Operatorial parametrizing of dynamic systems and application to biological problems

E. Montseny[†], A. Doncescu[†]

[†]Laboratoire d'Analyse et d'Architecture des Systèmes, LAAS-CNRS, Toulouse

Contents

1.Introduction

- **Context:**

Nonlinear dynamic problems
$\longrightarrow$ difficulties to treat problems (control etc.) dealing with nonlinear dynamic systems

1.Introduction

- **Context:**

Nonlinear dynamic problems
$\longrightarrow$ difficulties to treat problems (control etc.) dealing with nonlinear dynamic systems

- **Principle presented:**

Methodology to make operatorial transformations of dynamic systems to simplify associated problems
$\longrightarrow$ framwork : functional equations (trajectories)
$\longrightarrow$ simplification of abstract equation by graph parametrizing
$\longrightarrow$ particular case: operatorial parametrizing of dynamic systems
$\longrightarrow$ concret usuable operators and associated tools
$\longrightarrow$ application to fed-batch bioreactor equations

2.Operatorial formulation of dyn. eq.

Classical local formulation of differential equations:

$$\frac{dx(t)}{dt} = f(u(t), x(t)), \ t \in]0, T[$$

with $u(t) \in \mathbb{R}^m$, $x(t) \in \mathbb{R}^n$, $f : \mathbb{R}^m \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$.

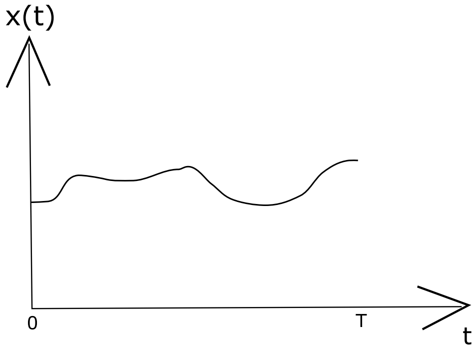
2.Operatorial formulation of dyn. eq.

Classical local formulation of differential equations:

$$\frac{dx(t)}{dt} = f(u(t), x(t)),\ t \in]0, T[$$

with $u(t) \in \mathbb{R}^m$, $x(t) \in \mathbb{R}^n$, $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$.

$\longrightarrow$ Can be considered as trajectorial equations:



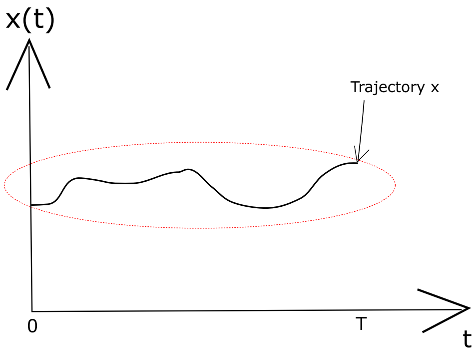
2.Operatorial formulation of dyn. eq.

Classical local formulation of differential equations:

$$\frac{dx(t)}{dt} = f(u(t), x(t)), \; t \in]0, T[$$

with $u(t) \in \mathbb{R}^m$, $x(t) \in \mathbb{R}^n$, $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$.

$\longrightarrow$ Can be considered as trajectorial equations:



2.Operatorial formulation of dyn. eq.

Classical local formulation of differential equations:

$$\frac{dx(t)}{dt} = f(u(t), x(t)),\ t \in]0, T[$$

with $u(t) \in \mathbb{R}^m$, $x(t) \in \mathbb{R}^n$, $f : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^n$.

$\longrightarrow$ Can be considered as trajectorial equations:

$$\partial_t x = f(u, x)$$

where u, x are belonging to functionnal spaces (e.g: $L^2([0, T], \mathbf{U})$, $\mathcal{C}^1([0, T], \mathbf{X})$, etc.)

2.Operatorial formulation of dyn. eq.

Classical local formulation of differential equations:

$$\frac{dx(t)}{dt} = f(u(t), x(t)), \; t \in]0, T[$$

with $u(t) \in \mathbb{R}^m$, $x(t) \in \mathbb{R}^n$, $f : \mathbb{R}^m \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$.

$\longrightarrow$ Can be considered as trajectorial equations:

$$\partial_t x = f(u, x)$$

where u, x are belonging to functionnal spaces (e.g: $L^2([0,T], \mathbf{U})$, $\mathcal{C}^1([0,T], \mathbf{X})$, etc.).

