the system can be rendered ISS (Input to State Stable) through static feedback (see Sontag [2007]). σ is chosen less than one to ensure an associated Lyapounov function decrease. a and b are chosen according to the Lipschitz constants of the $\mathcal{K}_\infty$ (consisting of all functions γ $\mathbf{R}^+ \to \mathbf{R}^+$ which are continuous, strictly increasing, satisfying $\gamma(0) = 0$ and $\lim_{\xi\to\infty} = \infty$. See, e.g., Sontag [2007]).

The present control goal is path tracking. We shall use geometric information on the reference trajectory y_r. Namely, we shall take the following event triggering scheme:

$$|e(t_k) - e(t_{k-1})| > e_{lim} \quad \wedge \quad t_k - t_{k-1} > \frac{\max(\sigma(\dot{y}_r)).h_M}{\sigma(\dot{y}_r(t_k))} \tag{11}$$

where σ is a saturation function, and h_M is the maximum sampling time ensuring stability in a time triggered scheme. We have chosen the following smooth saturating function

$$
\begin{aligned}
\sigma(\xi) &= \frac{H-l}{2(\xi_H-\xi_l)}\big(\phi(\xi)+\psi(\xi)\big)+\frac{\xi_H+\xi_l}{2} \\
\phi(\xi) &= \frac{1}{\zeta}\ \ln\big(\cosh(\zeta(\xi-\xi_l))\big) \qquad (12) \\
\psi(\xi) &= \frac{-1}{\zeta}\ \ln\big(\cosh(-\zeta(\xi-\xi_H))\big)
\end{aligned}
$$

with l and H the low and high saturated values, ξ_l and ξ_H the beginning and ending abscissa of the linear part, and ζ is a stiffness value. The $\ln(\cosh(\xi))$ functions enable to have a linear part (when $\xi \ll 0$, $\cosh(\xi) \approx \exp(-\xi)/2$, and $\ln(\cosh(\xi)) \approx -\xi/2$; when $\xi \gg 0$, $\cosh(\xi) \approx \exp(\xi)/2$, and $\ln(\cosh(\xi)) \approx \xi/2$) with smooth transitions between the constant and linear parts.

4. APPLICATION TO VEHICLE LONGITUDINAL CONTROL

4.1 Model

We shall take a simplified model of longitudinal car dynamics as example. See Kiencke et al. [2005]. No attempt will be made to take longitudinal slip into account. Thus, the motor torque is supposed to be directly transmitted to the longitudinal dynamics.

The simplified model is as the following:

$$
\begin{aligned}
M\dot{V}_x = \frac{C}{r} - C_{ae}(V_x+V_w)\big|V_x+V_w\big| - Mg\sin(\theta) - \\
MgC_{rr}\mathrm{sign}(V_x)\cos(\theta) \qquad (13)
\end{aligned}
$$

where M the vehicle's mass, V_x the vehicle's longitudinal speed. C the traction torque which is taken as control input. r the wheel's mean radius. C_{ae} the aerodynamics coefficient. V_w the wind speed disturbance. g the gravity constant. θ the road slope. C_{rr} the Rolling resistance coefficient.

The chosen values for the parameters are: $M = 1200$kg, $V_x = 0$ to 36m/s, $r = 0.025$m, $C_{ae} = 0.015\mathrm{Ns}^2/\mathrm{m}^2$, $V_w = 0$ to 14m/s, $\theta = 0$ to 0.52rad, $C_{rr} = 0.15$. In the second member of equation (13): The first term is the traction force. The second term is the aerodynamics force. The third term is the slope effect force, and the fourth term is the rolling resistance force.

