Event driven intelligent PID controllers with applications to motion control
Jing Wang, Hugues Mounier, Arben Cela, Silviu-Iulian Niculescu

To cite this version:
Jing Wang, Hugues Mounier, Arben Cela, Silviu-Iulian Niculescu. Event driven intelligent PID controllers with applications to motion control. 18th IFAC World Congress (International Federation of Automatic Control ) (IFAC2011), Aug 2011, Milano, Italy. p. 10080-10085, 2011. <hal-00832014>

HAL Id: hal-00832014
https://hal.archives-ouvertes.fr/hal-00832014
Submitted on 9 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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. [2002], 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 Arzén [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

\[ E(y, \dot{y}, \ldots, y^{(α)}, u, \dot{u}, \ldots, u^{(β)}) = 0 \]  

where \( E : \mathbb{R}^{α+1} \times \mathbb{R}^{β+1} \to \mathbb{R} \) is a sufficient smooth function of its arguments. Assume that for integer \( ν \), \( 0 < ν \leq t \), \( \partial E/\partial y^{(ν)} \neq 0 \). The implicit function theorem (see Krautz et al. [2002]) allows to express \( y^{(ν)} \) locally

\[ y^{(ν)} = \mathcal{E}(t, y, \dot{y}, \ldots, y^{(ν−1)}, y^{(ν+1)}, \ldots, y^{(α)}, u, \dot{u}, \ldots, u^{(σ)}) \]

with the function \( \mathcal{E} : \mathbb{R} \times \mathbb{R}^t \times \mathbb{R}^{α+1} \to \mathbb{R} \).

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

\[ y^{(ν)} = F + αu \]  

where

- \( α \in \mathbb{R} \) 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 \( \nu \) is also an engineer’s choice. 

- \( F \) is determined thanks to the knowledge of \( u, \alpha, \) and of the estimate of \( y^{(\nu)}. \)

An estimate of \( F \) is obtained as follows:

\[
\ddot{F} = \ddot{y}^{(\nu)} - \alpha \ddot{\bar{u}} \tag{3}
\]

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

\[
u = \frac{1}{\alpha} \left( \ddot{y}^{(\nu)}_r - \bar{u} + \Lambda(e^{(-\xi,\zeta)}) \right)
\]

where

- \( y_r \) is a reference trajectory which is selected as in flatness-based control (see Fleiss et al. [1995]).
- \( e = y_r - y \) is the tracking error.
- \( e^{(-\xi,\zeta)} = \left( \int f e, \int \dot{f} e, \ldots, e, \dot{e}, e^{(\nu)} \right), \xi, \zeta \in [0, \nu], \int^k \) is the \( k \) iterated integral, and \( \Lambda \) is an appropriate function \( \mathbb{R}^{\nu+1} \rightarrow \mathbb{R} \) such that the closed loop error dynamics

\[
\dot{e}^{(\nu)} = \Lambda(e^{(-\xi,\zeta)})
\]

is asymptotically stable.

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

\[
\mathcal{L}(\dot{y}) = \frac{1}{1 + T_I s} \mathcal{L}(y)
\]

typically, \( 1/T_I \) ranges from 8 to 20, and \( \mathcal{L} \) 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

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

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

\[
u = \frac{-\ddot{F} + \dot{y}_r}{\alpha} + K_pe + K_d\dot{e} \tag{4}
\]

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)

\[
u = \frac{-\ddot{F} + \dot{y}_r}{\alpha} + K_pe + K_d\dot{e} \tag{5}
\]

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

\[
u = \frac{-\ddot{F} + \dot{y}_r}{\alpha} + K_pe + K_i \int e \tag{6}
\]

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

\[
u = \frac{-\ddot{F} + \dot{y}_r}{\alpha} + K_pe \tag{7}
\]

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

Examples of event triggering laws \( L_{et} \) are:

- Error threshold law:

\[
|e(t_k)| > \epsilon_{tim} \tag{8}
\]

where \( e = y_r - y \) is the tracking error, \( t_k \) is the current discrete sensing time by \( \tau_{ed} \), and \( \epsilon_{tim} \) is a fixed limit.

