Networked Systems and Delay Differential Equations

Lorenz Sahlmann

February 2016

1 Project Idea

Cyber Physical Systems incorporate the connection between the physical world and computing devices. This connection is often given by a computer network, which needs hence to be considered in the system model.

2 Stability of Networked Control Systems

I dived into the overall subject by thoroughly reading and working through the article

> Wei Zhang, Michael S. Branicky, and Stephen M. Phillips. *Stability of Networked Control Systems*, Control Systems Magazine, IEEE 21.1 (2001): 84-99.

in order to get to know the concepts and nomenclature of the domain, and the mathematical concepts involved by reconstructing the performed calculations.

A **Networked Control Systems (NCS)** is a feedback control system with sensing and control data transmitted on a network. Sensors sample the state of a *plant* periodically and send their output as packets to a event-driven *controller*, which calculates a control signal as soon as the sensor data arrives. This is then transmitted to event-driven actuators in the plant, which perform action immediately on reception of the command.

The plant is considered to be continuous in time by its physical nature, whereas the controller is discrete in time.

This scenario has some possible issues, such as network induced **delay** and **loss** of network packets. For that reason, the plant output and the controller input are not delivered at the same time and the controller might not have received all the plant updates when it has to perform its control calculations. This makes NCSs different to conventional sampled-data systems.

2.1 Modelling of NCSs with Network-Induced Delay

The plant (physical component) is modeled as time continuous and the evolution of its state $x \in \mathbb{R}^n$ by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}$$

which depends additionally on a control signal $u \in \mathbb{R}^m$ provided by the discrete controller

$$u(kh) = -Kx(kh), \quad k \in N_0$$

or

$$u(kh) = -Ky(kh), \quad k \in N_0$$

if not the full state is known to the controller, but only some plant output (sensor data) $y \in \mathbb{R}^p$. The matrices A, B, K are chosen with suitable dimensions.

The network induces delays in the loop, namely τ_{sc} between sensor and controller, as well as τ_{ca} between controller and actuator.

In case of a time-invariant controller, the partial delays can be combined together into a single $\tau_k = \tau_{sc,k} + \tau_{ca,k} + t_{\text{calculation}}$ with the processing time of the controller.

Assuming that the delay τ_k of each sample k is less than the sampling period h and that each data sample $x(t)$ fits into a single packet, the system equations can be written as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ &\qquad t \in [kh + \tau_k, (k+1)h + \tau_{k+1}] \\ y(t) &= Cx(t) \\ u(t^+) &= -Kx(t - \tau_k) \\ &\qquad t^+ \in \{kh + \tau_k, k = 0, 1, 2, ...\} \end{aligned}$$

with the piecewise constant control signal $u(t^+)$ in the actuator.

This plant system can be solved using the standard *variation of constants* method in order to express the new state variables $u(kh)$ and $x((k+1)h)$ as function of their values at the previous sampling instant.

2.2 Hybrid Systems

A more general class of systems are called *hybrid*, which consist of a continuous dynamics part and discrete events. The NCS model above can be written as such what allows applying stability theory for hybrid systems to derive conditions for asymptotical stability depending on the sampling rate h and the network delay τ.

2.3 Compensation of Network Induced Delay

If the plant and the controller have synchronized clocks, the sensor-controller delay can be determined in the controller. Using an estimator to approximate the evolved full plant state at time of reception even if only the partial state measurements $y(t)$ are available, one can try to compensate the sensor-controller delay by an estimator-predictor scheme.

Having an estimation of the full plant state $\hat{x}(kh)$ for time kh, one awaits the reception of the plant output $y(kh)$ for this instant. Receiving this packet at time $kh + \tau_{sc,k}$ one can correct the former prediction $\hat{x}(kh)$ to a better estimation $\bar{x}(kh)$. Assuming that this estimation fulfills the equations describing the system, one forwards the estimation to $\bar{x}(kh + \tau_{sc,k})$ which is used to calculate the control command $u(kh + \tau_{sc,k})$. In order to prepare the next iteration, $\bar{x}(kh)$ is further forwarded to obtain a prediction of the plant state at time $(k+1)h$.

2.4 Modelling of Packet Loss

The potential loss of data packets on the network can be modeled as an **Asynchronous Dynamical System (ADS)**, which comprises continuous dynamics (described by differential/difference equations) and discrete dynamics (governed by finite automata). Assuming that the non-networked system is stable, that the network is lossy only between sensor and controller and that the packets contain $x(kh)$ to provide the full current state to the controller, a pair of difference equations

$$\begin{aligned} S_0 : \quad & \bar{x}(kh) = \bar{x}((k-1)h) \\ S_1 : \quad & \bar{x}(kh) = x(kh) \end{aligned}$$

is obtained.

This system can be interpreted as a switch that closes at a certain rate r, indicating if a message is lost (S_0) or delivered (S_1). In the case of S_0 the state in the controller $x(kh)$ is held at its previous value. For this system, Lyaponov theory gives conditions for exponential stability.