4.2 Model free setting

The model given in (13) can be expressed as

$$
\dot{V}_x = F + \frac{1}{Mr}C \qquad (14)
$$

with

$$
\begin{aligned}
F = \frac{1}{M}(-C_{ae}(V_x+V_w)\big|V_x+V_w\big| - Mg\sin(\theta) - \\
MgC_{rr}\mathrm{sign}(V_x)\cos(\theta)) \qquad (15)
\end{aligned}
$$

which is of the form (2) with $\alpha = 1/Mr$. Thus, we have

$$
\begin{aligned}
C &= Mr\left(\dot{V}_{xr} - \hat{F} - k_p e - k_i\int_0^t e(\tau)d\tau\right) \\
\hat{F} &= \hat{\dot{V}}_x - \frac{1}{Mr}\widetilde{C} \qquad (16) \\
e &= V_x - V_{xr}
\end{aligned}
$$

with V_{xr} the reference speed, $\hat{\dot{V}}_x$ an estimate of the derivative of V_x, and $\widetilde{C}$ a past value of C (an approximation of C).

For instance, we can take the above form in discrete time

$$
\begin{aligned}
C(t_k) &= C(t_{k-1}) + Mr\big(\hat{e}(t_k) + k_p e(t_k) + k_i I(t_k)\big) \\
\hat{e}(t_k) &= \dot{V}_{xr}(t_k) - \hat{\dot{V}}_x(t_k) \\
e(t_k) &= V_{xr}(t_k) - V_x(t_k) \\
I(t_k) &= I(t_{k-1}) + he(t_k) \qquad (17) \\
\hat{\dot{V}}_x(t_k) &= \frac{T_f}{T_f+h}\hat{\dot{V}}_x(t_{k-1}) + \frac{1}{T_f+h}\big(V_x(t_k) - V_x(t_{k-1})\big) \\
h &= t_k - t_{k-1}
\end{aligned}
$$

For comparison, a usual PID takes the following form

$$
\begin{aligned}
C(t_k) &= K_p e(t_k) + K_i I(t_k) \\
e(t_k) &= V_{xr}(t_k) - V_x(t_k) \\
I(t_k) &= I(t_{k-1}) + he(t_k) \qquad (18) \\
h &= t_k - t_{k-1}
\end{aligned}
$$

4.3 Simulations: continuous ideal flatness based control

Supposing we have the full knowledge of the dynamics, the ideal flatness based control is of the form:

$$
\begin{aligned}
C &= Mr\left(V_{xr} - F - k_p e - k_i\int_0^t e(\tau)d\tau\right) \\
F &= -C_{ae}(V_x+V_w)\big|V_x+V_w\big| - Mg\sin(\theta) - \\
&\qquad MgC_{rr}\mathrm{sign}(V_x)\cos(\theta) \qquad (19) \\
e &= V_x - V_{xr}
\end{aligned}
$$

The error in the case of flatness based control is depicted in figure 1.

4.4 Simulations: time triggered PI control and i-PID control

We first compare the cases of a time triggered PID and a time triggered *i*-PID.

Consider a fixed sampling time of $h = 10$ms (knowing that $h = 35$ms is the limit of stability). This yields 1976 actuation steps. We take a PI controller with gains $k_p = 17000$ and $k_i = 100$. The reference trajectory and the tracking error are depicted in figure 1.

4.5 Simulations: event triggered PI control and i-PID control

We now consider the event triggered controls. The event triggering scheme for PI control is the classical error difference of equation (9). The limit e_{lim} in (9) is taken as

$$
e_{lim} = \frac{\max(y_r) - \min(y_r)}{200}
$$

It yields 291 actuation steps and the tracking error is given in figure 1. The *i*-PI controller is with gains $K_p = 60$ and $K_i = 6$.

