

SWIM PALaiseau/VERSAILLES 2019

SUMMER WORKSHOP ON INTERVAL METHODS



Book of Abstracts

Julien Alexandre dit Sandretto
Olivier Mullier
Alexandre Chapoutot



Published by:
SSH
U2IS
ENSTA Paris
Institut Polytechnique de Paris
<https://swim2019.ensta-paristech.fr/>

Printed in Palaiseau by Ensta Paris — July 2019

Welcome message

On behalf of the SWIM'19 organizing committee, I am honored and delighted to welcome you to the 12th Summer Workshop on Interval methods, Palaiseau, France. This year SWIM is hosted by ENSTA Paris. ENSTA Paris belongs to the foremost graduate schools of engineering in France. It is a public educational and research establishment, self-governed under the supervision of the Ministry of Defence. Organizing committee wish to thank the school for the event hosting and the help in the organization.

Few words about our history: ENSTA ParisTech was originally the brainchild of Henry-Louis Duhamel du Monceau, inspector general of the Navy. He had identified the need to give the Navy's master carpenters a theoretical education, particularly in mathematics and physics, which were making quick progress, so that they would have a clearer understanding of their trade. After founding the first school in Toulon, he transferred it to Paris in 1741. This date is recognised as the origin of our institution. After undergoing 7 lean years of under-funding, he managed to persuade the duc de Choiseul to reopen it in 1765, and continued to run the school for the rest of his life. At the time, the institute, called School of Engineer-Constructors of Royal Vessels, was housed in the Louvre Palace. Later on, it became known as "Ecole nationale supérieure du Génie Maritime" (National Higher College of Maritime Engineering).

SWIM 2019: Our technical program is rich and varied with 25 contribution papers (each one reviewed by two people) and two round tables. The success of this edition depends on you, the participants, so thank you for your contribution !

This volume contains the Book of abstracts of The 12th Summer Workshop on Interval Methods SWIM 2019. The tradition of SWIM workshops was set up in France in 2008, and since that time it is held annually by the effort of Luc Jaulin and Nacim Ramdani. The workshop joins people from different communities working with interval methods. Thus, it provides a unique opportunity to meet scientists from robotics, optimization, control, estimation, verification and other areas.

J. Alexandre dit Sandretto, Workshop Chair

WORKSHOP PROGRAM

July 23rd, Tuesday

Interval Analysis: Implementation and Libraries

- 5 *K. A. Nasiotis, D. López, S. P. Adam, and L. G. Casado*
Set Inversion Via Interval Analysis – A Study on Parallel Processing Implementation
- 9 *N. Revol*
The MPFI library revisited
- 11 *Evgeniya Vorontsova*
Interval Computations in Julia programming language
- 15 *L. Benet, M. Forets, D. P. Sanders, and C. Schilling*
TaylorModels.jl: Taylor models in Julia and its application to validated solutions of ODEs

Differential Equations I

- 17 *J. Brown, F. Pessaux*
Interval-Based Simulation of Zélus IVPs Using DynIbex
- 21 *S. Selivanova, M. Ziegler*
Turnkey Solutions to PDEs in Exact Real Computation
- 23 *J. Damers, L. Jaulin, and S. Rohou*
Guaranteed interval integration for large initial boxes

July 24th, Wednesday

Robust control I

- 27 *J. Tillet, L. Jaulin, F. Le Bars*
Validation of a controller under state constraints
- 31 *A. Lefort*
Efficient computation of the set of stabilizing controllers for an LTI System using intervals
- 35 *J. Kersten, A. Rauh, H. Aschemann*
Analyzing Uncertain Dynamical Systems After State-Space Transformations Into Co-operative Forms

- 37 *T. Nico, L. Jaulin, and B. Zerr*
Guaranteed Polynesian Navigation

July 25th, Thursday

State Estimation

- 41 *D. Merhy, C. Stoica Maniu, T. Alamo, E. F. Camacho, T. Chevet, and M. Makarov*
Zonotopic set-membership state estimation applied to an octorotor model
- 45 *T. Gatto, L. Meyer, H. Piet-Lahanier*
A Polytopic Box Particle Filter for state estimation of Non Linear Discrete-Time Systems

Computation and methods

- 49 *M. Lange*
Rigorous bounds for ill-posed linear programming problems
- 53 *O. Mullier, J. Alexandre dit Sandretto*
Computation of integrals with interval endpoints

Fault detection and calibration

- 55 *S. Liu, J.-J. Gehrt, D. Abel, and R. Zweigel*
Identification of Multi-Faults in GNSS Signals using RSIVIA under Dual Constellation
- 59 *H. Dbouk, S. Schön*
Interval based Fault Detection and Exclusion for GNSS
- 63 *R. Voges, B. Wagner*
Extrinsic Calibration Between a 3D Laser Scanner and a Camera Under Interval Uncertainty

