Systèmes entrée-sortie non linéaires et applications en audio-acoustique

Séries de Volterra

Thomas Hélie, CNRS

Equipe S3AM (`http://s3am.ircam.fr`)
Laboratoire des Sciences et Technologies de la Musique et du Son
IRCAM - CNRS - SU, Paris, France

Ecole Thématique ”Théorie du Contrôle en Mécanique”
2019

Plan

1. Préambule
2. Séries de Volterra : généralités
3. Calcul des noyaux de Volterra d'un système différentiel
4. Exercices et applications en audio-acoustique
5. Convergence
6. Extension en dimension infinie et application
7. Conclusion

Plan

1. Préambule
2. Séries de Volterra : généralités
3. Calcul des noyaux de Volterra d'un système différentiel
4. Exercices et applications en audio-acoustique
5. Convergence
6. Extension en dimension infinie et application
7. Conclusion

Plan

1. Préambule
2. Séries de Volterra : généralités
3. Calcul des noyaux de Volterra d'un système différentiel
4. Exercices et applications en audio-acoustique
5. Convergence
6. Extension en dimension infinie et application
7. Conclusion

Plan

1 Préambule

2 Séries de Volterra : généralités

3 Calcul des noyaux de Volterra d'un système différentiel
- Lois d'interconnexion *(somme, produit, cascade)*
- Calcul algébrique de noyaux de transfert
- Réalisation numérique

4 Exercices et applications en audio-acoustique

5 Convergence

6 Extension en dimension infinie et application

7 Conclusion

Outline

1 Préambule

2 Séries de Volterra : généralités

3 Calcul des noyaux de Volterra d'un système différentiel
- Lois d'interconnexion *(somme, produit, cascade)*
- Calcul algébrique de noyaux de transfert
- Réalisation numérique

4 Exercices et applications en audio-acoustique

5 Convergence

6 Extension en dimension infinie et application

7 Conclusion

Interconnection laws: SUM

Computing $y(t) = y_a(t) + y_b(t)$

Interconnection laws: SUM

Computing $y(t) = y_a(t) + y_b(t)$

$$
\begin{aligned}
y(t) &= \sum_{n=1}^{+\infty} \int_{\mathbb{R}^n} a_n(\tau_{1:n}) u(t-\tau_1) \ldots u(t-\tau_n)\, \mathrm{d}\tau_1 \ldots \mathrm{d}\tau_n \\
&\quad + \sum_{n=1}^{+\infty} \int_{\mathbb{R}^n} b_n(\tau_{1:n}) u(t-\tau_1) \ldots u(t-\tau_n)\, \mathrm{d}\tau_1 \ldots \mathrm{d}\tau_n
\end{aligned}
$$

Interconnection laws: SUM

Computing $y(t) = y_a(t) + y_b(t)$

$$
\begin{aligned}
y(t) &= \textstyle\sum_{n=1}^{+\infty} \int_{\mathbb{R}^n} a_n(\tau_{1:n}) u(t-\tau_1)\ldots u(t-\tau_n)\,\mathrm{d}\tau_1\ldots\mathrm{d}\tau_n \\
&\quad + \textstyle\sum_{n=1}^{+\infty} \int_{\mathbb{R}^n} b_n(\tau_{1:n}) u(t-\tau_1)\ldots u(t-\tau_n)\,\mathrm{d}\tau_1\ldots\mathrm{d}\tau_n \\
&= \sum_{n=1}^{+\infty} \int_{\mathbb{R}^n} \left[a_n(\tau_{1:n}) + b_n(\tau_{1:n})\right] u(t-\tau_1)\ldots u(t-\tau_n)\,\mathrm{d}\tau_1\ldots\mathrm{d}\tau_n
\end{aligned}
$$

Interconnection laws: SUM

Computing $y(t) = y_a(t) + y_b(t)$

$$
\begin{aligned}
y(t) &= \textstyle\sum_{n=1}^{+\infty} \int_{\mathbb{R}^n} a_n(\tau_{1:n}) u(t-\tau_1)\dots u(t-\tau_n)\, d\tau_1 \dots d\tau_n \\
&\quad + \textstyle\sum_{n=1}^{+\infty} \int_{\mathbb{R}^n} b_n(\tau_{1:n}) u(t-\tau_1)\dots u(t-\tau_n)\, d\tau_1 \dots d\tau_n \\
&= \sum_{n=1}^{+\infty} \int_{\mathbb{R}^n} \left[a_n(\tau_{1:n}) + b_n(\tau_{1:n})\right] u(t-\tau_1)\dots u(t-\tau_n)\, d\tau_1 \dots d\tau_n
\end{aligned}
$$

