Comparative Study Between Two Tuning MPC Approaches: Application to a Simulated QTW UAV

Marwa TURKI, **Khaled Benkhoud**, Nicolas LANGLOIS & Adnan YACINE

PREDIRE est cofinancé par l'Union européenne.
L'Europe s'engage en Normandie
avec le Fonds européen de développement régional

GT CPNL
ONERA Châtillon
le 04/06/2018

Outline

- Introduction
- The convertible UAVs
- QTW T-S Fuzzy Modeling
- TSMPC tuning based PSO approch
- TSMPC tuning based Analytical approch
- Comparative Study
- Conclusion & outlook

Introduction

- Motivations
- Influence of parameters values
- Existing approaches

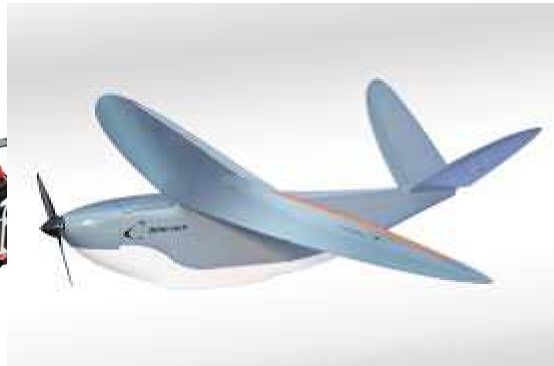
The convertible UAVs (1/2)

UAVs are divided in two major categories :

- **The rotary-wing systems :** which lift is provided by the rotation of the propellers
  - ✓ Take-off and landing vertically, perform the hover flight,…
- **The fixed-wing systems :** which lift is provided by the airflow over the wings induced by the own movement of the vehicle
  - ✓ fly forward at high speed, long range, superior endurance,…

There a… …ntional
Unmanne…

- the su…
- identi…
- medic…

The convertible UAVs (2/2)

The convertible aircrafts :

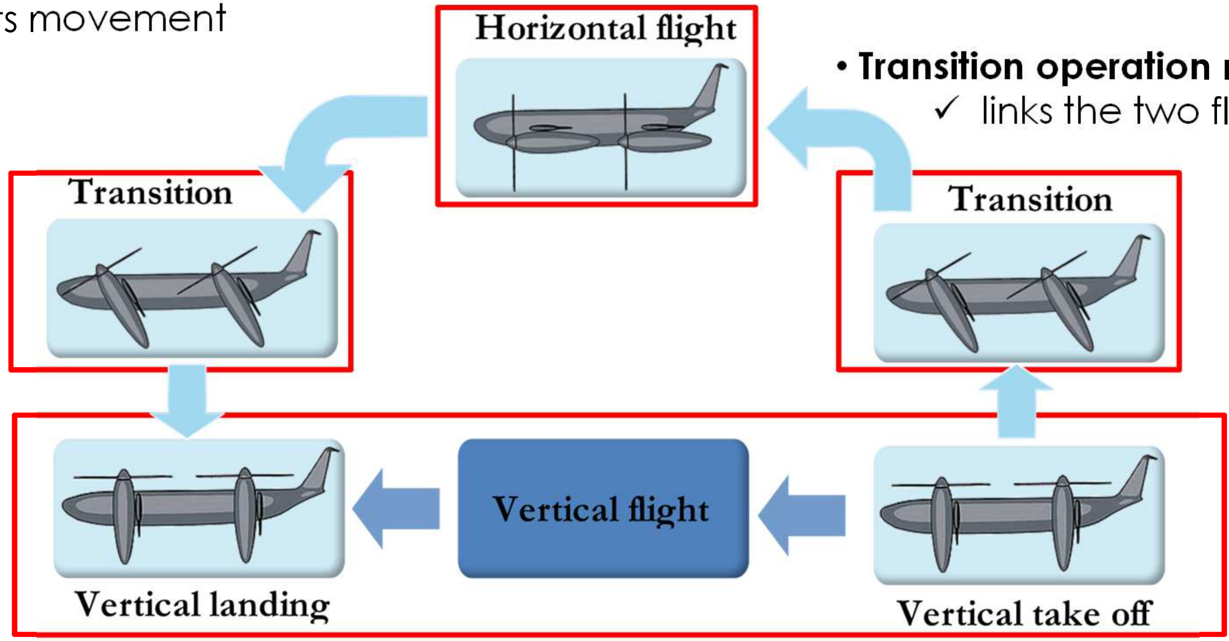
- combines the advantages of rotary-wings and fixed-wings aircrafts
- minimizes the energy consumed in forward flights
- can take-off and landing vertically
- transition between the hover flight to forward flight and vice versa

The transition between the flight phases :

- tilting the full vehicle body (tail-sitter or tilt-body)
- tilting only its rotors using a dedicated mechanism (tilt-rotor or tilt-wing)

Modeling of the QTW UAV (1/3)

- **Horizontal flight mode :**
  - ✓ behaves like a conventional plane
  - ✓ the tilt angle of wings almost equal to zero degree
  - ✓ generates the aerodynamic forces to lift and control its movement