3 Rigorous Integration of Delay Differential Equations

A method for modelling Networked Control Systems are **Delay Differential Equations**. A rigorous integration scheme based on Taylor methods is presented in

> Robert Szczelina. *Rigorous Integration of Delay Differential Equations*, Faculty of Mathematics and Computer Science, Jagiellonian University. Krakow, 2014.

The following subsection gives a summary on this algorithm.

3.1 Summary

The goal is to calculate strict bounds to the solution (i.e. for its values and derivatives) of a DDE on a time interval $[-1, T]$.

3.1.1 Setting

We restrict to a scalar DDE

$$\dot{x} = f(x(t), x(t-\tau))$$

with a single constant delay $\tau = 1$ and a right-hand side $f : \mathbb{R}^2 \to \mathbb{R}$ of class C^∞. The delay differential equation is assumed to admit a unique solution $x : [0, \tau] \to \mathbb{R}$ when equipped with the initial condition $x_0 : [-\tau, 0] \to \mathbb{R}$.

3.1.2 Taylor Expansion

We recapitulate the following theorem from basic calculus. Let $x : D \subset \mathbb{R} \to \mathbb{R}$ be a $(n+1)$-times continuously differentiable function and $a \in D$. There is a $c \in (a, t)$ such that x can be developed in a (finite) series expansion plus a remainder term

$$x(t) = \sum_{k=0}^{n} x^{[k]}(a)(t-a)^k + x^{[n+1]}(c)(t-a)^{n+1}$$

using the notation

$$x^{[k]}(a) := \frac{x^{(k)}(a)}{k!}$$

3.1.3 Canonical (p,n)-representation

For a fixed $p \in \mathbb{N}_{>0}$, let C_p^{n+1} the set of functions $f : \mathbb{R} \to \mathbb{R}$ which are $(n+1)$-times continuously differentiable on each subinterval $I_i = [-ih, -(i-1)h]$ of $[-1, 0]$, where $h = \tau/p$ and $i = 1, \ldots, p$.

We can represent a function $x \in C_p^{n+1}$ piecewisely by a collection of (forward) Taylor coefficients up to order n at equally spaced points

$$\bar{x}^{0,[0]} := \{x(0)\}$$
$$\bar{x}^{i,[k]} := \{x^{[k]}(-ih)\}$$

and an interval for the (Taylor) remainder term

$$\bar{x}^{i,[n+1]} := \{x^{[n+1]}(-ih + \xi) : \xi \in [0, h]\}$$

on each subinterval $I_i = [-ih, -(i-1)h]$ where $k = 0, \ldots, n$.

It is sufficient to define these representations on $[-1, 0]$ since the equations for a current time interval $[s - \tau, s]$ can be rescaled onto $[-1, 0]$. A DDE is time invariant.

However, such a (p,n)-representation is not unique in the sense that there can be multiple elements of C_p^{n+1} which admit the same coefficients.

Given a set of coefficients, one has a **rigorous bound** (i. e. a super set interval) for the values of the corresponding functions

$$x(t) \in \sum_{k=0}^{n+1} \bar{x}^{i,[k]} \xi^k \qquad (1)$$

and for their derivatives

$$x^{[k]}(t) \in \sum_{l=k}^{n+1} \binom{l}{k} \xi^{l-k} \bar{x}^{i,[l]} \qquad (2)$$

at each point $t = -ih + \xi$ of the subinterval I_i where $\xi \in [0, h)$. Here, the *sums* are in terms of interval arithmetic.

3.1.4 Integration of a DDE

The idea of the integration method is to compute the Taylor coefficients of the solution $\bar{x}_h^{i,[k]}$ at the next sampling point $t = h$, starting from the (p,n)-representation of the initial condition x_0. This procedure is then repeated on the obtained new coefficients.

The algorithm consists of two parts.

Shift part Since the time intervals $[-1, 0]$ and $[-1 + h, h]$ mostly overlap, most coefficients are actually shared, i. e. $\bar{x}_h^{i,[k]} = \bar{x}_0^{(i-1),[k]}$ for $i > 1$ and $\bar{x}_h^{1,[0]} = \bar{x}_0^{0,[0]}$.

Forwarding Part The remaining coefficients need to be computed.

1. Step Compute the coefficients $\bar{x}_h^{1,[k]}$ for $k = 1, \ldots, n$ using

$$\begin{aligned}\bar{x}_h^{1,[k]} &= \frac{1}{k!} x^{(k)}(0) \\ &= \frac{1}{k!} \frac{d^{k-1}}{dt^{k-1}} f(x_0(0-\tau), x(0)).\end{aligned}$$

Since the structure of f leads to derivatives of the form

$$x^{(k+1)}(t) =: F_{(k)} \Big(x(t-\tau), \ldots, x^{(k)}(t-\tau), \\ x(t), \ldots, x^{(k)}(t) \Big)$$

one can replace the second factor by

$$x^{[k]}(0) =: \frac{1}{k!} F_{(k-1)} \Big(0!\, x_0^{[0]}(-\tau), \ldots \\ \ldots, (k-1)!\, x_0^{[k-1]}(-\tau), 0!\, x^{[0]}(0), \ldots \\ \ldots, (k-1)!\, x^{[k-1]}(0) \Big)$$

what we write for simplicity as

$$=: F^{[k-1]} \Big(x_0^{[0]}(-\tau), \ldots, x_0^{[k-1]}(-\tau), \\ x^{[0]}(0), \ldots, x^{[k-1]}(0) \Big)$$

These expressions can be efficiently computed using **automatic differentiation** (with interval arithmetic), since all coefficients needed are known at the time of computation.