Differential Equations II

- 67 *A. Rauh, J. Kersten*
Toward the Development of Iteration Procedures for the Interval-Based Simulation of Fractional-Order Systems
- 69 *J. Alexandre dit Sandretto*
Confidence-based Contractor, Propagation and Potential Cloud for Differential Equations
-

July 26th, Friday

Localization

- 73 *V. Drevelle*
Bounded-error victim localization for UAV-based search and rescue operations
- 77 *N. Ramdani, D. Zeinalipour-Yazti, M. Karamousadakis, A. Panayides*
Towards an interval fingerprinting approach for indoor localization
- 81 *S. Rohou, P. Franek, C. Aubry, L. Jaulin*
Verifying the existence of loops in robot trajectories

Robust control II

- 83 *A. Kumar, O. Mullier*
Guaranteed Trajectory Tracking using Flatness
- 85 *É. Bertin, E. Brendel, B. Hérissé, A. Chapoutot, and J. Alexandre dit Sandretto*
Prospects on the application of necessary optimality conditions on the resolution of the Goddard problem with unknown bounded parameters using interval arithmetics

87 **List of Authors**

Set Inversion Via Interval Analysis: A Study on Parallel Processing Implementation^{*†}

K. A. Nasiotis^{1,2}, D. López², S. P. Adam¹, and L. G. Casado²

¹Dept. of Informatics and Telecommunications, University of Ioannina, Greece

²Supercomputing Group, University of Almería (CeIA3), Spain

Keywords: Interval analysis; Set membership techniques; Parallel branch-and-bound

Introduction

Among the success stories of interval computation [5] one may report the application of interval based global optimization, validation of numerical calculations, the use of intervals for modeling uncertainty and dealing with uncertain systems, etc.

This work focuses on Set Inversion Via Interval Analysis (SIVIA) [4] a method used for solving problems such as nonlinear parameter, state or error estimation for systems operating under bounded uncertainty. The method is primarily a set membership technique designed to solve such estimation problems provided that system operation is described by some analytic function $f : X \subseteq \mathbb{R}^n \rightarrow Y \subseteq \mathbb{R}^m$ for which some suitable interval extension $[f] : \mathbb{IR}^n \rightarrow \mathbb{IR}^m$ can be defined. Then, given an interval vector, i.e. a box, $[y] \subseteq Y$, one needs to determine the set of unknown vectors $x \in X$ such that $f(x) \in [y]$. SIVIA starts with an initial box $[X_0]$ such that $X \subseteq [X_0] \subseteq \mathbb{IR}^n$ and computes an approximation of the set of interest $S = \{x \in X \subseteq \mathbb{R}^n | f(x) \in [y]\} = f^{-1}([y]) \cap X$ as a union of axes aligned boxes.

Computation explores the search space $[X_0]$ applying a branch-and-bound, or more precisely, given the type of processing, a branch-and-prune (B&P) strategy whose performance depends on the size of the problem i.e. the size

of the search space $[X_0]$, its dimension, the function f itself, the distribution of the vectors of interest $x \in [X_0]$ and the “resolution” adopted for the approximation of the set S . SIVIA has been successfully applied in control systems problems with few parameters. For problems with higher dimensions, larger sized input space and fine “resolution” the performance of SIVIA deteriorates severely and becomes practically inapplicable. Hence, the need to investigate the possibility of a parallel implementation towards obtaining some affordable computational cost.

Background and Related Work

Interval computations have proven to be extremely important to a number of problems for which errors produced by calculations or due to uncertainty that can be modeled in terms of intervals. However, Kreinovich [6] provides a number of convincing arguments that interval computations are NP-hard and the only way to deal with NP-hardness is to use parallel versions of the algorithms, when this is feasible.

SIVIA itself relies on interval computations and it is also known to suffer from the curse of dimensionality which is inherent to its B&P processing style. A recent example of this argument is reported by the work of Adam et al. [1] who formulated the problem of estimating generalization of a multilayer perceptron as a parameter estimation problem and used SIVIA for exploring search spaces such as $[-1, 1]^{10}$. The sequential implementation of SIVIA in such experiments gave extremely interesting results but at the same time it proved the practical impossibility of SIVIA

^{*}This paper has been supported by the Spanish Ministry (RTI2018-095993-B-100), in part financed by the European Regional Development Fund (ERDF).

[†]Konstantinos A. Nasiotis placement in (CeIA3) is financed by the ERASMUS+ Program.

to cope with higher dimensions. This deficiency led to the definition of the so-called contractor programming in order to diminish the size of the boxes explored by SIVIA [3], while some practitioners tried to effectively parallelize SIVIA [7, 8].