$\longrightarrow$ Trajectories are **points those spaces**, function of those points are **operators**
$\longrightarrow$ More global view, more powerfull
$\longrightarrow$ Richer class of dynamic systems than classical differential systems

<u>Example of dynamic operator</u>: integration operator

$$\partial_t^{-1} : \begin{array}{l} \mathcal{C}^0([0,T], \mathbf{X}) \rightarrow \mathcal{C}^1([0,T], \mathbf{X}) \\ x \longmapsto (\partial_t^{-1} x)(t) = \int_0^t x \end{array}$$

3.Operatorial parametrizing of dyn. sys.

Graph parametrizing of an abstract equation (1)

- We consider the following abstract equation of unknown X, depending on data u:

$$\mathbf{\Phi}(u, X) = 0, \ u \in \mathcal{U}, \ X \in \mathcal{X},$$

with $\mathcal{U}$ and $\mathcal{X}$ two manifolds.

3.Operatorial parametrizing of dyn. sys.

Graph parametrizing of an abstract equation (1)

- We consider the following abstract equation of unknown X, depending on data u:

$$\mathbf{\Phi}(u, X) = 0, \; u \in \mathcal{U}, \, X \in \mathcal{X},$$

with $\mathcal{U}$ and $\mathcal{X}$ two manifolds.

- supposed to be **well-posed**, i.e: there exists a **continuous** application $\mathbf{F}$ such that:

$$X = \mathbf{F}(u)$$

$\longrightarrow$ In general, **F cannot be explicited**, or is **too complex** to be used (with nonlinear equations for example...)

3.Operatorial parametrizing of dyn. sys.

Graph parametrizing of an abstract equation (2)

- Let consider an operator:

$$\mathbf{A} : \mathcal{U} \times \mathcal{X} \to \mathcal{Y}$$

such that $\mathbf{A}_{|\,\mathrm{graph}(\mathbf{F})}$ is an **homeomorphism** between graph($\mathbf{F}$) and $\mathcal{Y}$.

3.Operatorial parametrizing of dyn. sys.

Graph parametrizing of an abstract equation (2)

- Let consider an operator:

$$\mathbf{A} : \mathcal{U} \times \mathcal{X} \to \mathcal{Y}$$

such that $\mathbf{A}_{|\,\mathrm{graph}(\mathbf{F})}$ is an **homeomorphism** between graph($\mathbf{F}$) and $\mathcal{Y}$.

- We denote

$$(\mathbf{B}, \mathbf{C}) := (\mathbf{A}_{|\,\mathrm{graph}(\mathbf{F})})^{-1} : \mathcal{Y} \to \mathcal{U} \times \mathcal{X}$$

  and

$$y := \mathbf{A}(u, X).$$

$\longrightarrow$ Then, any solution of $\mathbf{\Phi}(u, X) = 0$ is parametrized by $y \in \mathcal{Y}$

$\longrightarrow$ Solutions **are directly accessible without resolving** $\mathbf{\Phi}(u, X) = 0$.

3.Operatorial parametrizing of dyn. sys.

Interest of graph parametrizing

- Consider for exemple the constrained optimisation problem:

$$\min_{u \in \mathcal{U}} \{ J(u,X), \quad \mathbf{\Phi}(u,X) = 0 \};$$

3.Operatorial parametrizing of dyn. sys.

Interest of graph parametrizing

- Consider for exemple the constrained optimisation problem:

$$\min_{u \in \mathcal{U}} \{J(u, X), \quad \mathbf{\Phi}(u, X) = 0\};$$

- Then, by using graph parametrizing relations:

$$J(u, X) = J(\mathbf{B}(y), \mathbf{C}(y)) := \tilde{J}(y)$$

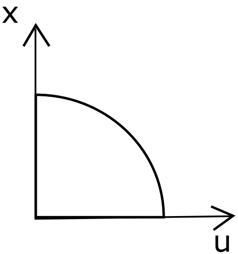
$$\longrightarrow \boxed{\min_{y \in \mathcal{Y}} \tilde{J}(y)}$$

the constraint $\mathbf{\Phi}(u, X) = 0$ being finalelly resumed in the fact that $y \in \mathcal{Y}$.

