`vle.discrete-time.generic` : a VLE package that provides simple atomics models.

RECORD Team
(Ronan Trépos, Eric Casellas)

September 7, 2017

Contents

1 Introduction

The package `vle.discrete-time.generic` provides simple atomic models for either:

- computing simple functions such as weighted sums.

The global description of output trajectories of these models is based on a function of intput trajectories.

2 Constant

Categories: Generative models

Parameters:

- c : the constant value
- Δ : the time step of the output

Input trajectory:

None

Output trajectory:

Δ	$2*\Delta$	$\dots$	$z*\Delta$
c	c	$\dots$	c

Perturbations: The value of c can be modified during simulation by using the input port corresponding to the output port.

Use cases:

- The Constant model can be used to develop a complex model in an incremental way. Consider for example the development a model DiscreteTime DT that depends on a external variable, that is not yet available. The constant model C provides, during the implementation time period, the inputs necessary to the dynamic of DT.

$$C \rightarrow DT$$

Miscellaneous: Based on vle::discrete_time::DiscreteTimeDyn.

3 Identity

Categories: Identity models

Parameters:

- Δ : the time step of the input

Input trajectory:

Δ	$2*\Delta$	$\dots$	$z*\Delta$
v_1	v_2	$\dots$	v_z

Output trajectory:

Δ	$2*\Delta$	$\dots$	$z*\Delta$
v_1	v_2	$\dots$	v_z

Use cases:

- The Identity model can be used to observe state variables of models built by an executive. For example, consider an executive model that instantiates, at time t_1, a model M and replaces, at t_2, M by another model M' providing the same outputs ; outputs of both M and M' are coupled to inputs of an Identity model Id.

$$\begin{cases} \text{between } t_1 \text{ and } t_2 & M \rightarrow Id \\ \text{after } t_2 & M' \rightarrow Id \end{cases}$$

  The Identity model allows the observation of the state variable of both M and M', seen as the same state variable.

Miscellaneous: Based on vle::discrete_time::DiscreteTimeDyn.

4 Sum

Categories: Aggregate Variable models

Parameters:

- Δ : the time step of the output

Input trajectory:

Δ	$2*\Delta$	$\dots$	$z*\Delta$
v_1^1	v_2^1	$\dots$	v_z^1
v_1^2	v_2^2	$\dots$	v_z^2
$\dots$	$\dots$	$\dots$	$\dots$
v_1^m	v_2^m	$\dots$	v_z^m

Output trajectory:

Δ	$2*\Delta$	$\dots$	$z*\Delta$
$\sum_{j=1}^m v_1^j$	$\sum_{j=1}^m v_2^j$	$\dots$	$\sum_{j=1}^m v_z^j$

Miscellaneous: Based on vle::discrete_time::DiscreteTimeDyn.

5 Average

Categories: Aggregate Variable models

Parameters:

- Δ : the time step of the output

Input trajectory:

Δ	$2*\Delta$	$\dots$	$z*\Delta$
v_1^1	v_2^1	$\dots$	v_z^1
v_1^2	v_2^2	$\dots$	v_z^2
$\dots$	$\dots$	$\dots$	$\dots$
v_1^m	v_2^m	$\dots$	v_z^m

Output trajectory:

Δ	$2*\Delta$	$\dots$	$z*\Delta$
$\frac{1}{m}\sum_{j=1}^m v_1^j$	$\frac{1}{m}\sum_{j=1}^m v_2^j$	$\dots$	$\frac{1}{m}\sum_{j=1}^m v_z^j$

Miscellaneous: Based on vle::discrete_time::DiscreteTimeDyn.

6 Product

Categories: Aggregate Variable models

Parameters:

- Δ : the time step of the output

Input trajectory:

Δ	$2*\Delta$	$\dots$	$z*\Delta$
v_1^1	v_2^1	$\dots$	v_z^1
v_1^2	v_2^2	$\dots$	v_z^2
$\dots$	$\dots$	$\dots$	$\dots$
v_1^m	v_2^m	$\dots$	v_z^m

Output trajectory:

Δ	$2 * \Delta$	$\dots$	$z * \Delta$
$\prod_{j=1}^{m} v_1^j$	$\prod_{j=1}^{m} v_2^j$	$\dots$	$\prod_{j=1}^{m} v_z^j$

Miscellaneous: Based on vle::discrete_time::DiscreteTimeDyn.