As reported in [8], Marvel et al. used SIVIA for estimating model parameters in biological systems, under bounded conditions of uncertainty. They adopted a parallel implementation using multiple processing cores and they developed a method for use on a single multi-core workstation using POSIX threads to process subsets of the parameter space while access to shared information was controlled by mutex-locked linked lists. The results obtained for two biological models, namely, the nonlinear Lotka-Volterra predator-prey model and the SEIR infectious disease model, using 8 threads on an 8-core machine, seem to be satisfactory but it remains unclear what the speedup will be when scaling up the implementation to 16, 32 or more more CPUs.

Another work that merits to be cited here is the effort of Le Ménéec [7] for computing, in real time, the viability kernel as a tool for decision making in autonomous systems. To this end he applied an interval based algorithm based on SIVIA for computing the viability kernel of the underlying nonlinear system. The author proposes a parallel implementation of SIVIA on a multi-core processing system while making use of the contractors concept. The parallel implementation scheme adopted in this work, [7], uses a pool of jobs each one corresponding to the evaluation of a hyper-box. A host streams the jobs to a multi-core system assigning a job to each available core while recovering the results of each completed job. This implementation is a possible way to deal with the inherent deficiency of the IBEX library to share objects between threads i.e. being thread unsafe. Moreover, the author does not present any performance results and so it is not possible to objectively criticize this work.

In the work presented, herein, we adopt

a more elaborated parallelization strategy which uses an Asynchronous Multiple Pool of boxes and a dynamic load balancing scheme based on workload estimation for optimizing the use of the computing resources.

Proposed Parallel Version

Parallelization of B&B style algorithms were widely studied in the literature, see [2] and references therein. In order to simplify the algorithm let us focus on SIVIA when $m=1$. So, we are dealing with determining the “level set” of the function f in the (one dimensional) interval $[y]$. Then, for $f(x) = x_1^2 + x_2^2$, $[y] = [1, 2]$, $[X_0] = [-1.5, 1.5]$ and $\epsilon = 0.05$ the result of the inversion operation is given in Figure 1. In this specific implementation

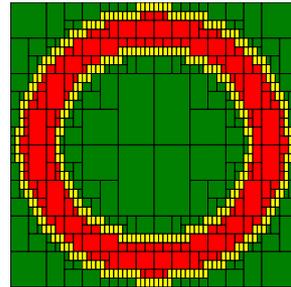


Figure 1: Circle level set in $[1, 2]$ interval. Red: in, yellow: border, green: out.

of SIVIA the selected box is divided by the widest dimension and Depth-First is used to select next box.

Our parallel implementation is based on a Asynchronous Multiple Pool of boxes using a dynamic number of threads [2] in C-XSC which was suitably modified to be thread safe. Dynamic load balancing is provided by generating a new thread, if the current number of threads is smaller than a threshold, and by moving the next selected box (the smallest one) to the new thread. Initial experiments run on a node of Bullion S8 with 8 Intel(R) Xeon(TM) E7 8860v3 @ 2.20GHz (16 cores) and 2.3TB of total RAM. The results, in number of boxes, for the circle problem, shown in Figure 1 for $\epsilon = 10^{-6}$ are: $|L_{in}|=13,501,140$, $|L_{border}|=40,503,776$ and $|L_{out}|= 13,501,276$

and the total number of evaluated boxes is 94,508,607. L_{in} , L_{out} and L_{border} are the lists of the boxes that are inside, outside or on the ϵ -border of the set discovered by SIVIA.

We also tested the parallel algorithm for the Griewank function using the same hardware, $[y]=[1.5, 3]_2$, $[X_0]=[-10, 10]^2$ and $\epsilon=10^{-6}$.

$$\sum_{i=1}^2 \frac{x_i^2}{4000} - \prod_{i=1}^2 \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

The results obtained are: $|L_{in}|=122,424,640$, $|L_{border}|=343,639,512$ and $|L_{out}|=122,445,084$ and the total number of evaluated boxes is 833,378,959. The speedup results for the above experiments are displayed in Figure 2, hereafter.

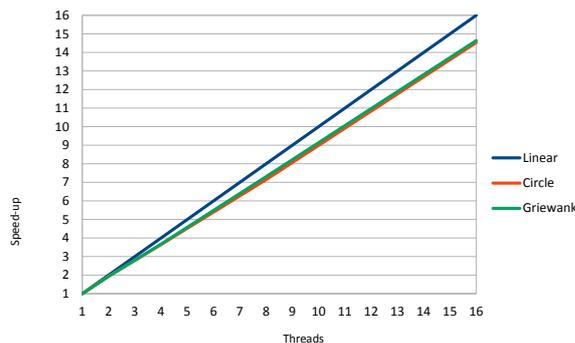


Figure 2: Speedup achieved by the current implementation for the two examples

The number of evaluated boxes increases not only with n but with the type of the problem, as well. The achieved, close to linear, speedup is due to a the dynamic load balancing used, which is based on the estimation of the workload of the boxes pending for evaluation. This allows to select half of the pending workload in order to migrate those boxes to a new thread. Additionally, a Cut-off of the parallelism is used in order to avoid migration of boxes without significant workload (subtree). We expect this strategy to further improve both the parallel version and the sequential one, as the number of evaluated boxes will be reduced.