$\longrightarrow$ **unconstrained** optimisation problem

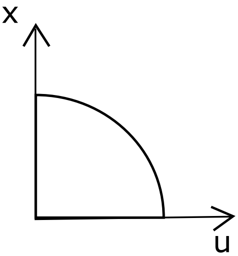
$\longrightarrow$ the solution u^* of initial problem is deduced with $u^* = \mathbf{B}(y^*)$, **without resolving** $\mathbf{\Phi}(u, X) = 0$

3.Operatorial parametrizing of dyn. sys. Link with curve parametrizing



$$x^2 + u^2 = 1,\ x \geq 0,\ u \geq 0$$

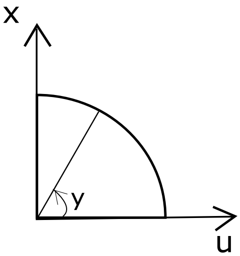
3.Operatorial parametrizing of dyn. sys. Link with curve parametrizing



$$x^2 + u^2 = 1,\ x \geq 0,\ u \geq 0$$

$$\longrightarrow \text{resolution : } x = \sqrt{1 - u^2}$$

3.Operatorial parametrizing of dyn. sys. Link with curve parametrizing

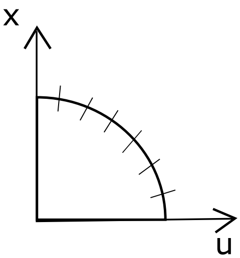


$x^2 + u^2 = 1$, $x \geq 0$, $u \geq 0$

$\longrightarrow$ resolution : $x = \sqrt{1 - u^2}$

$\longrightarrow$ parametrizing : $y = \mathrm{Arctan}(\frac{u}{x})$

3.Operatorial parametrizing of dyn. sys. Link with curve parametrizing



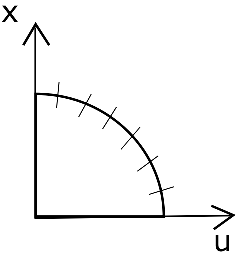
$$x^2 + u^2 = 1,\ x \geq 0,\ u \geq 0$$

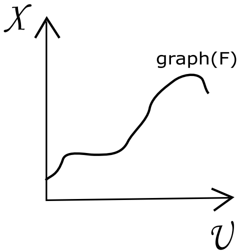
$\longrightarrow$ resolution : $x = \sqrt{1 - u^2}$

$\longrightarrow$ parametrizing : $y = \mathrm{Arctan}(\frac{u}{x})$

$$\Rightarrow \begin{cases} u = \cos(y) \\ x = \sin(y) \end{cases}$$

3.Operatorial parametrizing of dyn. sys. Link with curve parametrizing





$x^2 + u^2 = 1,\ x \geq 0,\ u \geq 0$

$\longrightarrow$ resolution : $x = \sqrt{1-u^2}$

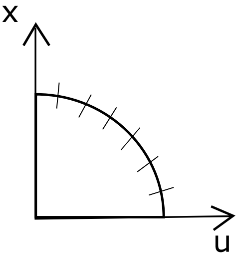
$\longrightarrow$ parametrizing : $y =$Arctan$(\frac{u}{x})$

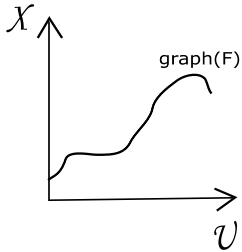
$$\Rightarrow \begin{cases} u = \cos(y) \\ x = \sin(y) \end{cases}$$

$\mathbf{\Phi}(u, X) = 0,\ u \in \mathcal{U},\ X \in \mathcal{X}$

$\longrightarrow x = \mathbf{F}(u)$ (implicit)

3.Operatorial parametrizing of dyn. sys. Link with curve parametrizing





$x^2 + u^2 = 1,\ x \geq 0,\ u \geq 0$

$\longrightarrow$ resolution : $x = \sqrt{1 - u^2}$

$\longrightarrow$ parametrizing : $y =$Arctan$(\frac{u}{x})$

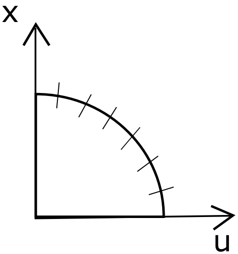
$$\Rightarrow \begin{cases} u = \cos(y) \\ x = \sin(y) \end{cases}$$

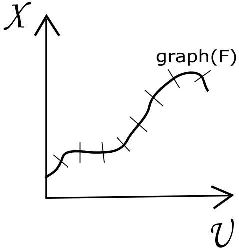
$\mathbf{\Phi}(u, X) = 0,\ u \in \mathcal{U},\ X \in \mathcal{X}$

$\longrightarrow x = \mathbf{F}(u)$ (implicit)

$y := \mathbf{A}(u, X)$ with $\mathbf{A} : \mathcal{U} \times \mathcal{X} \rightarrow \mathcal{Y}$

3.Operatorial parametrizing of dyn. sys. Link with curve parametrizing





$x^2 + u^2 = 1$, $x \geq 0$, $u \geq 0$

$\longrightarrow$ resolution : $x = \sqrt{1 - u^2}$

$\longrightarrow$ parametrizing : $y =$Arctan$(\frac{u}{x})$

$$\Rightarrow \begin{cases} u = \cos(y) \\ x = \sin(y) \end{cases}$$

$\mathbf{\Phi}(u, X) = 0,\ u \in \mathcal{U},\ X \in \mathcal{X}$

$\longrightarrow x = \mathbf{F}(u)$ (implicit)

$y := \mathbf{A}(u, X)$ with $\mathbf{A} : \mathcal{U} \times \mathcal{X} \rightarrow \mathcal{Y}$

$$\Rightarrow \begin{cases} u = \mathbf{B}(y) \\ x = \mathbf{C}(\mathbf{y}) \end{cases}$$

3.Operatorial parametrizing of dyn. sys.