4.6 Discussion

Note that the maximum absolute tracking error is $6.4.10^{-2}$ m/s in the PI case, and $3.2.10^{-3}$ m/s in the *i*-PI case which

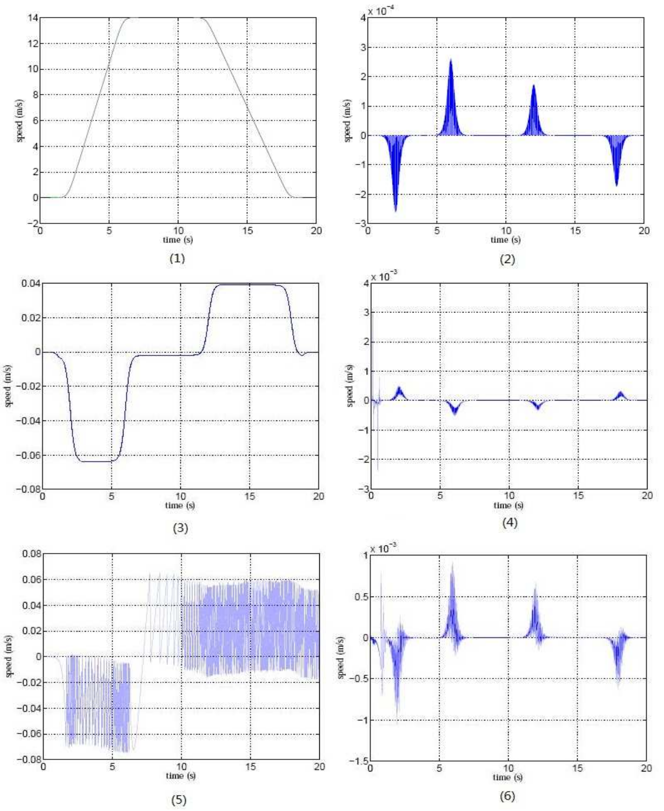

Fig. 1. (1) Reference trajectory. (2) Tracking errors of ideal flatness based control. (3) Tracking errors of time triggered PI control. (4) Tracking errors of time triggered i-PID control. (5) Tracking errors of event triggered PI control. (6) Tracking errors of event triggered i-PID control.

is 20 times less (2000%) than in the PI case. If we exclude the first second, the maximum absolute tracking error in the i-PI case is of $5.2.10^{-4}$, which is 123 times less than in the PI case.

Consider now an i-PI control. The event triggered scheme is the one given in equation (11), with $l = 1$, $L = 20$, $\xi_l = -2$, $\xi_H = 2$, and $\zeta = 6$. The corresponding tracking error is given in figure 1, and was performed in 569 actuation steps. The gain in performance, when using an i-PI instead of a PI, is 68.18 and the loss in actuation steps is 1.95.

Model free control has better performance than PI control. Using the event triggered schemes, i-PID can further reduce the number of actuation loops, which is very useful for real time control systems.

5. APPLICATION TO QUADROTOR CONTROL

5.1 Model

The chosen model of quadrotor is depicted in equations (20). See Bouabdallah [2007]. The rotation angles ϕ, θ and ψ are along the world axis x, y and z respectively, namely, roll, pitch and yaw. Ω_r $(i = 1..4)$ are the angular velocities of each rotor, which are the real inputs of the quadrotor. The forces T_i, H_i $(i = 1..4)$ are the thrust and hub forces of each motor. The moments R_i, Q_i $(i = 1..4)$ are the drag and rolling moments of each rotor. The quantities $\dot{\omega}_1\dot{\omega}_2(I_{ii}-I_{ii})$, $J_r\dot{\omega}\Omega_r$ $(\omega = \phi, \theta, \psi; i = x, y, z)$ are the body gyroscopic effects and propeller gyroscopic effects. The notations c and s represent cos and sin respectively. The values of all the parameters can be found in Bouabdallah [2007].