Conclusion

Despite its efficiency for small-sized estimation problems SIVIA is NP-hard and fails to

address problems of larger size in reasonable time. The aim of this study is to investigate an efficient parallel implementation of SIVIA on multi-core systems. The results obtained, so far, support the prospect of acceptable processing times while underlining the need of focusing on techniques accelerating the parallel version shown here. Scaling up the current implementation to systems with a larger number of cores is of primary concern for this study.

References

- [1] Stavros P. Adam, Aristidis C. Likas, and Michael N. Vrahatis. Evaluating generalization through interval-based neural network inversion. *Neural Computing and Applications*, Mar 2019.
- [2] Juan F. R. Herrera, José M. G. Salmerón, Eligius M. T. Hendrix, Rafael Asenjo, and Leocadio G. Casado. On parallel branch and bound frameworks for global optimization. *Journal of Global Optimization*, 69(3):547–560, Nov 2017.
- [3] Luc Jaulin, Michel Kieffer, Olivier Didrit, and Eric Walter. Interval analysis. In *Applied Interval Analysis*, pages 11–43. Springer, 2001.
- [4] Luc Jaulin and Eric Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053 – 1064, 1993.
- [5] Ralph B. Kearfott. Interval computations: Introduction, uses, and resources. *Euromath Bulletin*, 2(1):95–112, 1996.
- [6] Vladik Kreinovich and Andrew Bernat. Parallel algorithms for interval computations: An introduction. *Interval Computations*, 1994, 01 1994.
- [7] Stéphane Le Ménec. Interval Based Parallel Computing of the Viability Kernel. <https://swim2016.sciencesconf.org/data/pages/LeMenec.pdf>, 2013. SWIM 2016 – Lyon.
- [8] Skylar W. Marvel, Maria A. de Luis Balaguer, and Cranos M. Williams. Parameter Estimation in Biological Systems Using Interval Methods with Parallel Processing. In *Proc. of the 8th International Workshop on Computational Systems Biology*, pages 161–168, Zürich, Switzerland, 2011.

The MPFI library revisited

Nathalie Revol*¹

¹University of Lyon - Inria, France

Keywords: MPFI Library; Arbitrary Precision Interval Arithmetic; IEEE 1788-2015 Compliance

The IEEE 1788-2015 standard

Interval arithmetic has been defined and used since the 50s and 60s. However, no common definition existed for years and it made difficult to compare different works. In 2008, a group of interval experts, gathered at a seminar in Dagstuhl, felt that interval arithmetic was mature enough to undergo a standardization effort. This effort led to the IEEE 1788-2015 standard [2].

It was impossible to define a theory that encompasses the co-existing theories in use, such as set theory, Kaucher arithmetic, modal arithmetic, cset arithmetic. The adopted solution was to provide "hooks" to accommodate different theories within the standard: each provided theory is called a *flavor*. The only flavor defined in the 2015 version of the standard is the set-based flavor, from set theory.

Another peculiarity of the IEEE 1788-2015 standard is the handling of exceptions, called decorations. A *decoration* is attached to each interval and gives a summary of what happened during the computations that resulted in this interval: was every operation defined and continuous over its arguments, or simply defined, or even less, such as in $\sqrt{[-2, 1]}$ where the square root is not defined everywhere over its argument $[-2, 1]$? Incidentally, in the set-based flavor, $\sqrt{[-2, 1]}$ is computed as $\sqrt{[-2, 1] \cap \text{Dom } \sqrt{\cdot}} = \sqrt{[0, 1]} = [0, 1]$.

*Corresponding author.

Libraries compliant with the IEEE 1788-2015 standard

The development of the standard has been accompanied by the development of the C++ `libieee1788` library by M. Nehmeier, that served as a proof-of-concept. Unfortunately, M. Nehmeier left academia and this library is no more maintained. Two other libraries have been developed since then and are compliant with the standard: `JInterval` by D. Nadezhin and S. Zhilin, and the Octave `interval` package by O. Heimlich. The `JInterval` recently and untimely lost its main developer, D. Nadezhin. O. Heimlich also left academia but he still develops and maintains the Octave `interval` package.

No other library of interval arithmetic has been developed in compliance with the IEEE 1788-2015 standard, because it is difficult. A first difficulty is the implementation of the long list of functions and conversions mandated by the standard, with the prescribed accuracy. Another difficulty is the implementation of the decoration mechanism. On the one hand, it requires that an extra piece of information is attached to each interval, and this can destroy memory optimizations (padding etc.). On the other hand, decorations must be propagated and this implies some more extra code. These difficulties are less salient for the MPFI library, introduced below.