- Error difference threshold

\[
|e(t_k) - e(t_{k-1})| > \epsilon_{tim} \tag{9}
\]

- ISS based law:

\[
e(t_k) = \frac{a}{\sigma^b} |y(t_k)| \tag{10}
\]

assuming the system can be rendered ISS (Input to State Stable) through static feedback (see Sonntag [2007]). \( \sigma \) is chosen less than one to ensure an associated Lyapunov function decrease. \( a \) and \( b \) are chosen according to the Lipschitz constants of the \( K_{\infty} \) (consisting of all functions \( \gamma : \mathbb{R}^+ \rightarrow \mathbb{R}^+ \) which are continuous, strictly increasing, satisfying \( \gamma(0) = 0 \) and \( \lim_{t \rightarrow \infty} \gamma(t) = \infty \). See, e.g., Sonntag [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:

\[
|e(t_k) - e(t_{k-1})| > \epsilon_{tim} \quad \land \quad t_k - t_{k-1} > \frac{\max(\sigma(\dot{y}_r(t))), h_M}{\sigma|y_r(t_k)|} \tag{11}
\]

where \( \sigma \) 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

\[
\sigma(\dot{y}_r(t)) = \frac{1}{\sigma_0} \left( 1 - \frac{\dot{y}_r(t)}{\sigma_0} \right) \tag{12}
\]

where \( \sigma_0 > 0 \) is a parameter.

At each event generation, \( \tau_{ed} \), the controller computes the current error \( e(t_k) \), and chooses the next switching error \( e_{sw} \) as

\[
e(t_k) = \frac{a}{\sigma^b} |y_{sw}(t_k)| \tag{13}
\]

where \( y_{sw}(t_k) \) is the reference trajectory at \( t_k \) taking into account the current time delay.

The construction of the event detector \( \tau_{ed} \) is then split into two steps:

\[
\tau_{ed} = \min \left[ \tau_{sw} - \sigma^{-b} \frac{\max(\sigma(\dot{y}_r(t))), h_M}{\sigma|y_r(t_k)|}, \tau_{sw} - \epsilon_{tim} \right] \tag{14}
\]

### Event Detector

The event detector \( \tau_{ed} \), logs on \( \tau_{sw} \), the next switching error and the time delay \( \epsilon_{tim} \) as

\[
\tau_{sw} = \tau_{sw} - \sigma^{-b} \frac{\max(\sigma(\dot{y}_r(t))), h_M}{\sigma|y_r(t_k)|} \tag{15}
\]

where \( \sigma \), and \( \sigma_0 \) are the same as before.

### Controller

The controller computes the current switching error \( e_{sw} \) according to

\[
e_{sw}(t_k) = \frac{a}{\sigma^b} \left( \sigma_0 - \dot{y}_r(t_k) \right) \tag{16}
\]

where \( \sigma_0 \) is the constant saturation level.

### State Estimator

The state estimator \( \hat{y}_r(t) \) is computed as

\[
\hat{y}_r(t) = \hat{y}_r(t) + \dot{y}_r(t) \Delta t \tag{17}
\]

where \( \Delta t \) is the time delay between two consecutive events.

### Path Tracking

The path tracker takes into account the position error \( e_r \), and the rate of change of the position error \( e_{\dot{r}} \). The controller computes the switching error \( e_{sw} \) according to

\[
e_{sw}(t_k) = \frac{a}{\sigma^b} \left( \sigma_0 - \dot{y}_r(t_k) \right) \tag{18}
\]

where \( \sigma_0 \) is the constant saturation level.

The controller computes the current switching error \( e_{sw} \) according to

\[
e_{sw}(t_k) = \frac{a}{\sigma^b} \left( \sigma_0 - \dot{y}_r(t_k) \right) \tag{19}
\]

where \( \sigma_0 \) is the constant saturation level.