2. Step Compute an enclosure for the remainder coefficient, i. e.

$$\bar{x}_h^{1,[n+1]} = \Big\{ x^{[n+1]}(\xi) : \xi \in [0, h] \Big\} \qquad (3)$$

having by the mean value theorem, that there is a $\zeta \in (0, \xi)$ such that

$$x^{[n+1]}(\xi) = \\ \tfrac{1}{n+1} F^{[n]} \Big(x(-\tau), \ldots, x^{[n]}(-\tau), x(0), \ldots, x^{[n]}(0) \Big) \\ + F^{[n+1]} \Big(x(-\tau+\zeta), \ldots, x^{[n+1]}(-\tau+\zeta), \\ x(\zeta), \ldots, x^{[n+1]}(\zeta) \Big) \cdot \xi.$$

For the evaluation of $x^{[n+1]}([0, h])$, one needs to compute a rough enclosure of $x^{[n+1]}([0, h])$, since the equation depends on itself.

All other parameters have either been calculated in step 1 or, in the case of $x^{[k]}([0, h])$ and $x^{[k]}([-\tau, h-\tau])$, can be computed with the formulae (1) and (2) given above.

3. Step Finally, compute the rigorous bound $\bar{x}_h^{0,[0]} = \bar{x}_h(0) \ni x(h)$ for the solution at $t = h$ as

$$\bar{x}_h^{0,[0]} = \sum_{k=0}^{n+1} \bar{x}_h^{1,[k]} h^k$$

3.2 Numerical Issues

The algorithm presented above works on intervals. However, simple interval arithmetic leads easily to inaccurate results.

Noteworthy are the so called *wrapping effect*, which causes overestimations by representing issues in the chosen basis, and the *dependency problem*, which appears when two different occurrences of the same variable are treated as if they were two distinct variables. Advanced techniques like Lohner's method or zonotopes can reduce their impact.

3.3 Variable Delay

The algorithm presented above can be adopted to a variable delay.

Up to now we had an arbitrary but (for all iterations) fixed τ in the DDE.

$$\dot{x} = f(x(t), x(t-\tau)) \qquad (4)$$

Now we assume that τ can vary over a certain range from one iteration to another, i. e. we have $\tau = \tau_j \in [\tau_{\min}, \tau_{\max}]$ at integration iteration j, starting at $t = \sum_{l=1}^{j} \tau_l$. This means that τ is known at each integration iteration.

The initial condition needs to be given on $[-\tau_{\max}, 0]$ (therefore we set $\tau_0 := \tau_{\max}$) and $p \in \mathbb{N}$ needs to be fixed such that $(\tau_{\max} - \tau_{\min}) > \frac{1}{p}$.

At integration iteration j, three different cases need to be distinguished:

1. If $\tau_{j+1} = \tau_j$, the algorithm can be applied as explained in section 3.1.

2. If $\tau_{j+1} < \tau_j$, do the interpolation step given below in the first iteration of the integration procedure to obtain a representation of the initial condition on the smaller grid of size $h_{j+1} = \frac{\tau_{j+1}}{p}$. Continue then with the usual algorithm on the new coefficients and grid.

3. If $\tau_{j+1} > \tau_j$, do the interpolation step given below. In order only to interfere with the solution on the last interval, we

impose the condition $h_{j+1}(p-1) \leq \tau_j$, i.e. $\tau_{j+1} \leq \frac{p}{p-1}\tau_j$.

3.3.1 Interpolation Step

If the changing of τ demands an interpolation step, the first iteration of the integration scheme is altered in the following way

Shift Part

1. For $i = 1$ one can just set

$$\bar{x}^{1,[0]}_{h_{j+1}} = \bar{x}^{0,[0]}_{h_j}$$

This holds also for $k > 0$, but these coefficients $\bar{x}^{0,[k]}_{h_j}$ need to be computed in the forward part first.

2. For $i > 1$, interpolate the coefficients onto the grid given by the new τ_{j+1}:

$$\begin{aligned}\bar{x}^{i,[k]}_{h_{j+1}} &= \left\{x^{[k]}_{h_j}(-(i-1)h_{j+1})\right\} \\ &\subseteq \sum_{l=k}^{n+1} \binom{l}{k} \xi^{l-k} \bar{x}^{\eta,[l]}_{h_j}\end{aligned}$$

where $\xi \in [0, h_j)$, such that $-\eta h_j + \xi = -(i-1)h_{j+1}$ for each $i = 2, \ldots, p$ and $0 < k \leq n+1$.

For $k = n+1$ the formula simply reduces to the corresponding remainder interval.