Static example

- Consider the static equation:

$$\mathbf{\Phi}(u, X) = 0 : \left\{ \begin{array}{l} x_2 - u\cos(x_1) = 0 \\ e^{x_2} - x_1 x_2 = 0 \end{array} \right. , \; u \in \mathbb{R}, \; X = (x_1, x_2)^T$$

$\longrightarrow$ hard resolution (i.e: relation $X = \mathbf{F}(u)$ is implicit)

3.Operatorial parametrizing of dyn. sys.

Static example

- Consider the static equation:

$$\mathbf{\Phi}(u,X)=0:\ \begin{cases} x_2-u\cos(x_1)=0 \\ e^{x_2}-x_1x_2=0 \end{cases},\ u\in\mathbb{R},\ X=(x_1,x_2)^T$$

$\longrightarrow$ hard resolution (i.e: relation $X=\mathbf{F}(u)$ is implicit)

$\longrightarrow$ We consider de parametrizing $y=A(u,X)=u\cos(x_1)$

$\longrightarrow$ Then, we have the expression of solutions:

$$\begin{cases} u=\frac{y}{\cos(\frac{e^y}{y})}=\mathbf{B}(y) \\ X=\begin{pmatrix} y \\ \frac{e^y}{y} \end{pmatrix}=\mathbf{C}(y) \end{cases}$$

3.Operatorial parametrizing of dyn. sys.

Static example

- Consider the static equation:

$$\mathbf{\Phi}(u, X) = 0 : \left\{ \begin{array}{l} x_2 - u\cos(x_1) = 0 \\ e^{x_2} - x_1 x_2 = 0 \end{array} \right. , \ u \in \mathbb{R}, \ X = (x_1, x_2)^T$$

$\longrightarrow$ hard resolution (i.e: relation $X = \mathbf{F}(u)$ is implicit)

$\longrightarrow$ We consider de parametrizing $y = A(u, X) = u\cos(x_1)$

$\longrightarrow$ Then, we have the expression of solutions:

$$\left\{ \begin{array}{l} u = \frac{y}{\cos(\frac{e^y}{y})} = \mathbf{B}(y) \\ X = \begin{pmatrix} y \\ \frac{e^y}{y} \end{pmatrix} = \mathbf{C}(y) \end{array} \right.$$

- We want to use the formalism previously introduced to adopt the same approach with dynamic systems

  $\longrightarrow$ numbers "are replaced by" trajectories and, consequently, applications **A**, **B**, **C** will be **operators** on those trajectories.

3.Operatorial parametrizing of dyn. sys.

Particular case of dynamic systems (1)

- Dynamic systems of the form:

$$\mathcal{H}X - G(u, X) = \mathbf{\Phi}(u, X) = 0$$

with $\mathcal{H}$ linear dynamic operator, G static (nonlinear) operator.

3.Operatorial parametrizing of dyn. sys.

Particular case of dynamic systems (1)

- Dynamic systems of the form:

$$\mathcal{H}X - G(u, X) = \mathbf{\Phi}(u, X) = 0$$

with $\mathcal{H}$ linear dynamic operator, G static (nonlinear) operator.

- For example, classical differential systems:

$$\begin{cases} \partial_t X = g(t, u, X), \ t \in]0, T[\\ X(0) = X_0, \end{cases}$$

are particular cases by denoting:

$$\mathcal{H} = \begin{pmatrix} \partial_t \\ < \delta, \cdot > \end{pmatrix}, \ G(u, X) = \begin{pmatrix} g(t, u, X) \\ X_0 \end{pmatrix},$$

where $< \delta, \cdot >$ is the dirac distribution operator, ∂_t is the time-derivative operator, $\mathcal{U}$ and $\mathcal{X}$ are manifolds of functionnal spaces, for example $\mathcal{U} \subset L^\infty(0, T; \mathbb{R}^m), \ \mathcal{X} \subset C^0([0, T]; \mathbb{R}^n)$.