Result: Equivalent kernels c_n

$$c_n(\tau_{1:n}) = a_n(\tau_{1:n}) + b_n(\tau_{1:n})$$

Laplace T.: $$C_n(s_{1:n}) = A_n(s_{1:n}) + B_n(s_{1:n})$$

PRODUCT

Computing $y(t) = y_a(t)\, y_b(t)$

PRODUCT

Computing $y(t) = y_a(t)\, y_b(t)$

$$y(t) = \sum_{p=1}^{+\infty} \int_{\mathbb{R}^p} a_p(\theta_{1:p}) u(t-\theta_1)\dots u(t-\theta_p)\, d\theta_1 \dots d\theta_p$$
$$\times \sum_{q=1}^{+\infty} \int_{\mathbb{R}^q} b_q(\sigma_{1:q}) u(t-\sigma_1)\dots u(t-\sigma_q)\, d\sigma_1 \dots d\sigma_q$$

PRODUCT

Computing $y(t) = y_a(t)\, y_b(t)$

$$
\begin{aligned}
y(t) &= \textstyle\sum_{p=1}^{+\infty} \int_{\mathbb{R}^p} a_p(\theta_{1:p}) u(t-\theta_1) \dots u(t-\theta_p)\, \mathrm{d}\theta_1 \dots \mathrm{d}\theta_p \\
&\quad \times \textstyle\sum_{q=1}^{+\infty} \int_{\mathbb{R}^q} b_q(\sigma_{1:q}) u(t-\sigma_1) \dots u(t-\sigma_q)\, \mathrm{d}\sigma_1 \dots \mathrm{d}\sigma_q \\
&= \sum_{n=1}^{+\infty} \int_{\mathbb{R}^n} \Big[\sum_{\substack{p,q\geq 1 \\ p+q=n}} a_p(\theta_{1:p})\, b_q(\sigma_{1:q}) \Big] u(t-\theta_1) \dots u(t-\theta_p) \\
&\qquad u(t-\sigma_1) \dots u(t-\sigma_q)\, \mathrm{d}\theta_1 \dots \mathrm{d}\theta_p\, \mathrm{d}\sigma_1 \dots \mathrm{d}\sigma_q
\end{aligned}
$$

PRODUCT

Computing $y(t) = y_a(t)\, y_b(t)$

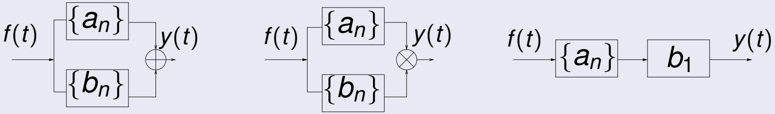
$$
\begin{aligned}
y(t) &= \sum_{p=1}^{+\infty} \int_{\mathbb{R}^p} a_p(\theta_{1:p}) u(t-\theta_1)\dots u(t-\theta_p)\, d\theta_1 \dots d\theta_p \\
&\quad \times \sum_{q=1}^{+\infty} \int_{\mathbb{R}^q} b_q(\sigma_{1:q}) u(t-\sigma_1)\dots u(t-\sigma_q)\, d\sigma_1 \dots d\sigma_q \\
&= \sum_{n=1}^{+\infty} \int_{\mathbb{R}^n} \Big[\sum_{\substack{p,q\geq 1 \\ p+q=n}} a_p(\theta_{1:p})\, b_q(\sigma_{1:q}) \Big] u(t-\theta_1)\dots u(t-\theta_p) \\
&\qquad u(t-\sigma_1)\dots u(t-\sigma_q)\, d\theta_1 \dots d\theta_p\, d\sigma_1 \dots d\sigma_q
\end{aligned}
$$

Result: Equivalent kernels c_n

$$c_n(\tau_{1:n}) = \sum_{p=1}^{n-1} a_p(\tau_{1:p})\, b_{n-p}(\tau_{p+1:n})$$

Laplace T.: $$C_n(s_{1:n}) = \sum_{p=1}^{n-1} A_p(s_{1:p})\, B_{n-p}(s_{p+1:n})$$

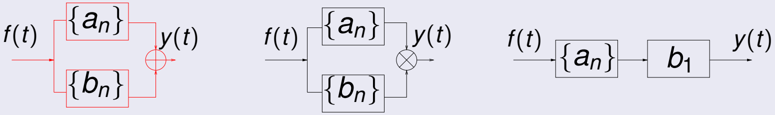
Interconnection laws: Sum, Product and Cascade



Equivalent transfer kernels C_n

where $\mathbb{M}_n^m = \{(p_1, \ldots, p_m) \in (\mathbb{N}^*)^m$ s.t. $p_1 + \cdots + p_m = n\}$