- **Transition operation mode:**
  - ✓ links the two flight modes

- **Vertical flight mode:**
  - ✓ the tilt angles of the wings are nearly equal to 90°
  - ✓ based only on its rotors
  - ✓ behaves like a Quadrotors

Modeling of the QTW UAV (2/3)

The Newton-Euler formalism

$$m\dot{V}_i = F_t$$

$$I_c\dot{\Omega}_c = -M_t + \Omega_c \wedge (I_c \times \Omega_c)$$

Translational equations of motion :

$$m\dot{V}_i = F_t$$

with F_t is the total external force :

$$F_t = R_{ib}(F + F_g + F_d + F_a)$$

where :

- F : the total thrust forces
- F_g : the gravity forces
- F_d : the external disturbances
- F_a : the aerodynamic forces

Rotational equations of motion :

$$I_c\dot{\Omega}_c = -M_t + \Omega_c \wedge (I_c \times \Omega_c)$$

with M_t the total external torque :

$$M_t = M + M_g + M_d$$

where :

- M : the thrust torques
- M_g : the gyroscopic effects
- M_d : the external disturbances
- M_a : The aerodynamic torques

Modeling of the QTW UAV (3/3)

Dynamics model of a QTW vehicle

$$
\begin{cases}
\ddot{x} = \dfrac{1}{m}\left[\left(c\psi c\theta c\gamma - (c\phi s\theta c\psi + s\phi s\psi) s\gamma \right) T_t + \left(c\psi c\theta \right) F_D + (c\phi s\theta c\psi + s\phi s\psi) F_L \right] \\
\ddot{y} = \dfrac{1}{m}\left[\left(s\psi c\theta c\gamma - (c\phi s\theta c\psi - s\phi c\psi) s\gamma \right) T_t + \left(s\psi c\theta \right) F_D + (c\phi s\theta s\psi - s\phi c\psi) F_L \right] \\
\ddot{z} = \dfrac{1}{m}\left[\left(-s\theta c\gamma - c\phi c\theta s\gamma \right) u_1 - s\theta F_D + \left(c\phi c\theta \right) F_L \right] + g \\
\ddot{\phi} = \dfrac{1}{I_{xx}}\left[s\gamma \tau_\phi + \left(I_{yy} - I_{zz} \right) qr - \left(J_r \theta \omega_p \right) s\gamma + M_\phi \right] \\
\ddot{\theta} = \dfrac{1}{I_{yy}}\left[s\gamma \tau_\theta + \left(I_{zz} - I_{xx} \right) pr + J_r \omega_p \left(\phi s\gamma + \psi c\gamma \right) + M_\theta \right] \\
\ddot{\psi} = \dfrac{1}{I_{zz}}\left[s\gamma \tau_\psi + \left(I_{xx} - I_{yy} \right) pq - \left(J_r \theta \omega_p \right) c\gamma + M_\psi \right]
\end{cases}
$$

The control inputs of the QTW :

$$
U = \begin{bmatrix} T_t \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} k & k & k & k \\ kl_s & -kl_s & kl_s & -kl_s \\ kl_l & kl_l & -kl_l & -kl_l \\ k\lambda & -k\lambda & -k\lambda & k\lambda \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}
$$

QTW T-S Fuzzy Modeling (1/4)

Takagi-Sugeno Fuzzy Models

- T-S models are based on linguistic representation rules such as *[Takagi et Sugeno (1985)]* :

***If* premise *Then* consequence**

- If we use singleton fuzzifier, product inference engine and center of gravity defuzzification the T-S fuzzy model can be represented as :

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{N} \mu_i(\theta)\left(A_i\ x(t) + B_i\ u(t)\right) \\ y(t) = \sum_{i=1}^{N} \mu_i(\theta)\ C_i\ x(t) \end{cases} \qquad \sum_{i=1}^{N} \mu_i(x(t),u(t)) = 1 \text{ et } 0 \leq \mu_i(x(t),u(t)) \leq 1$$

Obtaining T-S models

- Identification *[Tanaka et Sugeno (1992)]*
- Linearization around different operating points *[Johansen et al. (2000)]*
- Sectors non linearity approach *[Tanaka et al. (1998)]* : Convex Polytopic Transformation (CPT)

QTW T-S Fuzzy Modeling (2/4)

- **1st step :** the transformation of a nonlinear dynamic model into an LPV model

$$\begin{cases} \dot{x}(t) = A(\theta)\, x(t) + B(\theta)\, u(t) \\ y(t) = C(\theta)\, x(t) \end{cases}$$

- **2nd step :** the convex polytopic transformation is applied for each premise variables :

$$\theta(x,u) = F_0^j \overline{\theta} + F_1^j(\theta)\underline{\theta} \qquad \text{as} \qquad \begin{cases} \overline{\theta} = \max\{\theta(x,u)\} \\ \underline{\theta} = \min\{\theta(x,u)\} \end{cases}$$

with $F_1^j = 1 - \mu_1(\theta) = \dfrac{\overline{\theta} - \theta(x,u)}{\overline{\theta} - \underline{\theta}}$ and $F_0^j = \dfrac{\theta(x,u) - \underline{\theta}}{\overline{\theta} - \underline{\theta}}$