$$
\begin{aligned}
I_{xx}\ddot{\phi} &= \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + J_r\dot{\theta}\Omega_r + l(-T_2 + T_4) - \\
&\qquad h\Big(\sum_{i=1}^{4} H_{yi}\Big) + (-1)^{i+1}\sum_{i=1}^{4} R_{mxi} \\
I_{yy}\ddot{\theta} &= \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) - J_r\dot{\phi}\Omega_r + l(T_1 - T_3) - \\
&\qquad h\Big(\sum_{i=1}^{4} H_{xi}\Big) + (-1)^{i+1}\sum_{i=1}^{4} R_{myi} \\
I_{zz}\ddot{\psi} &= \dot{\theta}\dot{\phi}(I_{xx} - I_{yy}) + (-1)^i\sum_{i=1}^{4} Q_i + \\
&\qquad l(H_{x2} - H_{x4}) + l(-H_{y1} + H_{y3}) \\
m\ddot{z} &= -mg + (c\theta c\phi)\sum_{i=1}^{4} T_i \\
m\ddot{x} &= (s\psi s\phi + c\psi s\theta c\phi)\sum_{i=1}^{4} T_i - \sum_{i=1}^{4} H_{xi} - \frac{1}{2}C_x A_c \rho \dot{x}|\dot{x}| \\
m\ddot{y} &= (-c\psi s\phi + s\psi s\theta c\phi)\sum_{i=1}^{4} T_i - \sum_{i=1}^{4} H_{yi} - \frac{1}{2}C_y A_c \rho \dot{y}|\dot{y}|
\end{aligned}
\tag{20}
$$

The most important forces and moments are the thrust T and the rolling moments Q. Therefore, we can take

$$
u_1 = \sum_{i=1}^{4} T_i \qquad u_2 = l(-T_2 + T_4)
$$

$$
u_3 = l(T_1 - T_3) \qquad u_4 = (-1)^i \sum_{i=1}^{4} Q_i
$$

as control inputs to compute the needed torques for each rotor, and then use them to control the altitude z, position x, y and direction ψ.

5.2 Altitude z control

The equation given in (20) related to z can be expressed as

$$
m\ddot{z} = (c\theta c\phi)u_1 + F_z \tag{21}
$$

where F_z can be considered as disturbances (e.g. the wind) or some parts of dynamics neglected in (20). In discrete time, the unknown part F_z can be expressed as following. The estimate of $\ddot{z}(k)$ is denoted as $\hat{\ddot{z}}(k)$.

$$
\hat{F}_z = m\hat{\ddot{z}}(t_k) - (c\theta c\phi)u_1(t_{k-1}) \tag{22}
$$

Therefore, the chosen control input is

$$
u_1(t_k) = u_1(t_{k-1}) + \frac{m}{c\theta c\phi}(\hat{e}^z_{2d}(t_k) + k^z_1 e^z_d(t_k) + k^z_2 e^z(t_k)) \tag{23}
$$

with

$$\hat{e}^z_{2d}(t_k) = \ddot{z}_r(t_k) - \hat{\ddot{z}}(t_k),\ \ e^z_d(t_k) = \dot{z}_r(t_k) - \dot{z}(t_k)$$
$$e^z(t_k) = z_r(t_k) - z(t_k)$$
$$\hat{\ddot{z}}(t_k) = \frac{T_f}{T_f+h}\hat{\ddot{z}}(t_{k-1}) + \frac{1}{T_f+h}\big(z(t_k) - z(t_{k-1})\big)$$

$\ddot{z}_r, \dot{z}_r, z_r$ are the reference acceleration, velocity and position of z. The variable sampling step is $h = t_k - t_{k-1}$.

5.3 Position x,y control

We want to use u_2 and u_3 to control directly the position x, y. Therefore, we need to differentiate twice the equations related to x and y in (20) in order to appear the control inputs u_2 and u_3. Since the equations in x and y are coupled, we get

$$x^{(4)} = \frac{u_1}{mI_{xx}}(s\psi c\phi - c\psi s\theta s\phi)u_2 + \frac{u_1}{mI_{yy}}(c\psi c\theta c\phi)u_3 + F_x$$
$$y^{(4)} = -\frac{u_1}{mI_{xx}}(c\psi c\phi + s\psi s\theta s\phi)u_2 + \frac{u_1}{mI_{yy}}(s\psi c\theta c\phi)u_3 + F_y \tag{24}$$