The MPFI library

MPFI [3] is a C library for arbitrary precision interval arithmetic. An interval is represented by its endpoints, which are arbitrary precision floating-point numbers provided by the MPFR library [1]. Every single operation is

as accurate as possible, thanks to the MPFI library that provides correctly rounded operations, with directed roundings as needed, for each endpoint.

MPFR already offers a long list of functions and conversions between different types and MPFR floating-point numbers: incorporating them in MPFI is usually relatively easy, for most of them, as the bulk of the work has already been done by MPFR developers. However, some functions mandated by the standard, and in particular most of the *reverse* functions, useful for constraints solving, are still missing in MPFI.

As an interval is represented by two arbitrary, and thus variable, precision endpoints, adding a decoration to each interval is not an issue: padding or cache optimization are not at stake anyway, as the employed memory is already (usually) variable and large.

Finally, the mechanism for handling exceptions in MPFI is very different from the one adopted in the IEEE 1788-2015 standard: for instance, for $\sqrt{[-2, 1]}$, MPFI returns `NaN`, which stands for **Not an Interval**. The code of each MPFI operation must be reworked to handle and propagate decorations.

To sum up, there is some work to be done to make MPFI compliant with the IEEE 1788-2015 standard, but this work seems less demanding than for libraries based on fixed-precision floating-point numbers such as IEEE 754-2008 `binary32` or `binary64`. The relative overhead, both in terms of memory and of computation time, due to the incorporation of flavors and decorations, is also less important and probably negligible.

Work to be done

The main modifications will take place at two levels. The first one concerns the data structure of a MPFI interval.

- a) An extra field will be added to indicate the flavor in use. This is a bit different (but not incompatible) from the intended

use of a flavor, which is supposed to be set for a whole block of code rather than for an individual interval. However, MPFI will check that the flavors of the operands and of the result match before performing the required operation.

- b) An extra field, parameterized by the flavor, will be added to store the decoration attached to the interval.

The second kind of modifications concerns the code for each operation.

- a) A preprocessing will be added to check the compatibility of the flavors and to branch to the code corresponding to the flavor in use.
- b) For each branch, a postprocessing will propagate the decoration.

Lastly, for backward compatibility, a `MPFIoriginal` flavor will be added, that will branch to the original version of MPFI. It will be the default flavor, so that users can run their existing codes without any modification of their behaviour.

References

- [1] F. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann. MPFR: A Multiple-precision Binary Floating-point Library with Correct Rounding. *ACM Transactions on Mathematical Software*, 33(2):no 13, 2007.
- [2] IEEE: Institute of Electrical and Electronic Engineers. IEEE 1788-2015 Standard for Interval Arithmetic, 2015.
- [3] N. Revol and F. Rouillier. Motivations for an Arbitrary Precision Interval Arithmetic and the MPFI Library. *Reliable Computing*, 11(4):275–290, 2005.

Interval Computations in Julia programming language

Evgeniya Vorontsova^{1, 2}

¹Far Eastern Federal University, Vladivostok, Russia

²Grenoble Alpes University, Grenoble, France

Keywords: Numerical Computing; Interval Arithmetic; Julia Programming Language; JuliaIntervals; Octave Interval Package

Introduction

The general-purpose Julia programming language [5] was designed for speed, efficiency, and high performance. It is a flexible, optionally-typed, and dynamic computing language for scientific, numerical and technical computing applications. Julia is open source language with all sources free available on GitHub. The language was developed and incubated at MIT [6]. Currently, after Julia 1.0 was officially released to the public in August 2018, the language is becoming increasingly popular. Julia has been downloaded more than 8.4 million times, as of May 2019 [2], and is used at more than 1,500 universities.

So, it is very important for researchers, working in a field of interval analysis, to have fast, efficient and robust publicly available software packages for performing computations with interval arithmetic written in Julia.

IntervalArithmetic.jl

In this paper, we review and compare a recently developed Julia package for performing Validated Numerics, i.e. rigorous computations with finite-precision floating-point arithmetic, IntervalArithmetic.jl [4], with performance of GNU Octave interval package for real-valued interval arithmetic [1]. This Octave toolbox was chosen for comparison because of several important reasons. First of all, it is a free, open-source software, unlike

INTLAB [9], a Matlab/Octave toolbox for Reliable Computing. The other fundamental difference between INTLAB and GNU Octave interval package is non-conformance of INTLAB to IEEE 1788-2015 — IEEE Standard for Interval Arithmetic [3]. On the other hand GNU Octave interval package's main goal is to be compliant with the Standard. Likewise, authors of IntervalArithmetic.jl wrote [4] that they were working towards having the package be conforming with the Standard. So, all calculations in these packages are performed using interval arithmetic: all quantities are treated as intervals. The final result is also an interval contained the correct answer.

In next section we would like to show some practical examples with interval arithmetic in Julia.