Interconnection laws: Sum, Product and Cascade

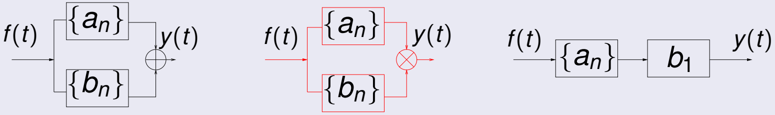


Equivalent transfer kernels C_n

Sum: $C_n(s_{1:n}) = A_n(s_{1:n}) + B_n(s_{1:n})$

where $\mathbb{M}_n^m = \{(p_1, \ldots, p_m) \in (\mathbb{N}^*)^m \text{ s.t. } p_1 + \cdots + p_m = n\}$

Interconnection laws: Sum, Product and Cascade



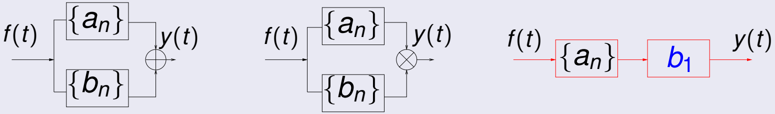
Equivalent transfer kernels C_n

$$\text{Sum:}\quad C_n(s_{1:n}) = A_n(s_{1:n}) + B_n(s_{1:n})$$

$$\text{Product:}\quad C_n(s_{1:n}) = \sum_{p=1}^{n-1} A_p(s_{1:p}) B_{n-p}(s_{p+1:n})$$

where $\mathbb{M}_n^m = \{(p_1, \dots, p_m) \in (\mathbb{N}^*)^m \text{ s.t. } p_1 + \dots + p_m = n\}$

Interconnection laws: Sum, Product and Cascade



Equivalent transfer kernels C_n

Sum: $C_n(s_{1:n}) = A_n(s_{1:n}) + B_n(s_{1:n})$

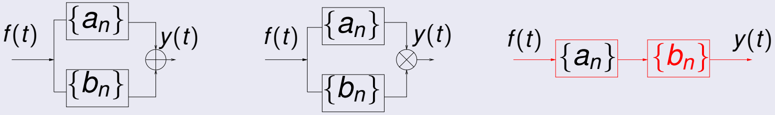
Product: $C_n(s_{1:n}) = \sum_{p=1}^{n-1} A_p(s_{1:p}) B_{n-p}(s_{p+1:n})$

Cascade: $C_n(s_{1:n}) = A_n(s_{1:n}) B_1(\widehat{s_{1:n}})$

(b_1: linear) with $\widehat{s_{1:n}} = s_1 + \ldots + s_n$

where $\mathbb{M}_n^m = \{(p_1, \ldots, p_m) \in (\mathbb{N}^*)^m \text{ s.t. } p_1 + \cdots + p_m = n\}$

Interconnection laws: Sum, Product and Cascade



Equivalent transfer kernels C_n

Sum: $C_n(s_{1:n}) = A_n(s_{1:n}) + B_n(s_{1:n})$

Product: $C_n(s_{1:n}) = \sum_{p=1}^{n-1} A_p(s_{1:p}) B_{n-p}(s_{p+1:n})$

Cascade: $C_n(s_{1:n}) = A_n(s_{1:n})\, B_1(\widehat{s_{1:n}})$

(b_1: linear) with $\widehat{s_{1:n}} = s_1 + \ldots + s_n$

Cascade: $C_n(s_{1:n}) = \sum_{m=1}^{n} \sum_{p \in \mathbb{M}_n^m} A_{p_1}(s_{1:p_1}) \ldots A_{p_m}(s_{p_1+\ldots+p_{m-1}+1:n})$

(general case) $.\, B_m\big(\widehat{s_{1:p_1}}, \ldots\ldots, \widehat{s_{p_1+\ldots+p_{m-1}+1:n}}\big)$

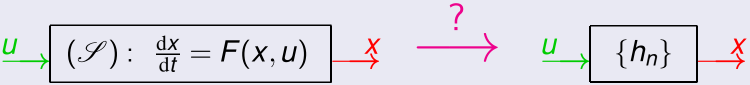
where $\mathbb{M}_n^m = \{(p_1, \ldots, p_m) \in (\mathbb{N}^*)^m \text{ s.t. } p_1 + \cdots + p_m = n\}$

Outline

1. Préambule
2. Séries de Volterra : généralités
3. Calcul des noyaux de Volterra d'un système différentiel
   - Lois d'interconnexion *(somme, produit, cascade)*
   - Calcul algébrique de noyaux de transfert
   - Réalisation numérique
4. Exercices et applications en audio-acoustique
5. Convergence
6. Extension en dimension infinie et application
7. Conclusion

How to derive the Volterra kernels of a system $\mathscr{S}$?