Forward Part All other coefficients and the remainder interval can be obtained with the already known procedure based on the interpolated values.

3.3.2 Continue

One can then continue with the usual algorithm starting from $\bar{x}^{i,[k]}_{h_{j+1}}$ since τ was supposed to stay constant for the time interval $[0, \tau_{j+1}]$. Once τ changes again restart the algorithm as explained above.

3.4 Multivariate Version

We are now interested in adapting the Taylor method integration scheme to delay differential equation whose initial condition depends on parameters. A first approach uses the multivariate version of the Taylor expansion.

3.4.1 Setting

Consider for an arbitrary but fixed $\tau > 0$ the real-valued DDE

$$\dot{x} = f(x(t), x(t-\tau))$$

with initial condition for $t \in [-\tau, 0]$

$$x(t) = x_0(t, \alpha_1, \ldots, \alpha_m).$$

The initial condition is supposed to depend on $k \in \mathbb{N}$ parameters $\alpha_i \in \mathbb{R}$, where α_0 is associated to t.

3.4.2 Taylor Model

One can develop x in a Taylor series expansion around the origin as

$$\begin{aligned}&x(t, \alpha_1, \ldots, \alpha_k) = \\ &\sum_{|K|\leq n} \frac{1}{K!}\partial^K x(0, \ldots, 0) \cdot (t, \alpha_1, \ldots, \alpha_m)^K + \\ &\qquad + R(t, \alpha_1, \ldots, \alpha_k)\end{aligned}$$

having appear all partial derivatives up to order n and using common multi-index notation for $K \in \mathbb{N}^{k+1}$. The remainder term is given by

$$\begin{aligned}&R(t, \alpha_1, \ldots, \alpha_k) = \\ &\qquad \sum_{|K|=n+1} \frac{\partial^K x(c)}{K!} \cdot (t, \alpha_1, \ldots, \alpha_m)^K\end{aligned}$$

for a certain $c \in \{s(t, \alpha_1, \ldots, \alpha_k)^T : s \in [0, 1]\}$, where the partial derivatives of order $n+1$ appear.

3.4.3 (p,n)-Representation

The general idea is adapted from above: represent a function (here the solution and the initial condition) by the coefficients of its Taylor series on the equidistant grid $\{-i\frac{\tau}{p} : i = 0, \ldots, p\}$. We develop around each grid point up to order n and obtain a remainder term of order $n+1$.

To represent a function, one needs all its Taylor coefficients. Unfortunately, their number grows exponentially, as demonstrated by the following lemma.

Lemma 1 (Number of Coefficients)**.** *Let $n, k \in \mathbb{N}$ (zero included). Then there are*

$$\frac{(n+k)!}{k!\,n!}$$

multi indices K with $|K| \leq n$.

Proof. For $n = 0$ there is only one multi index (all components zero). For $n = 1$ there are k. For $n > 1$, one can choose k arbitrary elements from $\mathbb{N}_1^k$ with putting back and without considering the order. One has $\mathbb{N}_n^k = \binom{k+n-1}{n}$ possibilities to do so.

Summing up leads to

$$\left|\mathbb{N}_{\leq n}^k\right| = 1 + k + \sum_{i=2}^{n} \binom{k+i-1}{i} = \frac{n+1}{k}\binom{n+k}{n+1} = \frac{(k+n)!}{k!\,n!}$$

□

The forward propagation of the coefficients demands the evaluation of the derivatives of the DDE's right hand side f. From the multidimensional chain rule, the *Faa di Bruno formula*, cf.

> G. M. Constantine and T. H. Savits. *A Multivariate Faa Di Bruno Formula With Applications*, Transactions Of The American Mathematical Society. Volume 348, Number 2, February 1996.

we can follow that all possible partial derivatives and hence all multivariate Taylor coefficients of the initial condition get involved.

Obviously, this approach is rather expensive, since a large number of coefficients need to be treated at each iteration step. For that reason, we consider another approach in the following section.

3.5 Zonotopic Rigorous DDE Integration

We consider again the setting of the previous section of the DDE with a parameter dependent initial condition. To determine a over-approximation of the solutions which still depends on the parameters, we follow a zonotopic approach representing the intervals by affine forms, which are presented in

> Eric Goubault and Sylvie Putot. *A Zonotopic Framework For Functional Abstractions*, Formal Methods in System Design. Volume 47, Number 3, January 2016.

The algorithm for rigorous integration given in the thesis which is summarized in section 3.1 can directly be adapted to affine arithmetic by replacing the interval arithmetic operations by their corresponding affine forms. The adaption is straight forward and further explained and illustrated by an example hereafter.

