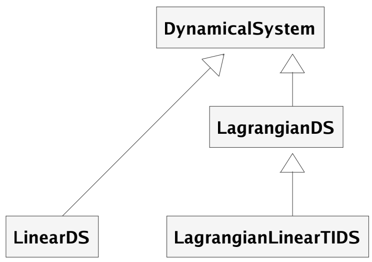
Various Dynamical Systems formulations in Siconos.

F. Pérignon

For Kernel version 1.1.4
March 22, 2006

1 Class Diagram

There are four possible formulation for dynamical systems in Siconos, two for first order systems and two for second order Lagrangian systems. The main class is DynamicalSystem, all other derived from this one, as shown in the following diagram:



2 General non linear first order dynamical systems $\rightarrow$ class *DynamicalSystem*

This is the top class for dynamical systems. All other systems classes derived from this one.

A general dynamical systems is described by the following set of n equations, completed with initial conditions:

$$\dot{x} = f(x,t) + T(x)u(x,\dot{x},t) + r \tag{1}$$
$$x(t_0) = x_0 \tag{2}$$

- x: state of the system - Vector of size n.
- $f(x,t)$: vector field - Vector of size n.
- $u(x,\dot{x},t)$: control term - Vector of size $uSize$.
- $T(x)$: $n \times uSize$ matrix, related to control term.
- r: input due to non-smooth behavior - Vector of size n.

The Jacobian matrix, $\nabla_x f(x,t)$, of f according to x, $n \times n$ square matrix, is also a member of the class.

Initial conditions are given by the member x_0, vector of size n. This corresponds to x value when simulation is starting, ie after a call to strategy->initialize().

There are plug-in functions in this class for f (vectorField), $jacobianX$, u and T. All of them can handle a vector of user-defined parameters.

3 First order linear dynamical systems → class *LinearDS*

Derived from DynamicalSystem, described by the set of n equations and initial conditions:

$$\begin{aligned} \dot{x} &= A(t)x(t) + Tu(t) + b(t) + r && (3) \\ x(t_0) &= x_0 && (4) \end{aligned}$$

With:

- $A(t)$: $n \times n$ matrix, state independent but possibly time-dependent.
- $b(t)$: Vector of size n, possibly time-dependent.

Other variables are those of DynamicalSystem class.
A and B have corresponding plug-in functions.

Warning: time dependence for A and b is not available at the time in the simulation part for this kind of dynamical systems.

Links with vectorField and its Jacobian are:

$$\begin{aligned} f(x,t) &= A(t)x(t) + b(t) && (5) \\ jacobianX &= \nabla_x f(x,t) = A(t) && (6) \end{aligned}$$

4 Second order non linear Lagrangian dynamical systems → class *LagrangianDS*

Lagrangian second order non linear systems are described by the following set of$nDof$ equations + initial conditions:

$$\begin{aligned} M(q)\ddot{q} + NNL(\dot{q},q) + F_{Int}(\dot{q},q,t) &= F_{Ext}(t) + p && (7) \\ q(t_0) &= q0 && (8) \\ \dot{q}(t_0) &= velocity0 && (9) \end{aligned}$$

With:

- $M(q)$: $nDof \times nDof$ matrix of inertia.
- q: state of the system - Vector of size $nDof$.
- $\dot{q}$ or $velocity$: derivative of the state according to time - Vector of size $nDof$.
- $NNL(\dot{q},q)$: non linear terms, time-independent - Vector of size $nDof$.
- $F_{Int}(\dot{q},q,t)$: time-dependent linear terms - Vector of size $nDof$.
- $F_{Ext}(t)$: external forces, time-dependent BUT do not depend on state - Vector of size $nDof$.
- p: input due to non-smooth behavior - Vector of size $nDof$.

The following Jacobian are also member of this class:

- jacobianQFInt = $\nabla_q F_{Int}(t,q,\dot{q})$ - $nDof \times nDof$ matrix.
- jacobianVelocityFInt = $\nabla_{\dot{q}} F_{Int}(t,q,\dot{q})$ - $nDof \times nDof$ matrix.
- jacobianQNNL = $\nabla_q NNL(q,\dot{q})$ - $nDof \times nDof$ matrix.
- jacobianVelocityNNL = $\nabla_{\dot{q}} NNL(q,\dot{q})$ - $nDof \times nDof$ matrix.

There are plug-in functions in this class for F_{int}, F_{Ext}, M, NNL and the four Jacobian matrices. All of them can handle a vector of user-defined parameters.

Links with first order dynamical system are:

$$
\begin{aligned}
n &= 2nDof && (10)\\
x &= \begin{bmatrix} q \\ \dot{q} \end{bmatrix} && (11)\\
f(x,t) &= \begin{bmatrix} \dot{q} \\ M^{-1}(F_{Ext} - F_{Int} - NNL) \end{bmatrix} && (12)\\
& && (13)\\
\nabla_x f(x,t) &= \begin{bmatrix} 0_{nDof\times nDof} & I_{nDof\times nDof} \\ \nabla_q(M^{-1})(F_{Ext} - F_{Int} - NNL) - M^{-1}\nabla_q(F_{Int} + NNL) & -M^{-1}\nabla_{\dot{q}}(F_{Int} + NNL) \end{bmatrix} && (14)\\
r &= \begin{bmatrix} 0_{nDof} \\ p \end{bmatrix} && (15)\\
u(x,\dot{x},t) &= u_L(\dot{q},q,t) \text{ (not yet implemented)} && (16)\\
T(x) &= \begin{bmatrix} 0_{nDof} \\ T_L(q) \end{bmatrix} \text{ (not yet implemented)} && (17)\\
& && (18)
\end{aligned}
$$

With 0_n a vector of zero of size n, $0_{n\times m}$ a $n \times m$ zero matrix and $I_{n\times n}$, identity $n \times n$ matrix.

Warning: control terms (Tu) are not fully implemented in Lagrangian systems. This will be part of future version.

5 Second order linear and time-invariant Lagrangian dynamical systems → class *LagrangianLinearTIDS*

$$M\ddot{q} + C\dot{q} + Kq = F_{Ext}(t) + p \tag{19}$$

With:

- C: constant viscosity $nDof \times nDof$ matrix
- K: constant rigidity $nDof \times nDof$ matrix

And:

$$
\begin{aligned}
F_{Int} &= C\dot{q} + Kq && (20)\\
NNL &= 0_{nDof} && (21)
\end{aligned}
$$