The membership functions :

$$\mu_i(\theta(t)) = \frac{w_i(\theta(t))}{\sum_{i=1}^{r} w_i(\theta(t))} \qquad \text{with} \qquad w_i(\theta(t)) = \prod_{j=1}^{p} w_{i_j}^j(\theta_j)$$

- **3nd step :** The local models constituting the T-S models are given by:

$$A_i = A(\theta(.))\big|_{w_i(.)=1}, B_i = B(\theta(.))\big|_{w_i(.)=1}, C_i = C(\theta(.))\big|_{w_i(.)=1}, D_i = D(\theta(.))\big|_{w_i(.)=1}$$

QTW T-S Fuzzy Modeling (3/4)

- QTW UAV attitude dynamics during the VTOL mode :

$$\begin{cases} \ddot{\phi} = \dfrac{\tau_\phi}{I_{xx}} + \dfrac{I_{yy} - I_{zz}}{I_{xx}} qr - \dfrac{J_r}{I_{xx}} q\omega_p \\ \ddot{\theta} = \dfrac{\tau_\theta}{I_{yy}} + \dfrac{I_{zz} - I_{xx}}{I_{yy}} pr + \dfrac{J_r}{I_{yy}} p\omega_p \\ \ddot{\psi} = \dfrac{\tau_\psi}{I_{zz}} + \dfrac{I_{xx} - I_{yy}}{I_{zz}} pq \end{cases} \quad \text{with} \quad \begin{cases} \phi \in]-\pi/2, +\pi/2[\\ \theta \in]-\pi/2, +\pi/2[\\ \psi \in]-\pi, +\pi[\end{cases}$$

- The quasi-LPV system obtained :

$$\begin{cases} \dot{x}(t) = A(\theta)\, x(t) + B\, u(t) \\ y(t) = C\, x(t) \end{cases}$$

with :

$$A(x) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1\dot{\theta} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2\dot{\phi} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3\dot{\theta} & 0 & 0 \end{bmatrix} ; B = \begin{bmatrix} 0 & 0 & 0 \\ b_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & b_3 \end{bmatrix}$$

$$a_1 = \frac{I_{yy} - I_{zz}}{I_{xx}} ; a_2 = \frac{I_{zz} - I_{xx}}{I_{yy}} ; a_3 = \frac{I_{xx} - I_{yy}}{I_{zz}}$$

QTW T-S Fuzzy Modeling (4/4)

- Based on the sectors non linearity approach two nonlinear continuous terms $(\dot{\phi}, \dot{\theta})$ can be observed.
- The nonlinear model is written as four local linear models $\left(r = 2^p = 2^2\right)$ as :

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1\underline{z}_2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2\underline{z}_1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3\underline{z}_2 & 0 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1\bar{z}_2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2\underline{z}_1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3\bar{z}_2 & 0 & 0 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1\underline{z}_2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2\bar{z}_1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3\underline{z}_2 & 0 & 0 \end{bmatrix} \quad A_4 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1\bar{z}_2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2\bar{z}_1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3\bar{z}_2 & 0 & 0 \end{bmatrix}$$

Particle Swarm Optimization(1/3)

- The PSO algorithm was first described in 1995 by **James Kennedy** and **Russell C. Eberhart** inspired by social behavior of bird flocking or fish schooling.

- PSO is an **artificial intelligence (AI)** technique that can be used to find approximate solutions to extremely difficult or impossible numeric maximization and minimization problems.

- PSO algorithm uses a swarm consisting of $\boldsymbol{n_p}$ particles, randomly distributed in the considered initial search space, to find an optimal solution of a generic **optimization problem**.

Particle Swarm Optimization(2/3)

- In every iteration, each particle is updated by two "**best**" values:
  - ✓ $\boldsymbol{p}_k^i$: the best previously obtained position of the ith particle;
  - ✓ $\boldsymbol{p}_g^i$: the best obtained position in the entire swarm.

- At each algorithm iteration, the ith particle position, $\boldsymbol{x}^i \in \mathrm{R}^d$, evolves based on the following update rules:

$$\boldsymbol{x}_{k+1}^i = \boldsymbol{x}_k^i + \boldsymbol{v}_{k+1}^i$$

$$\boldsymbol{v}_{k+1}^i = w\boldsymbol{v}_k^i + c_1 r_{1,k}^i \left(\boldsymbol{p}_k^i - \boldsymbol{x}_k^i \right) + c_2 r_{2,k}^i \left(\boldsymbol{p}_k^g - \boldsymbol{x}_k^i \right)$$

Where

w : the inertia factor;

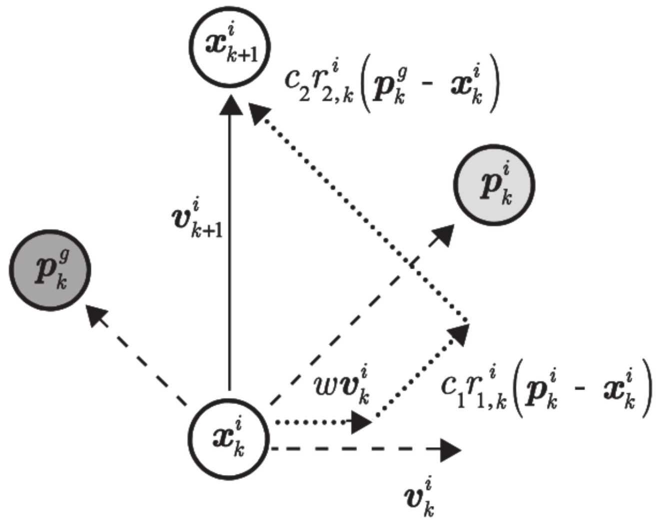
c_1, c_2 : the cognitive and the social scaling factors;

$r_{1,k}^i, r_{2,k}^i$: random numbers uniformly distributed.