3.5.1 Setting

Consider for an arbitrary but fixed $\tau > 0$ the real-valued DDE

$$\dot{x} = f(x(t), x(t-\tau))$$

with initial condition on $t \in [-\tau, 0]$

$$x(t) = x_0(t, \beta_1, \ldots, \beta_L)$$

The initial condition is assumed to depend on parameters $\beta_j \in [b_{j,1}, b_{j,2}]$, which parameterize a whole family of initial functions. Expressed as affine form, β_j can be represented by $m+1$ parameters $\alpha_i \in \mathbb{R}$ and the noise symbols $\epsilon_i \in [-1, 1]$:

$$\beta_l = \alpha_0 + \sum_{i=1}^{m_j} \alpha_i \epsilon_i$$

The initial condition can then be written in dependence of all these noise symbols as

$$x_0(t, \beta) = x_0(t, \epsilon_0, \ldots, \epsilon_m)$$

Remark: This is different to the case of the multivariate version where α_0 was associated to the time t.

3.5.2 Algorithm

We keep to describe the solution and the initial condition of the DDE by their (p,n)-representation. However this time, the coefficients $\bar{x}^{i,[k]}$ are expressed as affine forms instead of simple intervals.

Shift Part The shift part does not involve any interval operation and can hence be used without any adaptions.

The Forward Part The forward part does involve several interval operations. Additions and scalar multiplication are straight forward and interval multiplication is done using the affine variant. This can create new noise symbols which need to be considered in the following.

The derivative obtained by automatic differentiation needs to handle affine forms as parameters. Since the automatic differentiation algorithm is internally doing only basic arithmetic operations, these can be replaced by their affine variants.

The most advanced calculation needed in this part is the calculation of a rough enclosure. For this standard algorithms found in literature can be used. They don't need to do their calculations in affine arithmetic, it is sufficient to express their result as affine form and to continue the procedure.

3.5.3 Example

To demonstrate how the algorithm for the rigorous integration of a delay differential equation can be adapted to affine forms, we consider the following example DDE

$$\begin{cases} \dot{x}(t) = -x(t) \cdot x(t-\tau) \\ \qquad =: f\left(x(t), x(t-\tau)\right) \\ x(t) = x_0(t;\beta) = (1+\beta t)^2 \quad t \in [-\tau, 0] \end{cases}$$

Considering $\beta \in \left[\frac{1}{3}, 1\right]$ gives a family of possible initial functions. We want to calculate an envelope which contains the solutions for every β on the time interval $\left[0, \frac{1}{3}\right]$.

On $t \in [0,\tau]$ the exact solution of the DDE is given by

$$x(t) = \exp\left(-\frac{1}{3\beta}\left((1+(t-1)\beta)^3 - (1-\beta)^3\right)\right)$$

It can be obtained by separation of variables and replacing $x(t-\tau)$ by the initial function

$$\frac{\dot{x}(t)}{x(t)} = -x(t-1) = -x_0(t-1) = -\left(1+(t-1)\beta\right)^2$$

Integration and $x(0) = x_0(0) = 1$ lead to

$$\ln|x(t)| = -\int_0^t \left(1+(s-1)\beta\right)^2 ds = -\frac{1}{3\beta}\left((1+(t-1)\beta)^3 - (1-\beta)^3\right)$$

For later use we note

$$\dot{f}(x(t), x(t-\tau)) = -\dot{x}(t)x(t-\tau) - x(t)\dot{x}(t-\tau)$$

and

$$\ddot{f}(x(t), x(t-\tau)) = -2\dot{x}(t-\tau)\dot{x}(t) - x(t-\tau)\ddot{x}(t) - x(t)\ddot{x}(t-\tau)$$

In the following, we fix $\tau = 1$ and denote by $\odot$, $\oplus$ and $\ominus$ interval operations in the sense of affine arithmetic.

Using affine arithmetic, the parameter β can be written as $\beta = \frac{2}{3} + \frac{1}{3}\epsilon_1 = \alpha_0 + \alpha_1\epsilon_1$ with the noise symbol $\epsilon_1 \in [-1,1]$. The initial condition is rewritten in dependance of the noise symbols ϵ_1 and ϵ_2,

$$x_0(t;\epsilon_1,\epsilon_2) = 1 + 2t\underbrace{(\alpha_0 + \alpha_1\epsilon_1)}_{=\beta} + t^2\underbrace{\left(\alpha_0^2 + \frac{1}{2}\alpha_1^2 + 2\alpha_0\alpha_1\epsilon_1 + \frac{1}{2}\alpha_1^2\epsilon_2\right)}_{=\beta^2}$$

having $\beta^2 = \beta \odot \beta = \frac{1}{2} + \frac{4}{9}\epsilon_1 + \frac{1}{18}\epsilon_2$.