3.Operatorial parametrizing of dyn. sys.

Particular case of dynamic systems (2)

- The previous formulation allows to consider more genreral dynamic systems by remplacing ∂_t by a convolutive dynamic operator $H(\partial_t)$, non necessary time-local

  $\longrightarrow$ larger class of nonlinear dynamic systems (Volterra, PDE's, hybrid systems etc.)

  $\longrightarrow$ richer possibilities of transformations

  $\longrightarrow$ a mathematical framework must be chosen to manipulate operators: addition, composition, inversion etc., that is an **algebra** of **operators**

3.Operatorial parametrizing of dyn. sys.

Particular case of dynamic systems (2)

- The previous formulation allows to consider more genreral dynamic systems by remplacing ∂_t by a convolutive dynamic operator $H(\partial_t)$, non necessary time-local

  $\longrightarrow$ larger class of nonlinear dynamic systems (Volterra, PDE's, hybrid systems etc.)

  $\longrightarrow$ richer possibilities of transformations

  $\longrightarrow$ a mathematical framework must be chosen to manipulate operators: addition, composition, inversion etc., that is an **algebra** of **operators**

- Dynamic system are often associated to control problems, identification problems etc.

  $\longrightarrow$ Aim : find judicous **operatorial parametrizing** and **operatorial transformations** of the dynamic system that **simplify the resolution** of the problem

  $\longrightarrow$ specially interesting when the dynamic system is nonlinear and hard to deal with

4.Concretely usuable operators

Brief summary (1)

Dynamic equations under the form:

$$\mathbf{\Phi}(u, X) = 0,\ u \in \mathcal{U}, X \in \mathcal{X}$$

4.Concretely usuable operators

Brief summary (1)

Dynamic equations under the form:

$$\mathbf{\Phi}(u, X) = 0,\ \ u \in \mathcal{U}, X \in \mathcal{X}$$

Parametrizing :

$$y = \mathbf{A}(u, X)$$

$$\longrightarrow \ \left\{ \begin{array}{l} u = \mathbf{B}(y) \\ X = \mathbf{C}(y) \end{array} \right.$$

4.Concretely usuable operators

Brief summary (1)

Dynamic equations under the form:

$$\mathbf{\Phi}(u, X) = 0, \;\; u \in \mathcal{U}, X \in \mathcal{X} \tag{1}$$

Parametrizing :

$$y = \mathbf{A}(u, X)$$

$$\longrightarrow \;\; \begin{cases} u = \mathbf{B}(y) \\ X = \mathbf{C}(y) \end{cases} \tag{2}$$

Then a control problem on (1), **for example**:

$$\min_{u \in \mathcal{U}} \{ J(u, X), \quad \mathbf{\Phi}(u, X) = 0 \}$$

becomes:

$$\min_{y \in \mathcal{Y}} \tilde{J}(y)$$

$\longrightarrow$ Resolution of this simplified problem in y

$\longrightarrow$ Deduction of corresponding command u from (2) **without resolving** (1)

4.Concretely usuable operators

Brief summary (2)

Of course, the **choice of parametrizing operator** is important, because any parametrizing doesn't necessary simplify the problem, at least for two reasons:

1. The problem in y:

   $$\min_{y \in \mathcal{Y}} \tilde{J}(y) \quad (\text{with } \tilde{J}(y) := J(\mathbf{B}(y), \mathbf{C}(y)))$$

   must be concretely soluble

2. u and X must me concretely deduced from y

$\longrightarrow$ Operators **B** and **C must be practicable**

$\longrightarrow$ **They must lead to a class of operators** which we can deal with numerically, with eventual contraint of computation cost for real-time applications

4.Concretely usuable operators

Classe of usuable operators

Principal classes of interesting operators (algebras in fact):

- **Static operators**: $(\mathbf{G}(f))(t) = G(t, g(t))$ with G classical function

  $\longrightarrow$ very simple to evaluate (as cheap as evaluating a function)

4.Concretely usuable operators

Classe of usuable operators

Principal classes of interesting operators (algebras in fact):

- **Static operators**: $(\mathbf{G}(f))(t) = G(t, g(t))$ with G classical function

  $\longrightarrow$ very simple to evaluate (as cheap as evaluating a function)

- **Convolution opertors**: large class of **linear** operators, often denoted $H(\partial_t)$. Their action on a function u, denoted $H(\partial_t)u$, can be evaluated with a reasonable cost with *Fast Fourrier Transform* or *Diffusive Representation* (compatible with real-time applications)

4.Concretely usuable operators

Classe of usuable operators

Principal classes of interesting operators (algebras in fact):