Examples

Getting Started

The basic object in the IntervalArithmetic.jl package is the parameterized type `Interval`. By default, `Interval` objects contain `Float64`s. Intervals are created using the `@interval` macro:

```
using IntervalArithmetic
a = @interval(1, 2)
b = @interval(3, 4)
print(a + b, a - b, a * b, a / b)
```

The output of this code is

```
[4, 6] [-3, -1] [3, 8]
[0.25, 0.666667]
```

As you may have noticed, the package permits to write quite clear and intuitive code for interval computations.

Matrix Multiplication

In this section we present the results of experiments comparing the IntervalArithmetic.jl library with the GNU Octave interval package. In summary, we show that Julia interval library is significantly faster than the Octave library.

In our first experiment we measured the time to evaluate the interval matrix multiplication. The Julia code is:

```
function MultMatr(A, B)
    return A*B
end
n = 10
M1 = 10*rand(n, n)
M2 = 10*rand(n, n)
iM1 = map(Interval, M1)
iM2 = map(Interval, M2)
A = iM1 .± 5
B = iM2 .± 5
@benchmark MultMatr(A,B)
```

Here we use BenchmarkTools package by Jarrett Revels [8], a framework for writing and running groups of benchmarks.

And Octave code for MultMatr function is:

```
pkg load interval
function [t] = MultMatr(n)
    A = infsupdec(rand(n),
        10*rand(n) + 1);
    B = infsupdec(rand(n),
        10*rand(n) + 1);
    tic
    C = A*B;
    t = toc;
end
```

Table 1: Time for interval matrix multiplications

Matrix size, rows	Julia, ms	Octave, ms
10	0.095	13.317
100	111.91	849.61
1000	125870	863340

For Octave we create 10 random interval

matrix pairs and calculate the mean experimental time over all multiplications. The results of the first setting are summarized in Table 1. This experiment shows that performance of Julia interval package for that problem is significantly better.

Elementary functions

In our second experiment we compared the times for evaluation of the elementary functions (exp, sin, cos, etc.) for random interval arguments. The design of the experiment is taken from [7].

Table 2: Time for 10⁵ evaluations of the elementary functions

Function	Julia, s	Octave, s
exp	0.49	102.7
sin	0.749	147.85
cos	0.638	230.2
tan	0.49	126.13
arcsin	0.858	119.01
arccos	1.132	169.02
arctan	1.318	127.01

The results of the second setting are summarized in Table 2. We may see that these calculations in Julia are almost two orders of magnitude faster.

Plotting

In this section, we will illustrate how to visualize the interval extension of a given function over an interval. The process of splitting the interval into many smaller adjacent pieces for range evaluations of the given function is called *mincing*.

Figures 1- 2 show visualization of mincing process for one function (Julia code was adapted from [10]). For implementation The IntervalBox type constructed from an array of Interval was used.

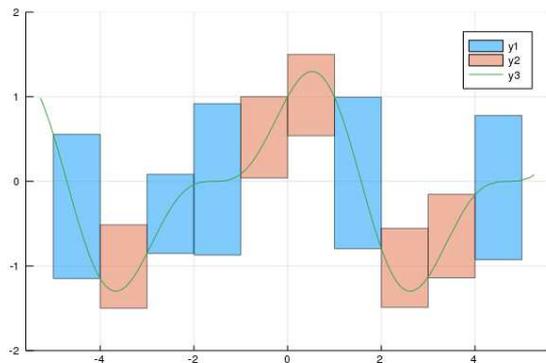


Figure 1: Function $\cos(x) + 0.5 \sin(2x)$, 10 sub-intervals.

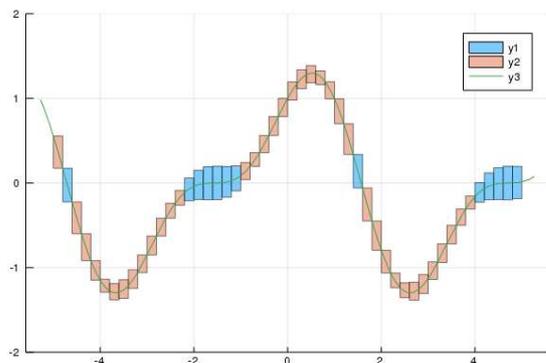


Figure 2: Function $\cos(x) + 0.5 \sin(2x)$, 50 sub-intervals.

Acknowledgement

The work was supported by the Russian Foundation for Basic Research, project no. 18-29-03071 mk.

Conclusion

Public available Julia package for interval arithmetic has been investigated. Experimental comparison of Octave and Julia packages for interval arithmetic shows that Julia IntervalArithmetic.jl package is significantly faster than Octave interval package. In addition, the implementation process of interval arithmetic computations in this Julia package is easy and convenient, due to intuitive syntax of the language and the package.