7 WeightedSum

Categories: Aggregate Variable models

Parameters:

- Δ : the time step of the output
- w_i : weight for all input variables

Input trajectory:

Δ	$2 * \Delta$	$\dots$	$z * \Delta$
v_1^1	v_2^1	$\dots$	v_z^1
v_1^2	v_2^2	$\dots$	v_z^2
$\dots$	$\dots$	$\dots$	$\dots$
v_1^m	v_2^m	$\dots$	v_z^m

Output trajectory:

Δ	$2 * \Delta$	$\dots$	$z * \Delta$
$\sum_{j=1}^{m} w_j * v_1^j$	$\sum_{j=1}^{m} w_j * v_2^j$	$\dots$	$\sum_{j=1}^{m} w_j * v_z^j$

Miscellaneous: Based on vle::discrete_time::DiscreteTimeDyn..

8 WeightedProduct

Categories: Aggregate Variable models

Parameters:

- Δ : the time step of the output
- w_i : weight for all input variables

Input trajectory:

Δ	$2 * \Delta$	$\dots$	$z * \Delta$
v_1^1	v_2^1	$\dots$	v_z^1
v_1^2	v_2^2	$\dots$	v_z^2
$\dots$	$\dots$	$\dots$	$\dots$
v_1^m	v_2^m	$\dots$	v_z^m

Output trajectory:

Δ	$2 * \Delta$	$\dots$	$z * \Delta$
$\prod_{j=1}^{m} w_j * v_1^i$	$\prod_{j=1}^{m} w_j * v_2^i$	$\dots$	$\prod_{j=1}^{m} w_j * v_z^i$

Miscellaneous: Based on vle::discrete_time::DiscreteTimeDyn.

9 MovingAverage

Categories: Aggregate Times models

Parameters:

- Δ : the time step of the output
- n : number of time-step to take into account in the average
- m : number of time-step to wait before taking into account the current input value (default 0).

Input trajectory:

Δ	$2*\Delta$	$3*\Delta$	$4*\Delta$	...	$z*\Delta$
v_1	v_2	v_3	v_4	...	v_z

Output trajectory: (for $m = 1$, $n = 3$)

Δ	$2*\Delta$	$3*\Delta$	$4*\Delta$	...	$z*\Delta$
0	0	$\frac{v_1+v_2}{n}$	$\frac{v_1+v_2+v_3}{n}$	...	$\frac{1}{n}\sum_{i=z-m-n}^{z-m} v_i$

Use cases:

- The MovingAverage model can be used to provide the classical moving average (with $m = 0$) of the sub sequence of v_i from $t - n$ to t where t is the current time.
- It can also simulates a delay (with $n = 1$). The output value is then v_{t-m} where t is the current time.

Miscellaneous: Based on vle::discrete_time::DiscreteTimeDyn.

10 MovingSum

Categories: Aggregate Times models

Parameters:

- Δ : the time step of the output
- n : number of time-step to take into account in the sum

Input trajectory:

Δ	$2*\Delta$	...	$z*\Delta$
v_1^1	v_2^1	...	v_z^1
v_1^2	v_2^2	...	v_z^2
...	...	...	...
v_1^m	v_2^m	...	v_z^m

Output trajectory: (for $n = 3$)

Δ	$2 * \Delta$	$3 * \Delta$	$4 * \Delta$	$\dots$	$z * \Delta$
v_1^1	$v_1^1 + v_2^1$	$v_1^1 + v_2^1 + v_3^1$	$v_2^1 + v_3^1 + v_4^1$	$\dots$	$\sum_{i=z-n}^{z} v_i^1$
v_1^2	$v_1^2 + v_2^2$	$v_1^2 + v_2^2 + v_3^2$	$v_2^2 + v_3^2 + v_4^2$	$\dots$	$\sum_{i=z-n}^{z} v_i^2$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
v_1^m	$v_1^m + v_2^m$	$v_1^m + v_2^m + v_3^m$	$v_2^m + v_3^m + v_4^m$	$\dots$	$\sum_{i=z-n}^{z} v_i^m$

Use cases:

- The MovingSum model can be used to provide the classical moving sum of the sub sequence of v_i from $t - n$ to t where t is the current time.

Miscellaneous: Based on vle::discrete_time::DiscreteTimeDyn.. It can accept multiple inputs and outputs, but a single n parameter.

Index