The state estimator \( \hat{y}_r(t) \) is computed as

\[
\hat{y}_r(t) = \hat{y}_r(t) + \dot{y}_r(t) \Delta t \tag{20}
\]

where \( \Delta t \) is the time delay between two consecutive events.
\[
\sigma(\xi) = \frac{H - l}{2(\xi H - \xi_l)} \left( \phi(\xi) + \psi(\xi) \right) + \frac{\xi H + \xi_l}{2} \\
\phi(\xi) = \frac{1}{\xi} \ln \left( \cosh(\xi - \xi_l) \right) \\
\psi(\xi) = -\frac{1}{\xi} \ln \left( \cosh(-\xi - \xi_H) \right)
\]

with \( l \) and \( H \) the low and high saturated values, \( \xi_l \) and \( \xi_H \) the beginning and ending ascissa of the linear part, and \( \zeta \) 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:

\[
M\dot{V}_x = C - C_{ae}(V_x + V_w)|V_x + V_w| - Mg\sin(\theta) - MgC_{rr}\text{sign}(V_x)\cos(\theta) \tag{13}
\]

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, \( \theta \) the road slope. \( C_{rr} \) the rolling resistance coefficient.

The chosen values for the parameters are: \( M = 1200 \text{kg}, V_x = 0 \) to 36\text{m/s}, \( r = 0.025 \text{m}, C_{ae} = 0.015 \text{Ns/m}^2, V_w = 0 \) to 14\text{m/s}, \( \theta = 0 \) to 0.52\text{rad}, \( C_{rr} = 0.15 \). 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

\[
\dot{V}_x = F + \frac{1}{Mr} C \tag{14}
\]

with

\[
F = \frac{1}{M}(-C_{ae}(V_x + V_w)|V_x + V_w| - Mg\sin(\theta) - MgC_{rr}\text{sign}(V_x)\cos(\theta)) \tag{15}
\]

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

\[
C = Mr \left( \dot{V}_x - \dot{\tilde{F}} - k_p e - k_i \int_0^t e(\tau)d\tau \right) \\
\dot{\tilde{F}} = \dot{\tilde{V}}_x - \frac{1}{Mr} \tilde{C} \\
e = V_x - V_{xr}
\]

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

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

\[
C(t_k) = C(t_{k-1}) + Mr(\dot{\tilde{V}}_x(t_k) + k_p e(t_k) + k_i I(t_k))
\]

\[
\dot{\tilde{V}}_x(t_k) = \frac{T_f}{T_f + h} \tilde{V}_x(t_{k-1}) + \frac{1}{T_f + h}(V_x(t_k) - V_x(t_{k-1})) \tag{17}
\]

\[
e(t_k) = V_x(t_k) - V_{xr}(t_k) \tag{18}
\]

\[
I(t_k) = I(t_{k-1}) + he(t_k)
\]

For comparison, a usual PID takes the following form

\[
C(t_k) = K_p e(t_k) + K_i I(t_k) \tag{19}
\]

\[
e = V_x - V_{xr}
\]

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

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:

\[
C(t_k) = Mr \left( V_{xr} - F - k_p e - k_i \int_0^t e(\tau)d\tau \right)
\]

\[
F = -C_{ae}(V_x + V_w)|V_x + V_w| - Mg\sin(\theta) - MgC_{rr}\text{sign}(V_x)\cos(\theta)
\]

\[
e = V_x - V_{xr}
\]

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 \( h = 10 \text{ms} \) (knowing that \( h = 35 \text{ms} \) 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-PID controller is with gains \( K_p = 60 \) and \( K_i = 6 \).

4.6 Discussion

Note that the maximum absolute tracking error is \( 6.410^{-2} \text{m/s} \) in the PI case, and \( 3.210^{-3} \text{m/s} \) in the i-PID case which
The forces of each rotor, which are the real inputs of the quadrotor.

\[ r \text{ roll, pitch and yaw. } \Omega \]

Using the event triggered schemes, is 1. The number of actuation loops, which is very useful

actuation steps. The gain in performance, when using an

the PI case. The event triggered scheme

2, which is 123 times less than in

PI case is of 5. APPLICATION TO QUADROTOR CONTROL

\[ \sum_{i=1}^{4} Q_i \]

\[ m\ddot{z} = -mg + (c\theta c\phi) \sum_{i=1}^{4} T_i \]

\[ m\ddot{x} = (s\psi s\phi + c\psi s\theta c\phi) \sum_{i=1}^{4} T_i - \sum_{i=1}^{4} H_{xi} - \frac{1}{2} C_x A_r \rho \dot{x} |\dot{x}| \]

\[ m\ddot{y} = (-c\psi s\phi + s\psi s\theta c\phi) \sum_{i=1}^{4} T_i - \sum_{i=1}^{4} H_{yi} - \frac{1}{2} C_y A_r \rho jy |\dot{y}| \]

(20)

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

\[ u_1 = \sum_{i=1}^{4} T_i \quad u_2 = l(-T_2 + T_4) \]

\[ u_3 = l(T_1 - T_3) \quad u_4 = (-1)^i \sum_{i=1}^{4} Q_i \]

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 \( \psi \).