We determine the coefficients for the $(p = 3, n = 1)$-representation of x_0 in affine form:

$$\begin{aligned}
\bar{x}_0^{0,[0]} &= x_0(0) = 1 \\
\bar{x}_0^{1,[0]} &= x_0\left(-\tfrac{1}{3}\right) = \left(1 - \tfrac{\beta}{3}\right)^2 = 1 - \tfrac{2}{3}\beta + \tfrac{1}{9}\beta^2 \\
&= \tfrac{11}{18} - \tfrac{14}{81}\epsilon_1 + \tfrac{1}{162}\epsilon_3 \\
\bar{x}_0^{1,[1]} &= \dot{x}_0\left(-\tfrac{1}{3}\right) = 2\beta\left(1 - \tfrac{\beta}{3}\right) \\
&= \tfrac{29}{27} + \tfrac{10}{27}\epsilon_1 + \tfrac{1}{27}\epsilon_4 \\
\bar{x}_0^{2,[0]} &= x_0\left(-\tfrac{2}{3}\right) = \left(1 - \tfrac{2\beta}{3}\right)^2 = 1 - \tfrac{4}{3}\beta + \tfrac{4}{9}\beta^2 \\
&= \tfrac{1}{3} - \tfrac{20}{81}\epsilon_1 + \tfrac{2}{81}\epsilon_5 \\
\bar{x}_0^{2,[1]} &= \dot{x}_0\left(-\tfrac{2}{3}\right) = 2\beta\left(1 - \tfrac{2\beta}{3}\right) \\
&= \tfrac{22}{27} + \tfrac{2}{27}\epsilon_1 + \tfrac{2}{27}\epsilon_6 \\
\bar{x}_0^{3,[0]} &= x_0\left(-\tfrac{3}{3}\right) = (1-\beta)^2 = 1 - 2\beta + \beta^2 \\
&= \tfrac{1}{6} - \tfrac{2}{9}\epsilon_1 + \tfrac{1}{18}\epsilon_7 \\
\bar{x}_0^{3,[1]} &= \dot{x}_0\left(-\tfrac{3}{3}\right) = 2\beta(1-\beta) \\
&= \tfrac{5}{9} - \tfrac{2}{9}\epsilon_1 + \tfrac{1}{9}\epsilon_8
\end{aligned}$$

and the remainder interval for every $i \in \{1,2,3\}$

$$\bar{x}_0^{i,[2]} = \beta^2 = \frac{1}{2} + \frac{4}{9}\epsilon_1 + \frac{1}{18}\epsilon_2$$

using $\dot{x}_0(t) = 2\beta\left(1+\beta t\right)$ and $\ddot{x}_0(t) = 2\beta^2$.

We perform one iteration step of the algorithm to determine the coefficients of the solution x at $t = h = \frac{1}{3}$. Most coefficients are

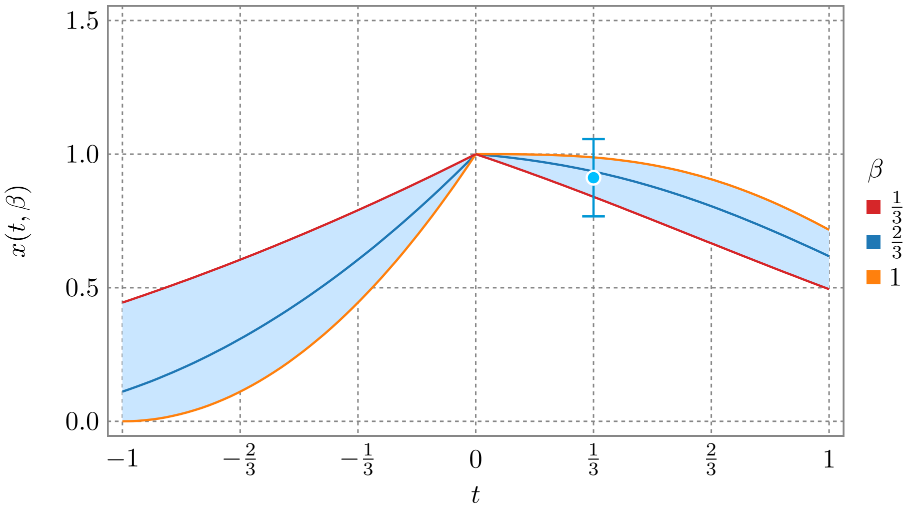

Figure 1: Exact envelope of the initial condition and the solution of the DDE. The obtained rigorous bound at $t = \frac{1}{3}$ is indicated.

obtained by shifting:

$$\begin{aligned}
\bar{x}_h^{1,[0]} &= \bar{x}_0^{0,[0]} \\
\bar{x}_h^{2,[0]} &= \bar{x}_0^{1,[0]} \\
\bar{x}_h^{3,[0]} &= \bar{x}_0^{2,[0]} \\
\bar{x}_h^{2,[1]} &= \bar{x}_0^{1,[1]} \\
\bar{x}_h^{3,[1]} &= \bar{x}_0^{2,[1]}
\end{aligned}$$

The rest is calculated in the forward part

$$\begin{aligned}
\bar{x}_h^{1,[1]} &= f(x(0), x_0(-1)) = -\bar{x}_h^{1,[0]} \odot \bar{x}_0^{3,[0]} \\
&= -\tfrac{1}{6} + \tfrac{2}{9}\epsilon_1 - \tfrac{1}{18}\epsilon_7
\end{aligned}$$

As depicted in figure 1, the connection of initial function and corresponding solution is continuous ($x_0(0) = x(0)$), but their derivatives are not ($\dot{x}_0(0) \neq \dot{x}(0)$).