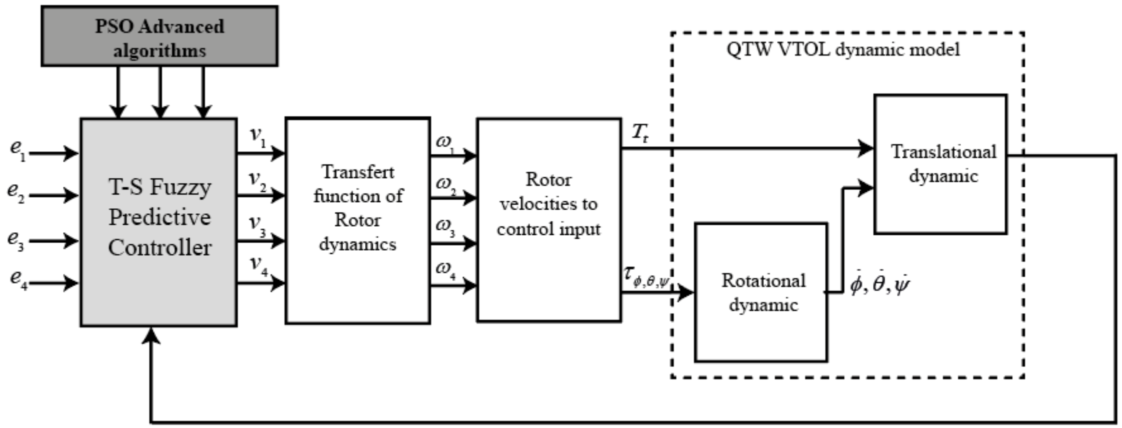
Particle Swarm Optimization(3/3)

Particle Swarm Optimization Algorithm

1. Define all PSO algorithm parameters
2. Randomly initialize the particles positions and velocities . Evaluate the initial population and determine and .
3. Increment the iterations . For each particle, apply the update motion equations
4. Evaluate the corresponding fitness values :
   i. if $\varphi_k^i \leq pbest_k^i$ then $pbest_k^i = \varphi_k^i$ and $\mathbf{p}_k^i = \mathbf{x}_k^i$,
   ii. if $\varphi_k^i \leq gbest_k$ then $gbest_k = \varphi_k^i$ and $\mathbf{p}_k^g = \mathbf{x}_k^i$,
5. If the termination criterion is satisfied, the algorithm terminates with the solution . Otherwise, go to step 3.



TSMPC tuning based PSO



- An optimization based approach for design and tuning of a MPC parameters (N_c, N_p, λ) for each local predictive controller.
- These design parameters present the decision variables of the following multi-objective optimization problem :

$$\begin{cases} \text{minimize} \quad f_i(\boldsymbol{x}) \\ \boldsymbol{x} = (\boldsymbol{N_c}, \boldsymbol{N_p}, \boldsymbol{\lambda})^T \in S \subseteq \mathbb{R}_+^{12} \\ \textit{subject to:} \\ g_1(\boldsymbol{x}) = \delta_\phi - \delta_\phi^{\max} \le 0 \\ g_2(\boldsymbol{x}) = \delta_\theta - \delta_\theta^{\max} \le 0 \\ g_3(\boldsymbol{x}) = \delta_\psi - \delta_\psi^{\max} \le 0 \end{cases}$$

where : $\begin{cases} f_i : \mathbb{R}^{15} \to \mathbb{R} \\ \delta_\phi, \delta_\theta, \delta_\psi \end{cases}$: the cost functions; : the overshoots of the controlled states

Analytical tuning approch proposed

N_c computation (1/2)

- Hessian condition number

$$H = (\Phi^t \Phi + \bar{R})^{-1}$$

$$(n_{in}\acute{N}c \times n_{in}Nc)$$

$$cond(H(k)) = \|H(k)\|_2 \cdot \left\|H(k)^{-1}\right\|_2 = \frac{\sigma_{max}(k)}{\sigma_{min}(k)}$$

- Condition number & stability $N_c \mapsto \infty, N_p \mapsto \infty$

- Concept of effective rank

N_c computation (2/2)

- Method proposed

1) To initialize $N_c \mapsto \infty$ and $N_p \mapsto \infty$ $(Nc < Np)$.
2) To take $A_{ER} = H$ [16].
3) To evaluate Q defined as follows [9]:

$$Q = min\{M_{ER}, N_{ER}\} = min\{n_{in}Nc, n_{in}Nc\}$$
$$= n_{in}Nc.$$

4) To decompose H into singular values and to evaluate $\sigma = [\sigma_1 \quad \sigma_2 \quad \cdots \quad \sigma_\infty]^T$:

$$H = U_H D_H V_H$$
$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_\infty$$