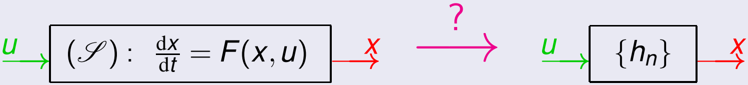
Goal: Find the Volterra kernels $\{h_n\}$ of $(\mathscr{S})$ where



Several methods are available... [Brockett,Isidori,Rugh,Boyd]

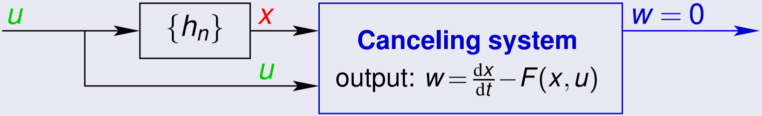
How to derive the Volterra kernels of a system $\mathscr{S}$?

Goal: Find the Volterra kernels $\{h_n\}$ of $(\mathscr{S})$ where



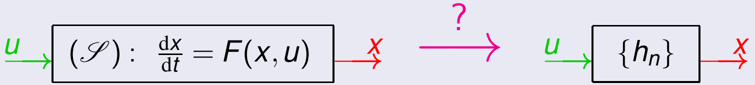
Several methods are available... [Brockett,Isidori,Rugh,Boyd]

<u>Here:</u> Introduce the "canceling system" of $\mathscr{S}$



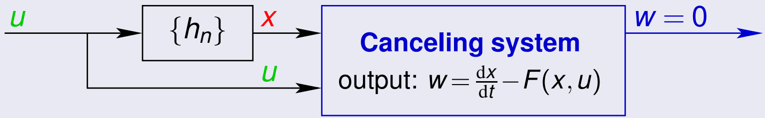
How to derive the Volterra kernels of a system $\mathscr{S}$?

Goal: Find the Volterra kernels $\{h_n\}$ of $(\mathscr{S})$ where



Several methods are available... [Brockett,Isidori,Rugh,Boyd]

Here: Introduce the "**canceling system**" of $\mathscr{S}$



Principle: this cascade defines the **null system** $u \rightarrow \{z_n = 0\} \rightarrow w$

Interconnection laws gives the equations satisfied by $\{h_n\}$.

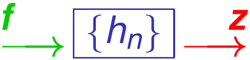
Example: nonlinear spring

$$\xrightarrow{f} \boxed{\{h_n\}} \xrightarrow{z}$$

Equation (at rest before t=0)

$$mz'' + az' + k_1 z + k_2 [z]^2 = f$$

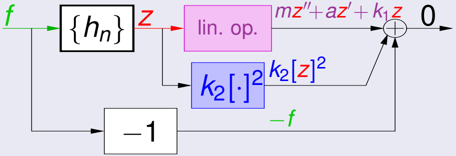
Example: nonlinear spring



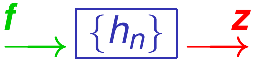
Equation (at rest before t=0)

$$mz'' + az' + k_1 z + k_2 [z]^2 = f$$

Canceling system



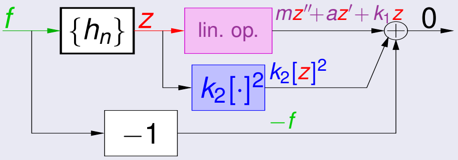
Example: nonlinear spring



Equation (at rest before t=0)

$$mz'' + az' + k_1 z + k_2 [z]^2 = f$$

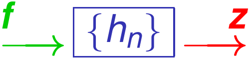
Canceling system



Elementary blocks → equivalent transfer kernels

$m\frac{\mathrm{d}^2}{\mathrm{d}t^2} + a\frac{\mathrm{d}}{\mathrm{d}t} + k_1$	$\rightarrow$	$Q_1(s) = ms^2 + as + k_1$, $Q_n = 0$ si $n \geq 2$
$k_2[\cdot]^2$	$\rightarrow$	interconnection "product" and $\times k_2$
-1	$\rightarrow$	$-\delta_{1,n} = -1$ if $n = 1$ and $-\delta_{1,n} = 0$ otherwise

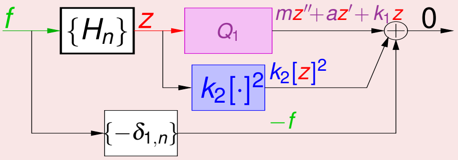
Example: nonlinear spring



Equation (at rest before t=0)

$$mz'' + az' + k_1 z + k_2 \left[z\right]^2 = f$$

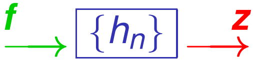
Canceling system



Elementary blocks → equivalent transfer kernels

$m\frac{d^2}{dt^2} + a\frac{d}{dt} + k_1$	$\rightarrow$	$Q_1(s) = ms^2 + as + k_1$, $Q_n = 0$ si $n \geq 2$
$k_2[\cdot]^2$	$\rightarrow$	interconnection “product” and $\times k_2$
-1	$\rightarrow$	$-\delta_{1,n} = -1$ if $n = 1$ and $-\delta_{1,n} = 0$ otherwise