- **Static operators**: $(\mathbf{G}(f))(t) = G(t, g(t))$ with G classical function

  $\longrightarrow$ very simple to evaluate (as cheap as evaluating a function)

- **Convolution opertors**: large class of **linear** operators, often denoted $H(\partial_t)$. Their action on a function u, denoted $H(\partial_t)u$, can be evaluated with a reasonable cost with *Fast Fourrier Transform* or *Diffusive Representation* (compatible with real-time applications)

- **Time-scaling transformations** (TST) operators (cf next slide)

4.Concretely usuable operators

Classe of usuable operators

Principal classes of interesting operators (algebras in fact):

- **Static operators**: $(\mathbf{G}(f))(t) = G(t, g(t))$ with G classical function

  $\longrightarrow$ very simple to evaluate (as cheap as evaluating a function)

- **Convolution opertors**: large class of **linear** operators, often denoted $H(\partial_t)$. Their action on a function u, denoted $H(\partial_t)u$, can be evaluated with a reasonable cost with *Fast Fourrier Transform* or *Diffusive Representation* (compatible with real-time applications)

- **Time-scaling transformations** (TST) operators (cf next slide)

- Others operators like t-local realisation operators (diffusive operators), quasi-static operators etc.

4.Concretely usuable operators

Classe of usuable operators

Principal classes of interesting operators (algebras in fact):

- **Static operators**: $(\mathbf{G}(f))(t) = G(t, g(t))$ with G classical function

  $\longrightarrow$ very simple to evaluate (as cheap as evaluating a function)

- **Convolution opertors**: large class of **linear** operators, often denoted $H(\partial_t)$. Their action on a function u, denoted $H(\partial_t)u$, can be evaluated with a reasonable cost with *Fast Fourrier Transform* or *Diffusive Representation* (compatible with real-time applications)

- **Time-scaling transformations** (TST) operators (cf next slide)

- Others operators like t-local realisation operators (diffusive operators), quasi-static operators etc.

- and, of course any finite combinaison of those kind of operators

4.Concretely usuable operators

Classe of usuable operators

Principal classes of interesting operators (algebras in fact):

- **Static operators**: $(\mathbf{G}(f))(t) = G(t, g(t))$ with G classical function

  $\longrightarrow$ very simple to evaluate (as cheap as evaluating a function)

- **Convolution opertors**: large class of **linear** operators, often denoted $H(\partial_t)$. Their action on a function u, denoted $H(\partial_t)u$, can be evaluated with a reasonable cost with *Fast Fourrier Transform* or *Diffusive Representation* (compatible with real-time applications)

- **Time-scaling transformations** (TST) operators (cf next slide)

- Others operators like t-local realisation operators (diffusive operators), quasi-static operators etc.

- and, of course any finite combinaison of those kind of operators

$\longrightarrow$ A judicious parametrizing must (when possible of course) transform the **nonlinear dynamic** problem (ie: $\mathbf{F}$) into a problem that deals only with finite combinaison of **dynamic linear** operators, **static nonlinear** operators etc.

4.Concretely usuable operators

Few words on Time-Scaling Transformations (TST)

- Some problems can present an **intrinsic time** under which equations are simplified
- Classical change of time is an **operator**:

$$x \mapsto x \circ \varphi$$

where $\varphi(t)$ is the new time scale, φ is an increasing function.

4.Concretely usuable operators

Few words on Time-Scaling Transformations (TST)

- Some problems can present an **intrinsic time** under which equations are simplified
- Classical change of time is an **operator**:

$$x \mapsto x \circ \varphi$$

where $\varphi(t)$ is the new time scale, φ is an increasing function.

- We consider that φ can be an **operatorial transformation** of a function v that **pilot** the clock, that means $\varphi = \Phi(v)$ with Φ an operator.

  $\longrightarrow$ Moreover, the **TST can directly depends on variables of the problem** (for example u and/or X)

  $\longrightarrow$ Rich class of TST

- Example: $S : (v, x) \mapsto x \circ (\partial_t^{-1} v)^{-1}$

5.Application to fed-batch bioreactors equations

Model under consideration

System of differential equations of fed-batch bioreactor:

$$\left\{\begin{array}{l} \partial_t x = \mu(X)\,x - x\,u \\ \partial_t s = -a_1\mu(X)\,x + (s_i - s)\,u \\ \partial_t p = a_2\mu(X)\,x - p\,u \\ X(0) = X_0, \end{array}\right. \quad \Leftrightarrow \quad \boldsymbol{\Phi}(\underbrace{u, X_0}_{\text{data of the problem}}, X) = 0$$

with $X := (x, s, p)^T$, x, s, p the respectives concentrations of biomass, substrate and product, $\mu(X)$ the growth rate (Monod etc.), s_i the substrate concentration in feed, u (the command) the dilution of feed and X_0 initial conditions.

