RINO: some implementation notes

Sylvie Putot

No Institute Given

1 ODEs

We are given a system of ODEs

$$\dot{z}(t) = f(z, u, t)$$

with $z(0) = z^0$. Here, for efficiency reasons, we do not systematically rewrite u as a dimension of the system.

- sysdim is the dimension of the state space of the system of ODE
- sysdim_params is the dimension of the (possibly uncertain) parameters u that are not dimensions of the Jacobian
- inputsdim is the dimension of the uncertain parameters and inputs u that appear in the Jacobian
- when these parameters can be piecewise constant and not constant, fullinputsdim is the full dimension of these uncertain parameters and inputs
- jacdim = sysdim + fullinputsdim

`TM_Jac::build` and TM_Jac::eval_Jac build and evaluate the Taylor model for the Jacobian $J_{ij}(t, \boldsymbol{z}_0) = \frac{\partial z_i}{\partial z_{0,j}}(t, \boldsymbol{z}_0)$ and $J_{ij}(t, \boldsymbol{z}_0) = \frac{\partial z_i}{\partial u_{0,j}}(t, \boldsymbol{z}_0)$, as defined in Equation (17) of [HSCC2017]:

- in class TM_Jac, J corresponds to J_j in (17), and J_rough to R_{j+1}.
- in TM_Jac::eval_Jac, Jaci corresponds to $Jac_z f^{[i]}$ in (17)

The code is slightly more difficult to understand when jacdim is not equal to sysdim. Let us consider an example with dimension 2 and 2 parameters, the sensitivity matrix J we want to compute relies on products

$$\mathrm{Jac}_z f^{[i]}([\boldsymbol{z}_j]).[\boldsymbol{J}_j]$$

of matrices of dimensions dimension (jacdim $\times$ jacdim) with, for example for $i = 1$,

$$\mathrm{Jac}_z f^{[1]}([\boldsymbol{z}_j]).[\boldsymbol{J}_j] = \begin{pmatrix} \frac{\partial f_0}{\partial z_0} & \frac{\partial f_0}{\partial z_1} & \frac{\partial f_0}{\partial u_0} & \frac{\partial f_0}{\partial u_1} \\ \frac{\partial f_1}{\partial z_0} & \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial u_0} & \frac{\partial f_1}{\partial u_1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} . \begin{pmatrix} \frac{\partial z_0}{\partial z_0^0} & \frac{\partial z_0}{\partial z_1^0} & \frac{\partial z_0}{\partial u_0} & \frac{\partial z_0}{\partial u_1} \\ \frac{\partial z_1}{\partial z_0^0} & \frac{\partial z_1}{\partial z_1^0} & \frac{\partial z_1}{\partial u_0} & \frac{\partial z_1}{\partial u_1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Thus, only part of the matrices are relevant. For efficiency, we store only the submatrices of dimension (sysdim $\times$ jacdim), and adapt the matrix operations such as matrix multiplication (multJacfzJaczz0) in that respect. We thus write

$$\mathrm{Jac}_z f.[\boldsymbol{J}_j] = \left(\mathrm{Jac}_z f.\mathrm{Jac}_{z^0} z \ \mathrm{Jac}_z f \mathrm{Jac}_u z + Jac_u^f\right)$$

This actually corresponds to Equations (8) and (9) (where beta is z0) of [HSCC2019].

2 Examples

Piecewise constant inputs Examples 19 and 21 are the following system:

$$\dot{x}(t) = f(u,t) = 2 + 2u + (1-2u)t$$

with $u \in [0,1]$ and $x(0) = 0$. For now, time is embedded as a component of the system.

This example is a good example to test piecewise constant time-varying inputs. Let us take time horizon of 2 and stepsize of 1. If u is constant equal to u_0 on the whole interval, then

$$x(t) = \int_0^t 2 + 2u_0 + (1-2u_0)sds = [(2+2u_0)s + (1-2u_0)s^2/2]_0^t$$

and in particular $x(2) = 6$. If u is only piecewise constant on each time setp of size 1, then

$$x(2) = \int_0^1 2 + 2u_1 + (1-2u_1)sds + \int_1^2 2 + 2u_2 + (1-2u_2)sds$$

and if both u_1 and u_2 are in $[0,1]$ then $x(2) \in [5,7]$.

On thix example, the maximal outer and inner approximations are exact. I thus added a non linearity to get non exact approximations in Example 22 which encodes

$$\dot{x}(t) = f(u,t) = 2 + 2(u^2+u) + (1-2(u^2+u))t$$

with $u \in [0,1]$ and $x(0) = 0$.