5) To evaluate the singular value distribution p_k with $k = [1, \quad 2, \quad \cdots, \quad \infty]$.
6) To calculate the Shannon entropy [16].
7) To solve $$\begin{cases} N_c^{opt} = round(\frac{e^{H_{Shannon}(p_1, p_2, \cdots, p_\infty)}}{n_{in}}) \\ \min(N_c^{opt}); \quad \min(cond(H(k)) - 1) \end{cases}$$

N_p computation (1/3)

- Closed-loop eigenvalues

$$det[\nu I_{(n_A \times n_A)} - (A - BK_{mpc})] = 0$$

- Optimal closed-loop stability

Insensitivity approach ➡
$$\begin{cases} \min \gamma \\ \begin{bmatrix} -\gamma I_{(n_A \times n_A)} & (A - BK_{mpc}) \\ (A - BK_{mpc})^T & -\gamma I_{(n_A \times n_A)} \end{bmatrix} < 0 \end{cases}$$
Yalmip toolbox ➡ $(K_{mpc})^{opt}$

N_p computation (2/3)

- Relationship between K_{MPC} and $\Phi^T\Phi$

From $K_{mpc} = I_{(n_{in} \times N_c)}(\Phi^T\Phi + \bar{R})^{-1}\Phi^T F$

Under the assumption $(\Phi^T\Phi)^{-1}\bar{R}$ is nilpotent of order n,

$$\begin{aligned} H = & (\Phi^T\Phi)^{-1} - (\Phi^T\Phi)^{-1}\bar{R}(\Phi^T\Phi)^{-1} \\ & + [(\Phi^T\Phi)^{-1}\bar{R}]^2(\Phi^T\Phi)^{-1} - \cdots \\ & + (-1)^{n-1}[(\Phi^T\Phi)^{-1}\bar{R}]^{n-1}(\Phi^T\Phi)^{-1} \end{aligned}$$

⇒

$$\begin{aligned} (\Phi^T\Phi)^{opt} = & [-\bar{R} + \bar{R}(\Phi^T\Phi)^{-1}\bar{R} - \cdots \\ & + (-1)^{n-1}(\Phi^T\Phi)^{-1}[(\Phi^T\Phi)^{-1}\bar{R}]^{n-1}] \\ & \times [(K_{mpc})^{opt}F^T(FF^T)^{-1}\Phi - I_{(n_{in}N_c \times n_{in}Nc)}]^{-1} \end{aligned}$$

N_p computation (3/3)

- Focus on $\Phi^T\Phi$

$$\Phi^T\Phi = \begin{bmatrix} \overbrace{\sum_{i=0}^{N_p-1}(CA^iB)^T(CA^iB)}^{X_{N_p}} & . & . & \cdots & . \\ . & \overbrace{\sum_{i=0}^{N_p-2}(CA^iB)^T(CA^iB)}^{X_{N_p-1}} & . & \cdots & . \\ . & . & \overbrace{\sum_{i=0}^{N_p-3}(CA^iB)^T(CA^iB)}^{X_{N_p-2}} & \cdots & \vdots \\ \vdots & & & \ddots & \\ . & . & . & \cdots & \overbrace{\sum_{i=0}^{N_p-N_c}(CA^iB)^T(CA^iB)}^{X_{N_p-N_c}} \end{bmatrix}$$

- Relationship between $\Phi^T\Phi$ and N_p

$$X_{N_p} = \sum_{i=0}^{N_p^{opt}-1}(CA^iB)^2 = \sum_{i=0}^{N_p^{opt}-1}\left(\sum_{k=0}^{i} C_{mj}A_{mj}^{k}B_{mj}\right)^2$$

$\bar{R}$ computation (1/2)

- Expanded cost fonction [11]

$$J = (Y_{des} - Fx(k))^t (Y_{des} - Fx(k)) \\ - 2\Delta U^t \Phi^T (Y_{des} - Fx(k)) + \Delta U^t (\Phi^t \Phi + \bar{R}) \Delta U$$

- Cost fonction minimization

$$\begin{cases} \dfrac{\partial J}{\partial \Delta U} = 0 \\ \\ \dfrac{\partial J}{\partial \bar{R}} = 0 \end{cases}$$

$\bar{R}$ computation (2/2)

- Cost fonction derivative

$$\frac{\partial J}{\partial \bar{R}} = (\Phi^T\Phi)^{-1}G^TG(\Phi^T\Phi)^{-1} - 2(\Phi^T\Phi)^{-1}\bar{R}(\Phi^T\Phi)^{-1}G^TG(\Phi^T\Phi)^{-1}$$

with
$$\begin{cases} G = \Psi^T\Phi. \\ \Psi = Y_{des} - Fx(k) \end{cases}$$

$\Rightarrow$
$$\bar{R}_{opt} = \frac{1}{2}(\Phi^T\Phi)$$
$\Rightarrow$
$$\bar{R}_{opt} = \begin{bmatrix} \lambda_1^{opt} & 0 & \cdots & 0 \\ 0 & \lambda_2^{opt} & \cdots & \vdots \\ \vdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \lambda_{n_{in}N_c}^{opt} \end{bmatrix}$$