$\longrightarrow$ nonlinear dynamic system

$\longrightarrow$ difficulties to develop control techniques of such a model (optimization of product production, that is optimal control, etc.)

5.Application to fed-batch bioreactors equations

Model under consideration

System of differential equations of fed-batch bioreactor:

$$\left\{\begin{array}{l}\partial_t x = \mu(X)\, x - x\, u \\ \partial_t s = -a_1 \mu(X)\, x + (s_i - s)\, u \\ \partial_t p = a_2 \mu(X)\, x - p\, u \\ X(0) = X_0,\end{array}\right. \quad \Leftrightarrow \quad \begin{array}{c}\mathbf{\Phi}(\underbrace{u, X_0}, X) = 0 \\ \text{data of the problem}\end{array}$$

with $X := (x, s, p)^T$, x, s, p the respectives concentrations of biomass, substrate and product, $\mu(X)$ the growth rate (Monod etc.), s_i the substrate concentration in feed, u (the command) the dilution of feed and X_0 initial conditions.

$\longrightarrow$ nonlinear dynamic system

$\longrightarrow$ difficulties to develop control techniques of such a model (optimization of product production, that is optimal control, etc.)

$\longrightarrow$ a judicious parametrizing (using TST) **leads to a rather simple equivalent system**

5.Application to fed-batch bioreactors equations

Time transformation of bioreactor equations

- We consider a time-scale changing φ such that $\partial_t \varphi = u > 0$, with $\varphi(0) = 0$

$\longrightarrow$ φ is an increasing function
$\longrightarrow$ we remark that this change of time depends on the command u of the system
$\longrightarrow$ Remark: it is equivalent to say: $\varphi = \partial_t^{-1} u$

5.Application to fed-batch bioreactors equations

Time transformation of bioreactor equations

- We consider a time-scale changing φ such that $\partial_t \varphi = u > 0$, with $\varphi(0) = 0$

$\longrightarrow$ φ is an increasing function
$\longrightarrow$ we remark that this change of time depends on the command u of the system
$\longrightarrow$ Remark: it is equivalent to say: $\varphi = \partial_t^{-1} u$

- By denoting by $\widetilde{\cdot}$ the quantities after change of time (i.e: in time τ), and using classical differential relation $\frac{dx}{dt} = \frac{dx}{d\tau}\frac{d\tau}{dt}$, we remark this time-scale changing leads to the following differential system:

$$\begin{cases} \partial_\tau \widetilde{x} = -\widetilde{x} + \frac{\mu(\widetilde{X})\widetilde{x}}{\widetilde{u}} \\ \partial_\tau \widetilde{s} = -\widetilde{s} + s_i - a_1 \frac{\mu(\widetilde{X})\widetilde{x}}{\widetilde{u}} \\ \partial_\tau \widetilde{p} = -\widetilde{p} + a_2 \frac{\mu(\widetilde{X})\widetilde{x}}{\widetilde{u}}, \end{cases}$$

$\longrightarrow$ after changing time to **intrinsic biological time**, **governed by dilution of feed**, by the system is simplified

$\longrightarrow$ It appears that the quantity $\frac{\mu(\widetilde{X})\widetilde{x}}{\widetilde{u}}$ can be a judicious parametrizing

5.Application to fed-batch bioreactors equations

Associated TST operator

We define the operator of TST previously used by:

$$\mathbf{S} : (u, x) \longmapsto \widetilde{x} := x \circ \left(\partial_t^{-1} u\right)^{-1}$$

$\longrightarrow$ the "reversal" TST operator $S^{-1}(u, \widetilde{x}) \longmapsto x$ is given by:

$$x = \widetilde{x} \circ \partial_t^{-1} u$$

$\longrightarrow$ we can also express those transormation depending on $\widetilde{u}$:

$$\widetilde{x} = x \circ \partial_\tau^{-1} \frac{1}{\widetilde{u}}$$

$$x = \widetilde{x} \circ \left(\partial_\tau^{-1} \frac{1}{\widetilde{u}}\right)^{-1}$$

Remark : those relations can (fortunatelly) be applied to u and $\widetilde{u}$

5.Application to fed-batch bioreactors equations

Operatorial parametrization (1)