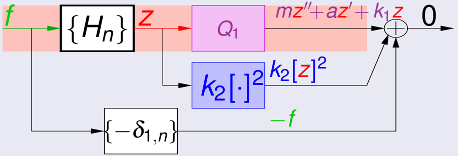
Example: nonlinear spring



Equation (at rest before t=0)

$$mz'' + az' + k_1 z + k_2 [z]^2 = f$$

Canceling system



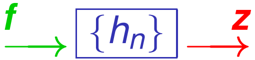
Kernel of order *n* of the *canceling system*

$$H_n(s_{1:n})Q_1(\widehat{s_{1:n}})$$

$$+ \quad k_2 \sum_{p=1}^{n-1} H_p(s_{1:p})H_{n-p}(s_{p+1:n})$$

$$+ \quad -\delta_{1,n} \qquad = 0$$

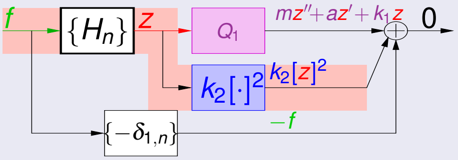
Example: nonlinear spring



Equation (at rest before t=0)

$$mz'' + az' + k_1 z + k_2 [z]^2 = f$$

Canceling system



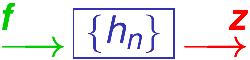
Kernel of order *n* of the *canceling system*

$$H_n(s_{1:n})Q_1(\widehat{s_{1:n}})$$

$$+ \quad k_2 \sum_{p=1}^{n-1} H_p(s_{1:p})H_{n-p}(s_{p+1:n})$$

$$+ \quad -\delta_{1,n} \qquad = 0$$

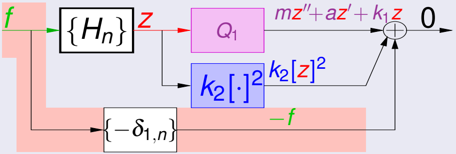
Example: nonlinear spring



Equation (at rest before t=0)

$$mz'' + az' + k_1 z + k_2 [z]^2 = f$$

Canceling system



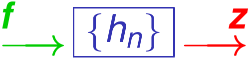
Kernel of order *n* of the *canceling system*

$$H_n(s_{1:n})Q_1(\widehat{s_{1:n}})$$

$$k_2 \sum_{p=1}^{n-1} H_p(s_{1:p})H_{n-p}(s_{p+1:n})$$

$$-\delta_{1,n}$$

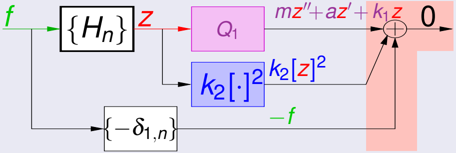
Example: nonlinear spring



Equation (at rest before t=0)

$$mz'' + az' + k_1 z + k_2 [z]^2 = f$$

Canceling system



Kernel of order *n* of the *canceling system*

$$\begin{aligned} & H_n(s_{1:n})Q_1(\widehat{s_{1:n}}) \\ + \quad & k_2 \sum_{p=1}^{n-1} H_p(s_{1:p})H_{n-p}(s_{p+1:n}) \\ + \quad & -\delta_{1,n} \qquad = 0 \end{aligned}$$

$\longrightarrow$ linear eq. w.r.t. H_n.

Kernels $\{H_n\}$ of the system $\xrightarrow{f} \boxed{\{h_n\}} \xrightarrow{z}$

General solution: recursive algebraic equation ($n \geq 1$)

$$H_n(s_{1:n}) = \frac{\delta_{1,n} - k_2 \overbrace{\sum_{p=1}^{n-1} H_p(s_{1:p}) H_{n-p}(s_{p+1:n})}^{\text{orders } < n}}{Q_1(\widehat{s_{1:n}})}$$

where $Q_1(s) = ms^2 + as + k_1$

Kernels $\{H_n\}$ of the system $\xrightarrow{f} \boxed{\{h_n\}} \xrightarrow{z}$

General solution: recursive algebraic equation ($n \geq 1$)

$$H_n(s_{1:n}) = \frac{\delta_{1,n} - k_2 \overbrace{\sum_{p=1}^{n-1} H_p(s_{1:p}) H_{n-p}(s_{p+1:n})}^{\text{orders } < n}}{Q_1(\widehat{s_{1:n}})}$$

where $Q_1(s) = ms^2 + a\,s + k_1$

First transfer kernels ($n = 1, 2, 3$,etc)

$H_1(s_1) = 1/Q_1(s_1)$, (second order AR filter)

$H_2(s_{1:2}) = -k_2\, H_1(s_1)\, H_1(s_2)\, H_1(\widehat{s_{1:2}})$,

$H_3(s_{1:3}) = -k_2 \left[H_2(s_{1:2})\, H_1(s_3) + H_1(s_1)\, H_2(s_{2:3})\right] H_1(\widehat{s_{1:3}})$,

etc.