In summary

- Multi-objective issue adressed

$$\min(N_c^{opt})$$

$$\min(cond(H(k)) - 1)$$

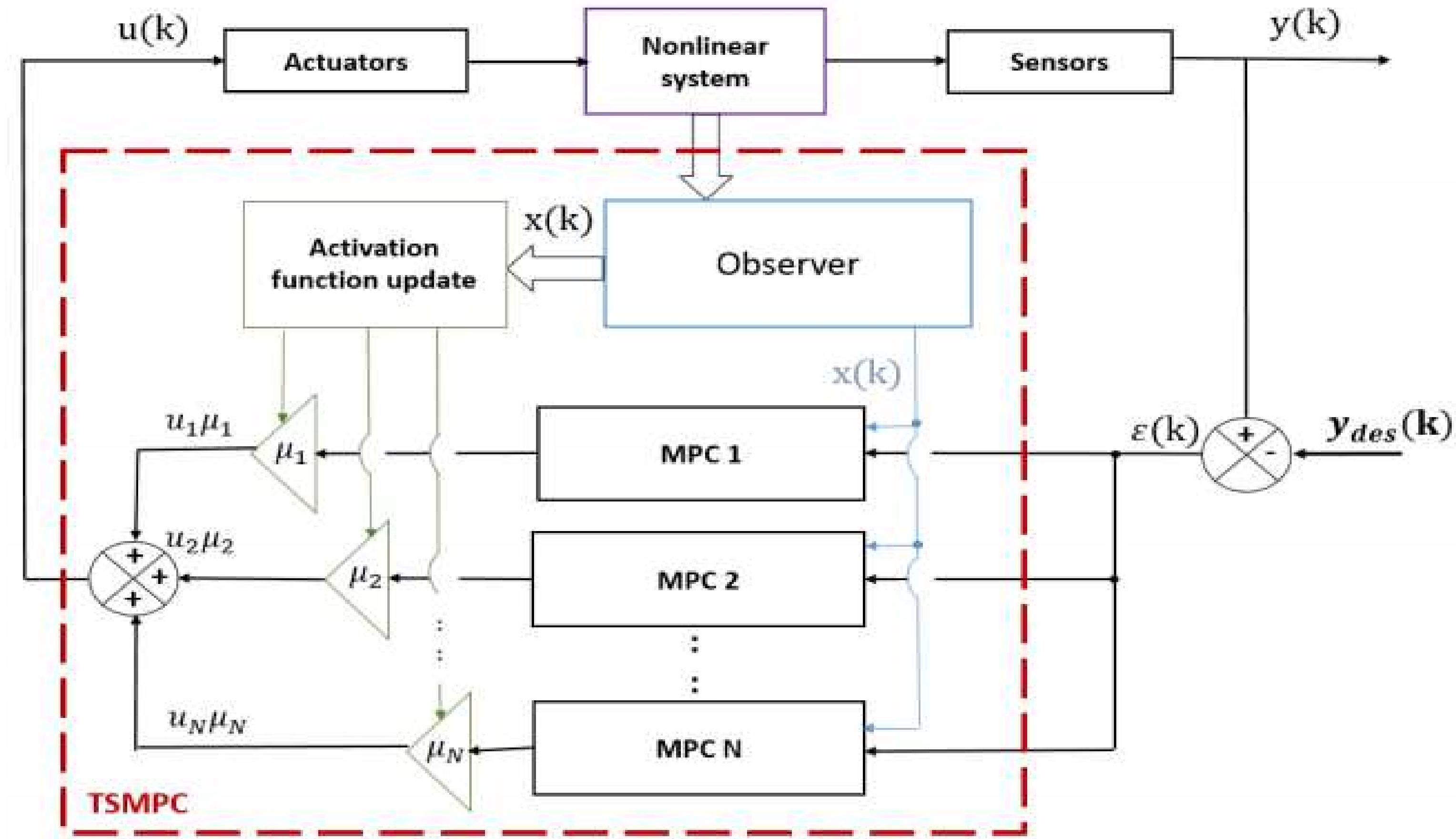
$$N_c^{opt} = round(\frac{e^{H_{Shannon}(p_1, p_2, \cdots, p_\infty)}}{n_{in}})$$

$$X_{N_p} = \sum_{i=0}^{N_p^{opt}-1}(\sum_{k=0}^{i}(C_m A_m^k B_m)^T (C_m A_m^k B_m)$$