The new remainder interval is determined by

$$\begin{aligned}
\bar{x}_h^{1,[2]} &\subseteq \tfrac{1}{2!}\ddot{x}(0) + \tfrac{1}{2!}x^{(3)}([0,h]) \odot [0,h] \\
&= \tfrac{1}{2}\dot{f}(x(0), x_0(-1)) \\
&\quad + \tfrac{1}{2}\ddot{f}(x([0,h]), x_0([-1,h-1])) \odot [0,h] \\
&= \tfrac{1}{2}F_1 + \tfrac{1}{2}F_2 \odot [0,h]
\end{aligned}$$

where by (automatic) differentiation we have

$$\begin{aligned}
&F_1\left(x_0^{3,[0]}, x_0^{3,[1]}, x_h^{1,[0]}, x_h^{1,[1]}\right) := \\
&\qquad \frac{d}{dt} f(x(0), x_0(-1)) \\
&\qquad = -\bar{x}_h^{1,[1]} \odot \bar{x}_0^{3,[0]} \ominus \bar{x}_h^{1,[0]} \odot \bar{x}_0^{3,[1]} \\
&\qquad = -\tfrac{325}{648} + \tfrac{4}{27}\epsilon_1 + \tfrac{1}{54}\epsilon_7 - \tfrac{1}{9}\epsilon_8 + \tfrac{11}{216}\epsilon_{12}
\end{aligned}$$

and

$$\begin{aligned}
&F_2\Big(x_0^{[0]}([-1,h-1]), x_0^{[1]}([-1,h-1]), \\
&\qquad x_0^{[2]}([-1,h-1]), x^{[0]}([0,h]), \\
&\qquad x^{[1]}([0,h]), x^{[2]}([0,h])\Big) \\
&\qquad = -2 \cdot x_0^{[1]}([-1,h-1]) \odot x^{[1]}([0,h]) \\
&\quad \ominus 2 \cdot x_0^{[2]}([-1,h-1]) \odot x_0^{[0]}([-1,h-1]) \\
&\qquad\qquad \odot 2 \cdot x^{[2]}([0,h]) \ominus x^{[0]}([0,h])
\end{aligned}$$

For this expression, we need to calculate upper bounds in affine form of the coefficients over the interval $[0,h] = \frac{1}{6} + \frac{1}{6}\epsilon_9$

$$\begin{aligned}
x_0^{[0]}([-1,h-1]) &= (1 + \beta \odot [-1,h-1])^2 \\
&= \tfrac{79}{324} - \tfrac{20}{81}\epsilon_1 - \tfrac{8}{81}\epsilon_9 + \tfrac{4}{81}\epsilon_13 + \tfrac{49}{324}\epsilon_14 \\
x_0^{[1]}([-1,h-1]) &= 2\beta \odot (1 + \beta \odot [-1,h-1]) \\
&= \tfrac{37}{54} - \tfrac{2}{27}\epsilon_1 - \tfrac{4}{27}\epsilon_9 + \tfrac{2}{27}\epsilon_15 + \tfrac{11}{54}\epsilon_16 \\
x_0^{[2]}([-1,h-1]) &= 2\beta^2 = \tfrac{1}{2} + \tfrac{4}{9}\epsilon_1 + \tfrac{1}{18}\epsilon_2
\end{aligned}$$

Additionally, we need a rough enclosure of $x^{[2]}([0,h])$, which we guess as

$$\tilde{x}_h^{1,[2]} = \tfrac{1}{2} \cdot [-1,0] = -\tfrac{1}{4} + \tfrac{1}{4}\epsilon_{11}$$

in order to determine the remaining coefficients using formulae (1) and (2)

$$\begin{aligned} & x^{[0]}([0,h]) = \\ & \bar{x}_h^{1,[0]} \oplus \left(\bar{x}_h^{1,[1]} \odot [0,h]\right) \oplus \left(\tilde{x}_h^{1,[2]} \odot [0,h]^2\right) \\ & \quad = \tfrac{277}{288} + \tfrac{1}{27}\epsilon_1 - \tfrac{1}{108}\epsilon_7 - \tfrac{1}{24}\epsilon_9 - \tfrac{1}{288}\epsilon_{10} \\ & \qquad + \tfrac{1}{96}\epsilon_{11} + \tfrac{5}{108}\epsilon_{17} + \tfrac{5}{288}\epsilon_{18} \end{aligned}$$

and

$$\begin{aligned} & x^{[1]}([0,h]) = \bar{x}_h^{1,[1]} \oplus \left(2 \cdot [0,h] \odot \tilde{x}_h^{1,[2]}\right) \\ & = -\tfrac{1}{4} + \tfrac{2}{9}\epsilon_1 - \tfrac{1}{18}\epsilon_7 - \tfrac{1}{12}\epsilon_9 + \tfrac{1}{12}\epsilon_{11} + \tfrac{1}{12}\epsilon_{19} \end{aligned}$$