Outline

1 Préambule

2 Séries de Volterra : généralités

3 Calcul des noyaux de Volterra d'un système différentiel
- Lois d'interconnexion *(somme, produit, cascade)*
- Calcul algébrique de noyaux de transfert
- Réalisation numérique

4 Exercices et applications en audio-acoustique

5 Convergence

6 Extension en dimension infinie et application

7 Conclusion

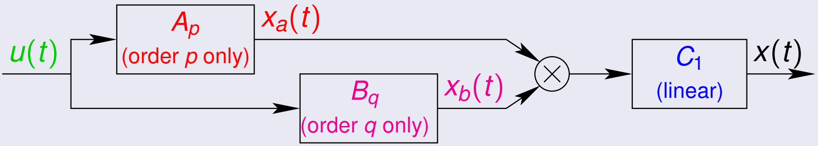
How to simulate the system using these kernels?

Several methods are available... (realization theory in [Rugh])

How to simulate the system using these kernels?

Several methods are available... (realization theory in [Rugh])

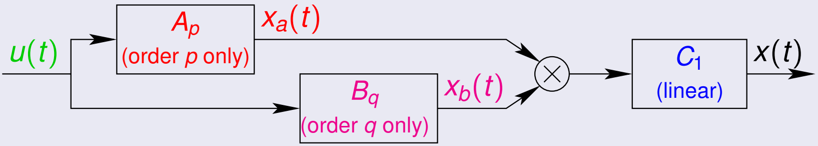
Consider the following **“elementary system”**



How to simulate the system using these kernels?

Several methods are available... (realization theory in [Rugh])

Consider the following "elementary system"



Equivalent system (using "product" & "cascade")

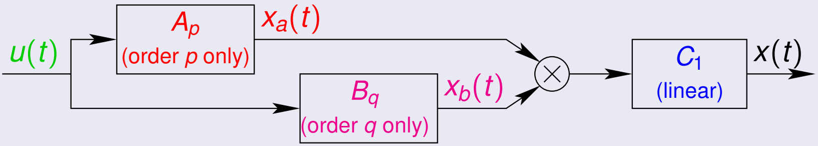
The Volterra transfer kernels of this system are all zero except

$$H_{p+q}(s_{1:p+q}) = A_p(s_{1:p})\, B_q(s_{p+1:p+q})\, C_1(\widehat{s_{1:p+q}})$$

How to simulate the system using these kernels?

Several methods are available... (realization theory in [Rugh])

Consider the following **"elementary system"**



Equivalent system (using "product" & "cascade")

The Volterra transfer kernels of this system are all zero except

$$H_{p+q}(s_{1:p+q}) = A_p(s_{1:p})\, B_q(s_{p+1:p+q})\, C_1(\widehat{s_{1:p+q}})$$

Method:

Recursively build a realization as a **sum of such systems**.

Example: nonlinear spring

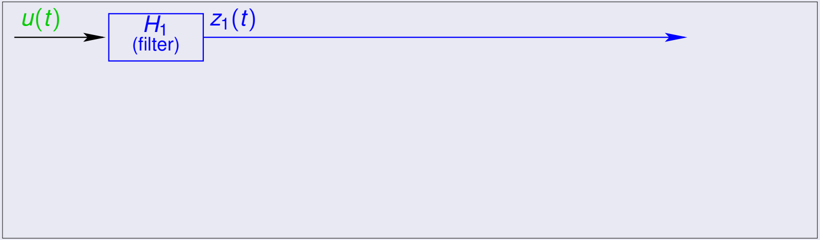
Transfer kernels

$H_1(s_1) = 1/Q_1(s_1)$, (AR filter)

Example: nonlinear spring

Transfer kernels

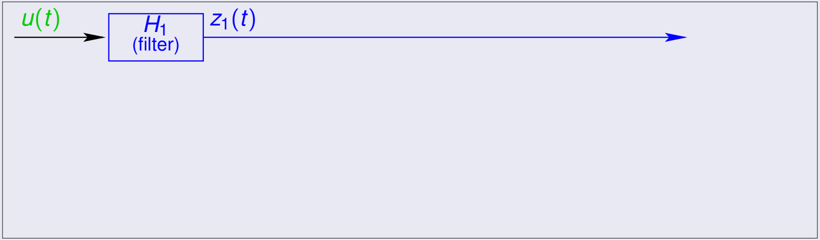
$H_1(s_1) = 1/Q_1(s_1)$, (AR filter)