5.2 Altitude z control

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

\[ m\ddot{z} = (c\theta c\phi) u_1 + F_z \]  

(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 \( \ddot{z}(k) \).

\[ \ddot{F}_z = m\ddot{z}(k) - (c\theta c\phi) u_1(t_{k-1}) \]

(22)

Therefore, the chosen control input is

\[ u_1(t_k) = u_1(t_{k-1}) + \frac{m}{c\theta c\phi} \left( \ddot{z}(k) + F_z \right) \]

(23)

with

5. APPLICATION TO QUADROTOR CONTROL

5.1 Model

The chosen model of quadrotor is depicted in equations (20). See Bouabdallah [2007]. The rotation angles \( \phi, \theta \) and \( \psi \) are along the world axis \( x, y \) and \( z \) respectively, namely, roll, pitch and yaw. \( \Omega_r (i = 1..4) \) 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 \( \dot{\omega}_1 \dot{\omega}_2 (I_{i1} - I_{i2}), J_r \omega_r (\omega = \phi, \theta, \psi; i = x, y, z) \) 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].

\[ I_{xx}\ddot{\phi} = \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + J_{r}\dot{\Omega}_r + l(-T_2 + T_4) - h(\sum_{i=1}^{4} H_{yi}) + (-1)^{i+1}\sum_{i=1}^{4} R_{mi1} \]

\[ I_{yy}\ddot{\theta} = \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) - J_{r}\dot{\Omega}_r + l(T_1 - T_3) - h(\sum_{i=1}^{4} H_{xi}) + (-1)^{i+1}\sum_{i=1}^{4} R_{mi1} \]

\[ I_{zz}\ddot{\psi} = \dot{\phi}\dot{\theta}(I_{xx} - I_{yy}) + (-1)^{i+1}\sum_{i=1}^{4} Q_{i1} + l(H_{x2} - H_{x4}) + l(-H_{y1} + H_{y3}) \]

\[ m\ddot{z} = -mg + (c\theta c\phi) \sum_{i=1}^{4} T_i \]

\[ m\ddot{x} = (s\psi s\phi + c\psi s\theta c\phi) \sum_{i=1}^{4} T_i - \sum_{i=1}^{4} H_{xi} - \frac{1}{2} C_x A_r \rho \dot{x} |\dot{x}| \]

\[ m\ddot{y} = (-c\psi s\phi + s\psi s\theta c\phi) \sum_{i=1}^{4} T_i - \sum_{i=1}^{4} H_{yi} - \frac{1}{2} C_y A_r \rho jy |\dot{y}| \]

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.

\[ \sum_{i=1}^{4} Q_{i1} \]

\[ m\ddot{z} = c\theta c\phi u_1 + F_z \]
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

$$
\dot{z}_2(t_k) = \ddot{z}_r(t_k) - \ddot{z}(t_k), \quad \ddot{z}_2(t_k) = \dddot{z}_r(t_k) - \dddot{z}(t_k)
$$

$$
\ddot{z}(t_k) = \frac{T_f}{T_f + h} \ddot{z}(t_{k-1}) + \frac{1}{T_f + h} (z(t_k) - \ddot{z}(t_{k-1}))
$$

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

5.4 Yaw control

For yaw control, we consider the equation of $\psi$ as

$$
I_{zz} \dddot{\psi} = u_4 + F_\psi
$$

Then the control feedback is

$$
u_4(t_k) = u_4(t_{k-1}) + I_{zz} \left( \dddot{\psi}(t_k) + k_1 \dddot{\psi}(t_k) + k_2 \dddot{\psi}(t_{k}) \right)
$$

where $\dddot{\psi}$ is the control error between the reference $\dddot{\psi}$ and the estimate $\dddot{\psi}$.

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.

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 non-linear 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

ACKNOWLEDGEMENTS

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

REFERENCES