References

- [1] O. Heimlich GNU Octave interval package, version 3.2.0, 2015–2018, <https://octave.sourceforge.io/interval/>.
- [2] Julia computing newsletter, May 2019. <https://juliacomputing.com/blog/2019/05/03/may-newsletter.html>, 2019.
- [3] IEEE Std 1788-2015 — IEEE Standard for Interval Arithmetic. IEEE: Institute of Electrical and Electronic Engineers, IEEE Computer Society, New York, June 2015, <https://ieeexplore.ieee.org/document/7140721>.
- [4] L. Benet and D. Sanders. JuliaIntervals.jl Package — Rigorous numerics with interval arithmetic & applications. <https://github.com/JuliaIntervals/IntervalArithmetic.jl>.
- [5] J. Bezanson, A. Edelman, S. Karpinski, and V. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59:1, 65–98, 2017.
- [6] J. Bezanson, S. Karpinski, V. Shah, and A. Edelman. Why we created julia. <https://julialang.org/blog/2012/02/why-we-created-julia>, 2012.
- [7] W. Mascarenhas. Moore: Interval Arithmetic in Modern C++, <https://arxiv.org/pdf/1611.09567.pdf>.
- [8] J. Revels. BenchmarkTools Julia package, <https://github.com/JuliaCI/BenchmarkTools.jl>.
- [9] S. Rump. Intlab — interval laboratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, 77–104, Kluwer Academic Publishers, Dordrecht, 1999.
- [10] D. Sanders. Métodos numéricos para sistemas dinámicos, https://github.com/dpsanders/dinamica_nacional.

TaylorModels.jl: Taylor models in Julia and their application to validated solutions of ODEs

Luis Benet^{*1}, Marcelo Forets², David P. Sanders³, and Christian Schilling⁴

¹Instituto de Ciencias Físicas, Universidad Nacional Autónoma de México, México

²Universidad de la República, Uruguay

³Facultad de Ciencias, Universidad Nacional Autónoma de México, México

⁴IST Austria, Klosterneuburg, Austria

Keywords: Taylor models; validated solutions of ODEs; verified numerics

Taylor models, introduced by Berz and Makino [3, 6, 7], define a tool that allows to rigorously bound functions or compute validated solutions of ODEs, among other applications. A Taylor model $M(f) = (p_n, \Delta)$ of an $(n + 1)$ -continuously differentiable function $f(x)$, $x \in D \subset \mathbb{R}^d$ (defined over an open set containing the domain D of interest), is defined by the n -th order Taylor approximation $p_n(x)$ of $f(x)$ around the point $x_0 \in D$, and an interval $[\Delta]$, such that $f(x) \in p_n(x) + [\Delta]$ for all $x \in D$. In its original form, the coefficients of $p(x)$ are floating-point numbers.

This definition was recently extended by M. Joldes [5], where Taylor models with absolute remainder are defined as above (using interval coefficients for the Taylor polynomial as well as the expansion point), and Taylor models with relative remainder are defined by $f(x) \in [p_n](x) + [\Delta]x^{n+1}$ for all $x \in D$.

Here we present `TaylorModels.jl` [1], a package written in Julia for the rigorous approximation of functions in one and several variables. The package implements both Taylor models with absolute remainder for one and several variables and Taylor models with relative remainder for univariate functions. The polynomial coefficients may be floating-point numbers or intervals, and allow to per-

^{*}Corresponding author.

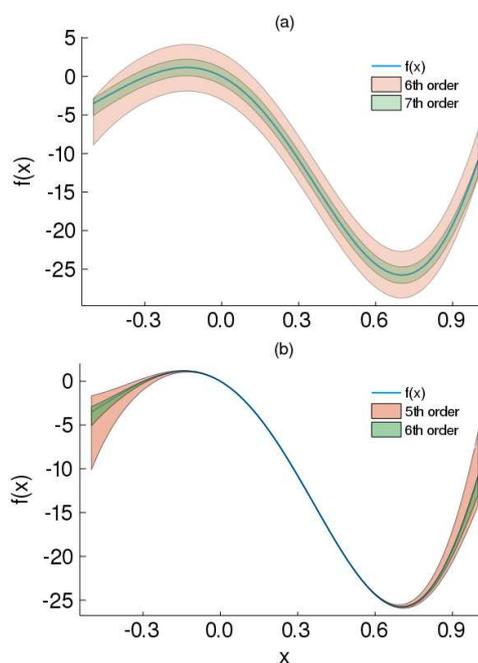


Figure 1: Examples of rigorous bounds for $f(x) = x(x - 1.1)(x + 2)(x + 2.2)(x + 2.5)(x + 3)\sin(1.7x + 0.5)$ in $D = \{x \mid -0.5 \leq x \leq 1.0\}$ using (a) absolute-remainder Taylor models of order 6 and 7, and (b) relative-remainder Taylor models of order 5 and 6.

form computations using extended precision formats. Figure 1 displays an example from Ref. [6] of a univariate function bounded by Taylor models with absolute or relative remainder. We shall describe examples of its use as well as its application to obtain validated solutions of ODEs. This work is built on other packages developed by us for interval arithmetic [8], Taylor series [2], and set-based

reachability [4].