where F_x, F_y are considered as the badly known parts. For simplicity, we define $A = \frac{u_1}{mI_{xx}}(s\psi c\phi - c\psi s\theta s\phi)$, $B = \frac{u_1}{mI_{yy}}(c\psi c\theta c\phi)$, $C = -\frac{u_1}{mI_{xx}}(c\psi c\phi + s\psi s\theta s\phi)$ and $D = \frac{u_1}{mI_{yy}}(s\psi c\theta c\phi)$. Using the model free control scheme as before, we get

$$\begin{pmatrix} u_2(t_k) \\ u_3(t_k) \end{pmatrix} = \begin{pmatrix} u_2(t_{k-1}) \\ u_3(t_{k-1}) \end{pmatrix} + \begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} \begin{pmatrix} \hat{e}^x_{4d} + \sum_{i=0}^{3} k^x_i e^x_{id} \\ \hat{e}^y_{4d} + \sum_{i=0}^{3} k^y_i e^y_{id} \end{pmatrix} \tag{25}$$

where $\hat{e}^x_{4d}, \hat{e}^y_{4d}$ are the errors between the references $x^{(4)}_r, y^{(4)}_r$ and the estimates of $x^{(4)}, y^{(4)}$.

5.4 Yaw control

For yaw control, we consider the equation of ψ as

$$I_{zz}\ddot{\psi} = u_4 + F_\psi \tag{26}$$

Then the control feedback is

$$u_4(t_k) = u_4(t_{k-1}) + I_{zz}(\hat{e}^\psi_{2d}(t_k) + k^\psi_1 e^\psi_d(t_k) + k^\psi_2 e^\psi(t_k)) \tag{27}$$

where $\hat{e}^\psi_{2d}$ is the error between the reference $\ddot{\psi}_r$ and the estimate of $\ddot{\psi}$.

5.5 Simulation: time triggered control

The task is to follow a rounded square path with length of 2m while hovering at the altitude of 10m, which is given in (28). The desired length is h_d, and T_f is the time needed to reach the desired length. Here we choose h_d equals 2m, and T_f equals 6s. The reference trajectory is in figure 2.

In the time triggered i-PID control, the sampling time is 10ms, and it yields 2785 actuation steps. The results are given in figure 3. The red lines are the desired trajectories. The maximum errors in x and y are both less than 0.05m, that is, less than 2, 5% of the desired length.

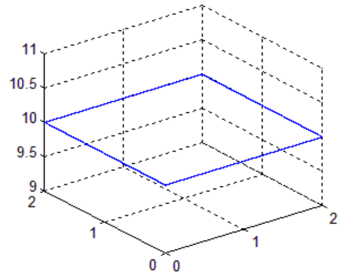

Fig. 2. Reference trajectory for the quadrotor.

$$\sigma(t) = \begin{cases} 0 & 0 \leqslant t \leqslant t_1, t_4 < t \leqslant 30 \\ h_d \dfrac{t^5}{t^5 + (T_f - t)^5} & t_1 < t \leqslant t_2 \\ 2 & t_2 < t \leqslant t_3 \\ h_d - h_d \dfrac{t^5}{t^5 + (T_f - t)^5} & t_3 < t \leqslant t_4 \end{cases}$$
$$\begin{aligned} & h_d = 2, T_f = 6 \\ & x = \sigma(t) \quad with\ t_1 = 3, t_2 = 9, t_3 = 15, t_4 = 21. \\ & y = \sigma(t) \quad with\ t_1 = 9, t_2 = 15, t_3 = 21, t_4 = 27. \\ & z = 10 \end{aligned} \tag{28}$$

5.6 Simulation: event triggered control

In the event triggered i-PID control, the event triggering law is the absolute error limit. We set the error limit of z to be 0.1m. The error limit of yaw angle is 0.1rad. For x and y, we take the error limits both as 0.1m. The event triggered i-PID control yields 2389 actuation steps. The results are given in figure 4.

5.7 Discussion

The system mentioned in (20) is not complete. The aerodynamics of the system is complicated, and many more forces and moments will affect the system. Therefore, a control scheme which can adapt to the changes of the system is needed. The time triggered model free control performed nicely. It controls the system without the need of computing all the forces and moments in the system. In event triggered i-PID control, we set the the error limit to be 5% of the reference. It has 396 steps less comparing to the time triggered model free scheme while still achieving stability.