$$\bar{R}_{opt} = \frac{1}{2}(\Phi^T \Phi)$$

Case of Nonlinear MIMO System

- MPC Control structure



Application : comparative study

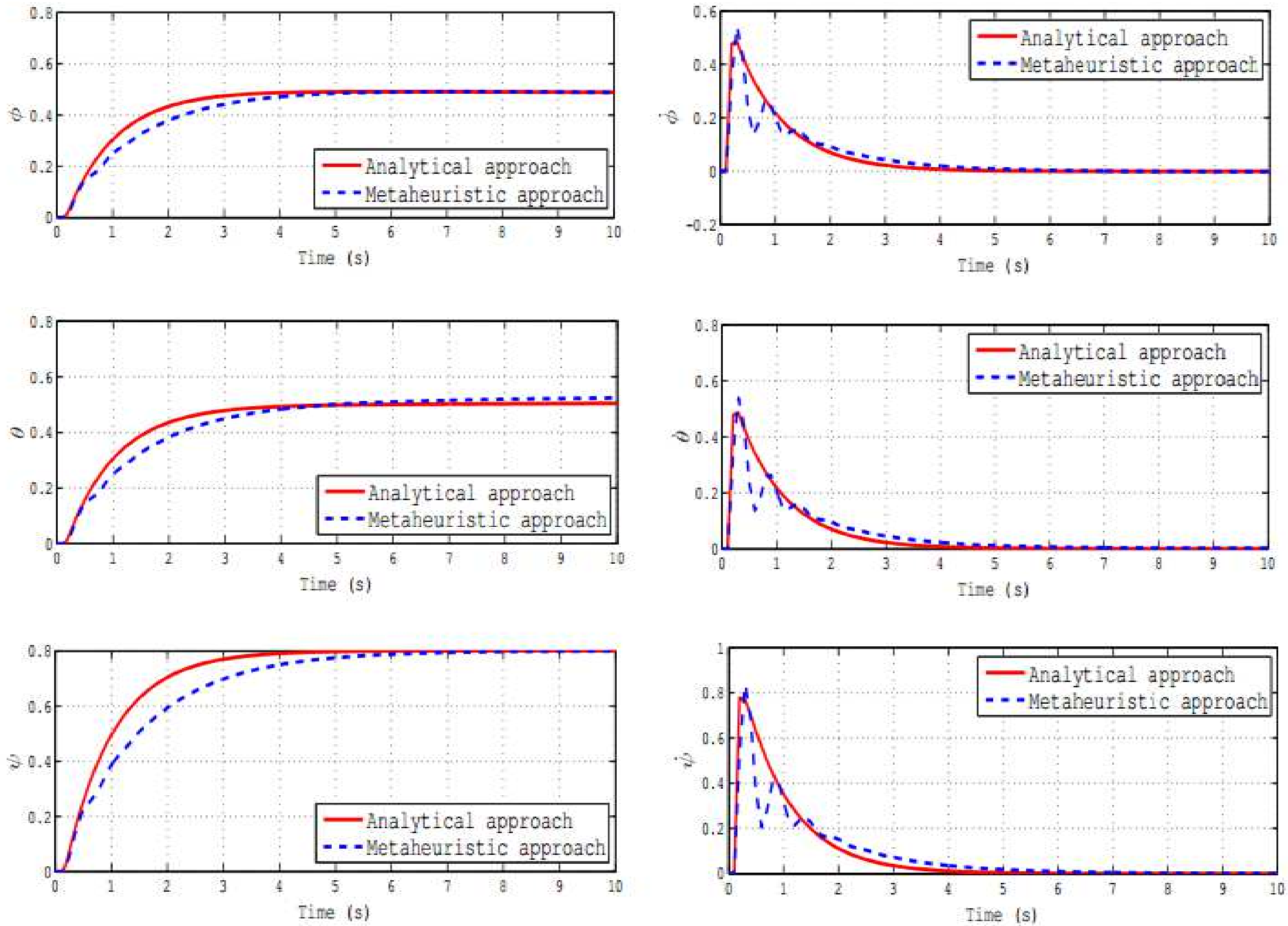
- Computing the weighting factor based on
  - Our metaheuristic approch
  - Our analytical approach

- MPC parameters

	Metaheuristic approach				Analytical approach			
	OA1	OA2	OA3	OA4	OA1	OA2	OA3	OA4
N_c	5	5	5	5	5	5	5	5
N_p	29	29	29	29	29	29	29	29
	0.9450	0.0495	0.1655	3	0.0095	0.0033	0.0133	0.051
λ	1	1.6982	2.2678	2.9962	0.008	0.0024	0.065	1.022
	0.0297	3	0.0018	2.9807	0.0067	0.005	0.0065	0.0065