Example: nonlinear spring

Transfer kernels

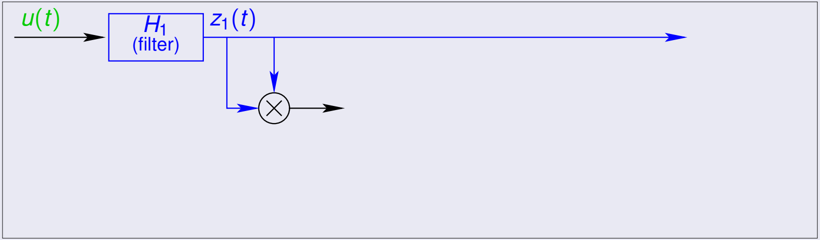
$$H_2(s_{1:2}) = -k_2\, H_1(s_1)\, H_1(s_2)\, H_1(\widehat{s_{1:2}})$$



Example: nonlinear spring

Transfer kernels

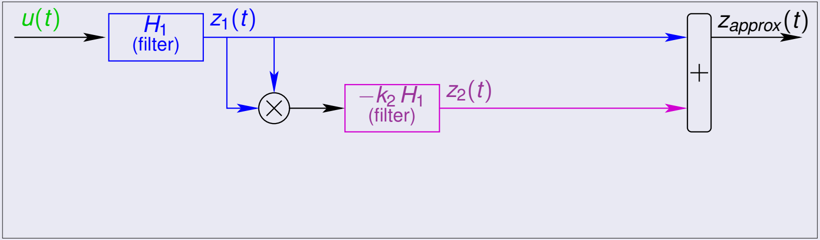
$$H_2(s_{1:2}) = -k_2\, H_1(s_1)\, H_1(s_2)\, H_1(\widehat{s_{1:2}})$$



Example: nonlinear spring

Transfer kernels

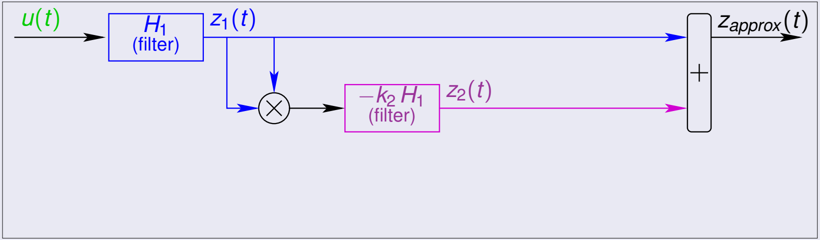
$$H_2(s_{1:2}) = -k_2\, H_1(s_1)\, H_1(s_2)\, H_1(\widehat{s_{1:2}})$$



Example: nonlinear spring

Transfer kernels

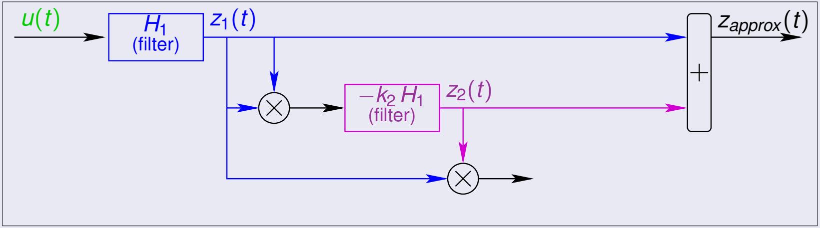
$$H_3(s_{1:3}) = -k_2\left[H_2(s_{1:2})\,H_1(s_3) + H_1(s_1)\,H_2(s_{2:3})\right]\,H_1(\widehat{s_{1:3}})$$



Example: nonlinear spring

Transfer kernels

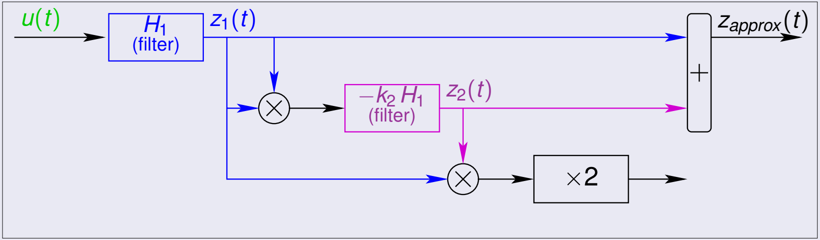
$$H_3(s_{1:3}) = -k_2 \left[H_2(s_{1:2})\, H_1(s_3) + H_1(s_1)\, H_2(s_{2:3}) \right] H_1(\widehat{s_{1:3}})$$