6. CONCLUSION

Event triggered model free controllers which yields strong robustness while needing few computing resources is proposed in this paper. It is very efficient to control the nonlinear multi-input-output system which traditional PID is not able to. The i-PID control is also efficient to solve the partially known systems. From the simulation of a quadrotor model, we see that the i-PID control scheme avoids the heavy computations of the control laws, forces, moments and 4th derivatives of the variables. Moreover, the event triggered scheme enables to eliminate the small vibrations in the system while diminishing the number of actuation steps.

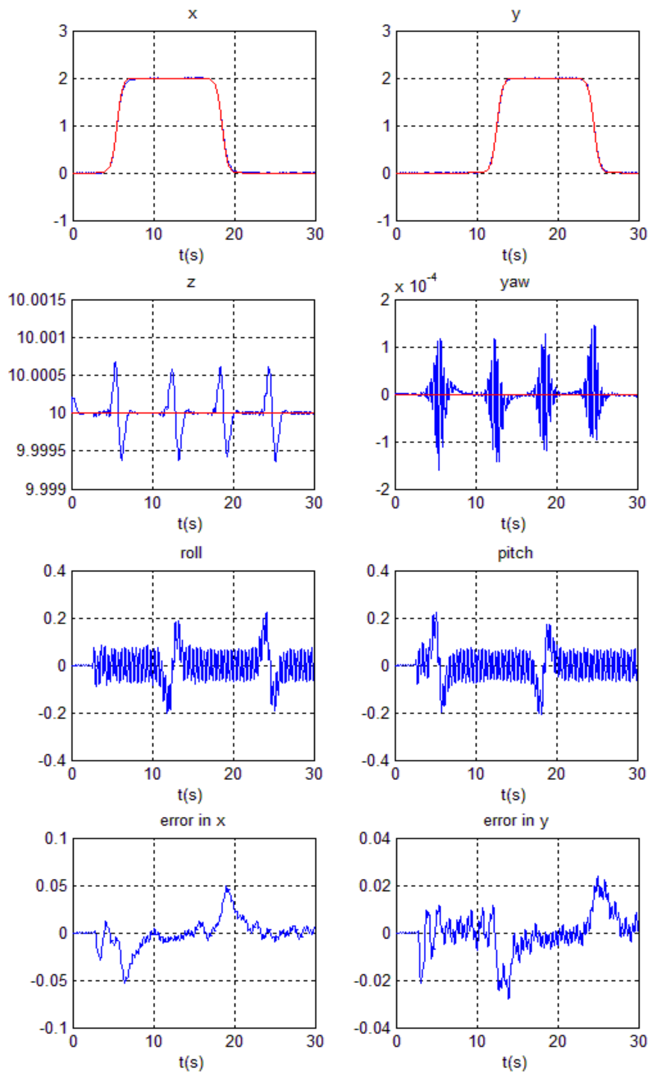

Fig. 3. Time triggered i-PID control of quadrotor

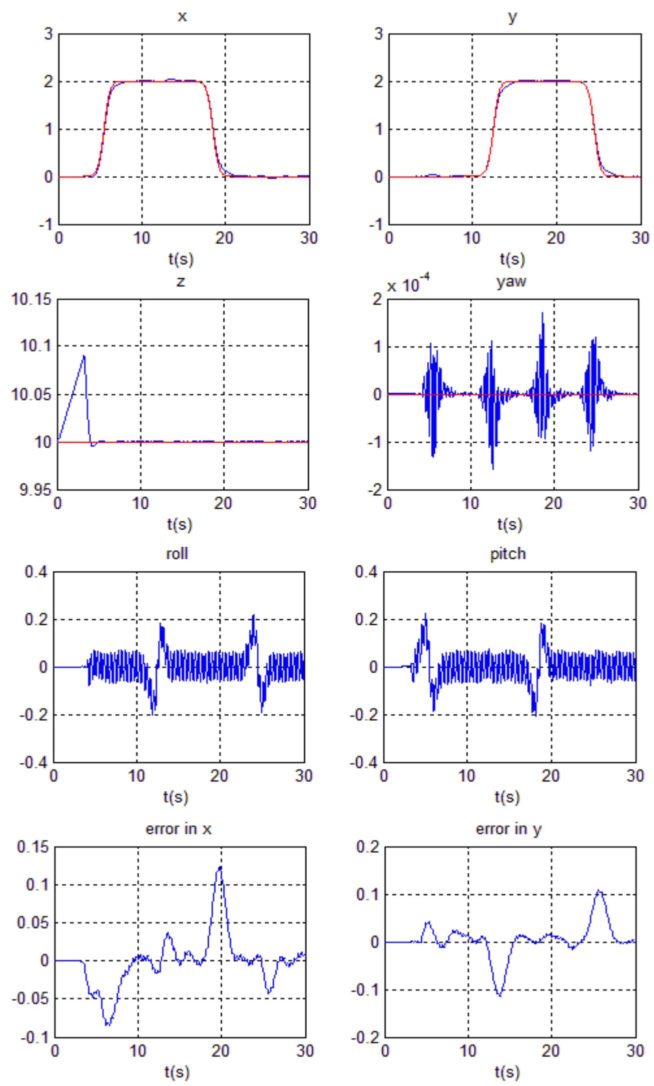

Fig. 4. Event triggered i-PID control of quadrotor

ACKNOWLEDGEMENTS

Jing Wang is financially supported by IPSA (Institut Polytechnique des Sciences Avancées).

REFERENCES

K.E. Årzén. A simple event-based PID controller. *Proc. of 14th IFAC World Congress*, Beijing, PR China, 1999.

B. d'Andréa-Novel, M. Fliess, C. Join, H. Mounier, B. Steux. A mathematical explanation via "intelligent" PID controllers of the strange ubiquity of PIDs. *Proc. of 18th Mediterranean Conference on Control and Automation*, Marrakech, Morroco, 2010.

S. Bouabdallah. Design and Control of Quadrotors with Application to Autonomous Flying. *PhD thesis*, Ecole Polytechnique Federale De Lausanne, 2007.