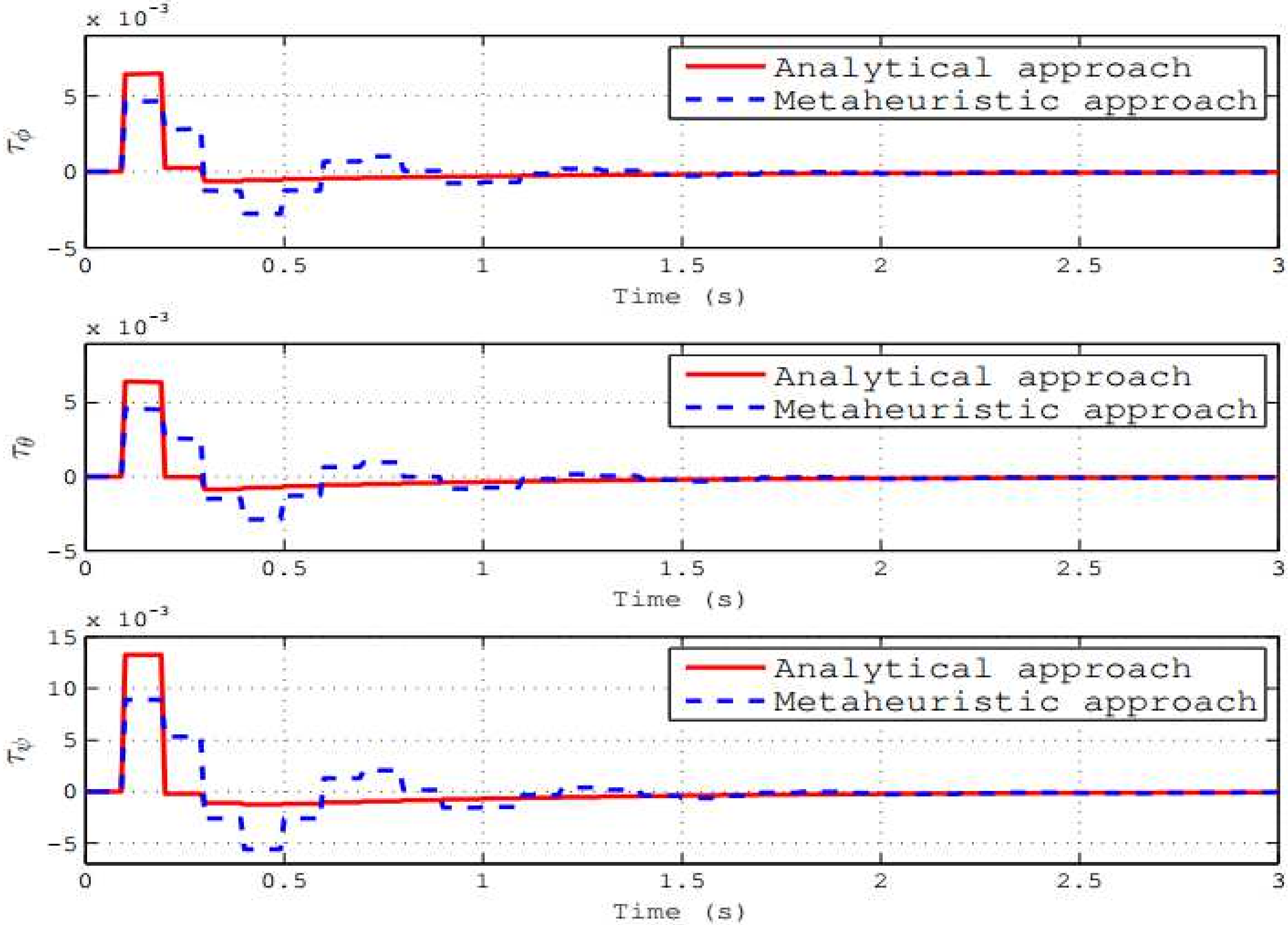
Application : simulation results

- Desired output and system output vs. time



Application : simulation results

- Control output vs. time



Application : perf. comparison (2/3)

- Performance obtained y1

	Metaheuristic approach	Analytical approach
RT (s)	2.1895	1.9747
ST (s)	3.8713	2.9816
OV (%)	1.2457	0
SE	0.0285	0.0091
SDI (%)	1.7764	1.1102
VARU (e^-7)	5.4067	5.1076
CSE (e^-4)	1.6959	1.3961
CEE (e^-4)	2.9198	2.7208

- Performance obtained y2

	Metaheuristic approach	Analytical approach
RT (s)	2.0893	1.3022
ST (s)	3.8225	2.6328
OV (%)	2.1384	1.4898
SE	0.0149	0.0214
SDI (%)	3.1086	3.1086
VARU (e^-7)	5.3987	5.1076
CSE (e^-4)	1.6935	1.3961
CEE (e^-4)	2.9121	2.7208

Application : perf. comparison (3/3)

- Performance obtained y3

	Metaheuristic approach	**Analytical approach**
RT (s)	2.0358	1.3415
ST (s)	3.7998	2.6161
OV (%)	2.0401	0
SE	0.0245	0.0002
SDI (%)	1.9984	1.2273
VARU (e^-7)	5.4068	5.4026
CSE (e^-4)	1.6959	1.5961
CEE (e^-4)	2.9200	2.9107

Conclusion & Outlook

PSO Tuning approach:

- ❑ **Advantages**
  - Applicable to NL MIMO systems
  - Efficient search for the optimal MPC controller parameters
  - Easy implementation

- ❑ **Disadvantages**
  - Offline optimization
  - Number of iteration
  - Don't take into account the disturbances

Analytical Tuning approach:

- ❑ **Advantages**
  - Applicable to NL MIMO systems
  - Optimal closed-loop stability
  - Energy consumption reduced

- ❑ **Disadvantages**
  - Requiert a representative model
  - Requiert an important computational effort to be applied online

Thank you

GT CPNL
ONERA Châtillon
le 04/06/2018

PREDIRE est cofinancé par l'Union européenne.
L'Europe s'engage en Normandie avec le Fonds européen de développement régional

Groupe de Recherche en Electrotechnique et Automatique du Havre

Normandie Université