Example: nonlinear spring

Transfer kernels

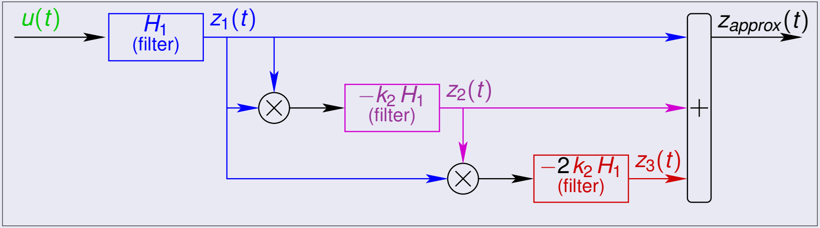
$$H_3(s_{1:3}) = -k_2 \left[H_2(s_{1:2})\, H_1(s_3) + H_1(s_1)\, H_2(s_{2:3}) \right] H_1(\widehat{s_{1:3}})$$



Example: nonlinear spring

Transfer kernels

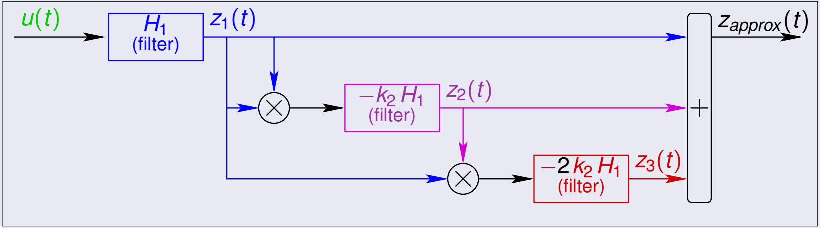
$$H_3(s_{1:3}) = -k_2 \left[H_2(s_{1:2})\, H_1(s_3) + H_1(s_1)\, H_2(s_{2:3}) \right] H_1(\widehat{s_{1:3}})$$



Example: nonlinear spring

Transfer kernels

$$H_3(s_{1:3}) = -k_2 \left[H_2(s_{1:2})\, H_1(s_3) + H_1(s_1)\, H_2(s_{2:3})\right] H_1(\widehat{s_{1:3}})$$



RESULT: The system is composed of **sums**, **products** and

linear systems (filters) ($\longrightarrow$ standard digital versions!)

Rk: if linear systems are stable, the system is stable!

Aliasing rejection in simulations

Product of signals

signal	$a(t)$	$b(t)$	$c(t) = a(t)\,b(t)$
frequency range	$(-f_a, f_a)$	$(-f_b, f_b)$	$(-f_a - f_b, f_a + f_b)$

Aliasing rejection in simulations

Product of signals

signal	$a(t)$	$b(t)$	$c(t) = a(t)\, b(t)$
frequency range	$(-f_a, f_a)$	$(-f_b, f_b)$	$(-f_a - f_b, f_a + f_b)$

Aliasing rejection

(Global sol.) Oversample the input/Downsample the output: factor N for a VS truncated at order N.

(Local sol.) Idem with a factor 2 for each product of two signals.

In summary...

Derivation of the transfer kernels

1. Use the cancelling system and interconnection laws...
2. ... to transform the weakly nonlinear problem into a infinite sequence of solvable linear equations

In summary...

Derivation of the transfer kernels

1. Use the cancelling system and interconnection laws...
2. ... to transform the weakly nonlinear problem into a infinite sequence of solvable linear equations

Simulation in practice

1. Truncate the series to catch the first distortions
2. Decompose the kernels into sums of elementary systems
3. Build the corresponding structure composed of **linear filters**, **sums** and **products** of signals
4. Implement digital versions of the filters
5. Add oversampler/downsampler (aliasing rejection)

In summary...

Derivation of the transfer kernels

1. Use the cancelling system and interconnection laws...
2. ... to transform the weakly nonlinear problem into a infinite sequence of solvable linear equations

Simulation in practice

1. Truncate the series to catch the first distortions
2. Decompose the kernels into sums of elementary systems
3. Build the corresponding structure composed of **linear filters**, **sums** and **products** of signals
4. Implement digital versions of the filters
5. Add oversampler/downsampler (aliasing rejection)

Possible generalization to Partial Differential Equations

Same principle (cf. Applications)

Plan

1 Préambule

2 Séries de Volterra : généralités

3 Calcul des noyaux de Volterra d'un système différentiel

4 Exercices et applications en audio-acoustique

5 Convergence

6 Extension en dimension infinie et application

7 Conclusion

Plan

1. Préambule
2. Séries de Volterra : généralités
3. Calcul des noyaux de Volterra d'un système différentiel
4. Exercices et applications en audio-acoustique
5. Convergence
6. Extension en dimension infinie et application
7. Conclusion

Plan

1. Préambule
2. Séries de Volterra : généralités
3. Calcul des noyaux de Volterra d'un système différentiel
4. Exercices et applications en audio-acoustique
5. Convergence
6. Extension en dimension infinie et application
7. Conclusion

Plan

1. Préambule
2. Séries de Volterra : généralités
3. Calcul des noyaux de Volterra d'un système différentiel
4. Exercices et applications en audio-acoustique
5. Convergence
6. Extension en dimension infinie et application
7. Conclusion