Thus we have

$$\begin{aligned} F_2 = & -\tfrac{4219}{7776} - \tfrac{1319}{972}\epsilon_1 - \tfrac{277}{2592}\epsilon_2 + \tfrac{83}{972}\epsilon_7 \\ & + \tfrac{7}{216}\epsilon_9 + \tfrac{1}{288}\epsilon_{10} - \tfrac{71}{288}\epsilon_{11} + \tfrac{2}{81}\epsilon_{13} + \tfrac{49}{648}\epsilon_{14} \\ & + \tfrac{1}{27}\epsilon_{15} + \tfrac{11}{108}\epsilon_{16} - \tfrac{5}{108}\epsilon_{17} - \tfrac{5}{288}\epsilon_{18} \\ & - \tfrac{37}{324}\epsilon_{19} - \tfrac{461}{972}\epsilon_{20} - \tfrac{59}{216}\epsilon_{21} - \tfrac{1159}{7776}\epsilon_{22} \end{aligned}$$

and for the new remainder coefficient

$$\begin{aligned} & \bar{x}_h^{1,[2]} = \tfrac{1}{2}F_1 + \tfrac{1}{2}F_2 \odot [0,h] = \\ & - \tfrac{27493}{93312} - \tfrac{455}{11664}\epsilon_1 - \tfrac{277}{31104}\epsilon_2 + \tfrac{191}{11664}\epsilon_7 \\ & - \tfrac{1}{18}\epsilon_8 - \tfrac{3967}{93312}\epsilon_9 + \tfrac{1}{3456}\epsilon_{10} - \tfrac{71}{3456}\epsilon_{11} \\ & + \tfrac{11}{432}\epsilon_{12} + \tfrac{1}{486}\epsilon_{13} + \tfrac{49}{7776}\epsilon_{14} + \tfrac{1}{324}\epsilon_{15} \\ & + \tfrac{11}{1296}\epsilon_{16} - \tfrac{5}{1296}\epsilon_{17} - \tfrac{5}{3456}\epsilon_{18} - \tfrac{37}{3888}\epsilon_{19} \\ & - \tfrac{461}{11664}\epsilon_{20} - \tfrac{59}{2592}\epsilon_{21} - \tfrac{1159}{93312}\epsilon_{22} + \tfrac{24331}{93312}\epsilon_{23} \end{aligned}$$

We finally obtain the rigorous bound for the value of the solution after one (sub-)time step as

$$\begin{aligned} \bar{x}_h^{0,[0]} & = \sum_{k=0}^{2} h^k \cdot \bar{x}_h^{1,[k]} \\ & = \bar{x}_h^{1,[0]} \oplus \left(h \cdot \bar{x}_h^{1,[1]}\right) \oplus \left(h^2 \cdot \bar{x}_h^{1,[2]}\right) \\ & = \tfrac{765659}{839808} + \tfrac{7321}{104976}\epsilon_1 - \tfrac{277}{279936}\epsilon_2 - \tfrac{1753}{104976}\epsilon_7 \\ & - \tfrac{1}{162}\epsilon_8 - \tfrac{3967}{839808}\epsilon_9 + \tfrac{1}{31104}\epsilon_{10} - \tfrac{71}{31104}\epsilon_{11} \\ & + \tfrac{11}{3888}\epsilon_{12} + \tfrac{1}{4374}\epsilon_{13} + \tfrac{49}{69984}\epsilon_{14} + \tfrac{1}{2916}\epsilon_{15} \\ & + \tfrac{11}{11664}\epsilon_{16} - \tfrac{5}{11664}\epsilon_{17} - \tfrac{5}{31104}\epsilon_{18} - \tfrac{37}{34992}\epsilon_{19} \\ & - \tfrac{461}{104976}\epsilon_{20} - \tfrac{59}{23328}\epsilon_{21} - \tfrac{1159}{839808}\epsilon_{22} + \tfrac{24331}{839808}\epsilon_{23} \end{aligned}$$

This means that

$$\begin{aligned} x\left(\tfrac{1}{3};\beta\right) \in \gamma\left(\bar{x}_h^{0,[0]}\right) & = \left[\tfrac{5965}{7776}, \tfrac{443549}{419904}\right] \\ & \approx [0.767, 1.056] \end{aligned}$$

for every $\beta \in \left[\tfrac{1}{3}, 1\right]$.

The exact interval obtained using the explicit solution of the DDE is given by $[0.840, 0.988]$.

4 Literature

Further related literature includes

Clement E. Falbo, *Some Elementary Methods for Solving Functional Differential Equations.*

giving an introduction to integrating DDEs and

Larry F. Shampine, and Sylvester Thompson. *Numerical Solution of Delay Differential Equations*, Delay Differential Equations. 2009.

A Matlab-Toolbox for solving DDEs is presented in

Nicolas W. Bauer et al. *Networked Control Systems Toolbox: Robust Stability Analysis Made Easy*, 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems. 2012.

For Taylor models, a suggested reading is

Martin Berz, and Kyoko Makino. *Verified Integration of ODEs and Flows Using Differential Algebraic Methods on High-Order Taylor Models*, Reliable Computing. 1998.

and some examples and applications are presented in

Xiaoqing Jin et al. *Powertrain Control Verification Benchmark*, 17th International Conference on Hybrid Systems: Computation and Control. ACM, 2014.