S. Choi, B. d'Andréa-Novel, M. Fliess, H. Mounier. Model-free control of automotive engine and brake for stop-and-go scenario. *Proc. of 10^{th} Europ. Control Conf.* (ECC'09), Budapest.

M. Fliess, J. Levine, P. Martin and P. Rouchon. Flatness and defect of nonlinear systems : introductory theory and examples. *Internat. J. Control*, Vol. 61, pp. 1327–1361, 1995.

M. Fliess and C. Join. Model-free control and intelligent PID controllers: towards a possible trivialization of nonlinear control?. *Proc. of 15^{th} IFAC Symp. System Identif.*, Saint-Malo, France, 2009.

U. Kiencke, L. Nielsen. *Automotive Control Systems: For Engine, Driveline, And Vehicle.* Springer, Berlin, 2005.

S.G. Krantz, H.R. Parks. *The Implicit Function Theorem: History, Theory, and Applications.* Birkhser, Boston, United States, 2002.

R.M. Murray, K.J. Astrom, S.P. Boyd, R.W. Brockett, G. Stein. Future Directions in Control in an Information-Rich World. *IEEE Contr. Syst. Mag.*, vol. 23, pp. 20–33, 2003.

M. Mboup, C. Join, and M. Fliess. Numerical differentiation with annihiators in noisy environment. *Numer. Algor.*, vol. 50, pp. 439–467, 2009.

R. Rajamani. *Vehicle Dynamics and Control.* Springer, New York, 2006.

E.D. Sontag. Input to state stability: Basic concepts and results. *Nonlinear and Optimal Control Theory*, pp. 163-220, Springer-Verlag, Berlin, 2007.