We define the following parametrization by y of bioreactor equations:

$$\mathbf{A} : (u, x) \longmapsto y = (\widetilde{\mathbf{A}} \circ \mathbf{S})(u, X)$$

with

$$\widetilde{\mathbf{A}} : (\widetilde{u}, \widetilde{X}) \longmapsto \left(\frac{\mu(\widetilde{X})\widetilde{x}}{\widetilde{u}}, \langle \delta, \widetilde{x} \rangle, \langle \delta, \widetilde{s} \rangle, \langle \delta, \widetilde{p} \rangle \right)^T$$

5.Application to fed-batch bioreactors equations

Operatorial parametrization (1)

We define the following parametrization by y of bioreactor equations:

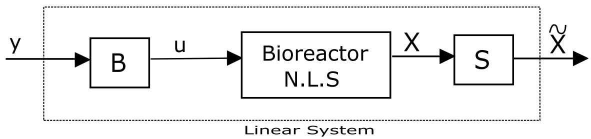
$$\mathbf{A} : (u, x) \longmapsto y = (\widetilde{\mathbf{A}} \circ \mathbf{S})(u, X)$$

with

$$\widetilde{\mathbf{A}} : (\widetilde{u}, \widetilde{X}) \longmapsto \left(\frac{\mu(\widetilde{X})\widetilde{x}}{\widetilde{u}}, \langle \delta, \widetilde{x} \rangle, \langle \delta, \widetilde{s} \rangle, \langle \delta, \widetilde{p} \rangle \right)^T$$

$\longrightarrow$ Under this parametrizing, the system of equations becomes:

$$\begin{cases} \partial_\tau \widetilde{x} = -\widetilde{x} + y_1 \\ \partial_\tau \widetilde{s} = -\widetilde{s} + s_i - a_1 y_1 \\ \partial_\tau \widetilde{p} = -\widetilde{p} + a_2 y_1, \end{cases}$$



$\longrightarrow$ The **initial nonlinear system** has been transformed into an **equivalent linear one** after the operatorial parametrizing $y = \mathbf{A}(u, X)$

$\longrightarrow$ Classical methods can be investigated on this equivalent system: stabilization, regulation, optimal control etc.

5.Application to fed-batch bioreactors equations

Operatorial parametrization (2)

Operators $(\mathbf{B}, \mathbf{C})$ associated are given by:

$$\left\{\begin{array}{l} \partial_\tau \widetilde{x} = -\widetilde{x} + y_1 \\ \partial_\tau \widetilde{s} = -\widetilde{s} + s_i - a_1 y_1 \\ \partial_\tau \widetilde{p} = -\widetilde{p} + a_2 y_1, \\ \\ + \ y = \mathbf{A}(u, X) \end{array}\right. \Rightarrow \left\{\begin{array}{l} \begin{pmatrix} u \\ X_0 \end{pmatrix} = \mathbf{B}(y) = \mathbf{S}^{-1} \circ \widetilde{\mathbf{B}} \\ X = \mathbf{C}(y) = \mathbf{S}^{-1} I_3 \circ \widetilde{\mathbf{C}}, \end{array}\right.$$

with:

$$\widetilde{\mathbf{C}}(y) = \begin{pmatrix} (\partial_\tau + 1)^{-1}(y_1) + y_2\, e^{-\cdot} \\ (\partial_\tau + 1)^{-1}(s_i - a_1\, y_1) + y_3\, e^{-\cdot} \\ (\partial_\tau + 1)^{-1}(a_2\, y_1) + y_4\, e^{-\cdot} \end{pmatrix}$$

$$\widetilde{\mathbf{B}}(y) = \begin{pmatrix} \frac{\mu(\mathbf{C}(y))\, \mathbf{C}_1(y)}{y_1} \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}.$$

$\longrightarrow$ All involved operators are finite combinaison of static/linear dynamic/TST operators
$\longrightarrow$ Concretely usuable

5.Application to fed-batch bioreactors equations

Operatorial parametrization (3)

- After operatorial parametrizing, classical control problems can be treated on this fed-batch model. For example:

$$\max_{u\in\mathcal{U}}\{J(u,X), \quad \mathbf{\Phi}(u,X)=0\}$$

with $J(u,X)=p(T)$ (i.e: optimization of the production of the product), becomes:

$$\max_{y\in\mathcal{Y}}(\mathbf{C}_3(y))(T)$$

$\longrightarrow$ **linear** objective function (because $\mathbf{C}_3$ is a linear dynamic operator), to **optimize without contraint**
$\longrightarrow$ determination of y^*
$\longrightarrow$ $u^*=\mathbf{B}(y^*)$: optimal open-loop command

- Many other possibilities: trajectory planification around optimal command, closed-loop stabilisation, etc.

5.Application to fed-batch bioreactors equations

Example of closed-loop using operatorial parametrizing