Acknowledgment

LB acknowledges financial support from from the PAPIIT-UNAM project IG-IG100819. CS was supported in part by the Austrian Science Fund (FWF) under grants S11402-N23 (RiSE/SHiNE) and Z211-N23 (Wittgenstein Award) and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 754411.

References

- [1] L. Benet and D. P. Sanders. TaylorModels.jl: Rigorous function approximation with Taylor models in Julia. <https://github.com/JuliaIntervals/TaylorModels.jl>, 2019.
- [2] L. Benet and D. P. Sanders. TaylorSeries.jl: Taylor expansions in one and several variables in Julia. *Journal of Open Source Software*, 4(36):1043, 2019.
- [3] M. Berz. Differential algebraic description of beam dynamics to very high orders. *Part. Accel.*, 24:109–124, 1989.
- [4] S. Bogomolov, M. Forets, G. Frehse, K. Potomkin, and C. Schilling. JuliaReach: A toolbox for set-based reachability. In *HSCC*, pages 39–44, New York, NY, USA, 2019. ACM.
- [5] M. M. Joldes. *Rigorous Polynomial Approximations and Applications*. PhD Thesis, École Normale Supérieure de Lyon (ENS Lyon), 2011.
- [6] K. Makino. *Rigorous analysis of nonlinear motion in particle accelerators*. PhD Thesis, Michigan State University, 1998.
- [7] K. Makino and M. Berz. Taylor models and other validated functional inclusion methods. *International Journal of*

Pure and Applied Mathematics, 6(3):239–312, 2003.

- [8] D. P. Sanders and L. Benet. IntervalArithmetic.jl: Rigorous floating-point calculations using interval arithmetic in Julia. <https://github.com/JuliaIntervals/IntervalArithmetic.jl>, 2019.

Interval-Based Simulation of Zélus IVPs Using Dynlbex

Jason Brown¹ and François Pessaux*¹

¹U2IS, ENSTA Paris, Institut Polytechnique de Paris, 828 boulevard des Maréchaux, 91120 Palaiseau, France

Keywords: Dynlbex, Zélus, Compilation, Hybrid System, Interval, Guaranteed Integration, Simulation

Introduction

Hybrid systems are commonly defined as dynamical systems mixing discrete and continuous times. They are widely present in control command systems where a continuous physical process is controlled by software components which run at discrete instants. One of the verification techniques is to simulate the global system. In such a simulation process, the continuous physical process is modeled as differential equations whose solutions are approximated by dedicated integration algorithms. The discrete processing is the software components. Both parts of the system have to interact, allowing the discrete process to react to events of the continuous one.

Simulations can be very dependent on the initial conditions of the system. Small variations may have important impacts. Moreover, the initial conditions may not always be accurately known. A solution to address these uncertainties is to compute using intervals, hence to rely on interval-based guaranteed integration tools [2, 6].

Tools and Domain Specific Languages exist to ease the modeling, development and verification of hybrid systems (MODELICA, SIMULINK/STATEFLOW, LABVIEW, Zélus and others [4]). These languages provide numerous advantages compared to a manual implementation requiring to explicitly bind the code of the software components with the runtime/library of simulation. They often pro-

pose high-level constructs (automata, differential equations, guards) with dedicated static verifications (typechecking, initialization analysis, scheduling, causality analysis) and compile the hybrid model to low-level code (C, C++) to produce an executable simulation.

This work proposes to bind the flexibility of a hybrid programming language, Zélus[3], with the safety of interval-based guaranteed integration using Dynlbex[1, 5]. Zélus natively generates imperative OCaml code linked with a point-wise simulation runtime. Dynlbex is a plug-in of the C++ lbex library, bringing various validated numerical integration methods to solve Initial Value Problems (IVPs). We do not address the compilation of arbitrary Zélus programs toward Dynlbex. We present the compilation scheme for an IVP described in a subset of Zélus to a C++ simulation code using Dynlbex.

1 IVPs in Zélus

An IVP in Dynlbex is represented by a vector-valued ordinary differential equation (ODE) with initial conditions whereas an IVP in Zélus is represented by a system of coupled equations. Compilation from Zélus to Dynlbex therefore requires a transformation between these representations.

The model of a simple harmonic oscillator with dampening described by the equation $\ddot{x} + k_2 \dot{x} + k_1 x = 0$ with initial values $x(0) = 1, \dot{x}(0) = 0$ can be written in Zélus as :

```
let hybrid shm_decay (x0, x'0, k1, k2) = x
  where rec der x = x' init x0
  and der x' = -.k1 *. x -. k2 *. x' init
    x'0
```

*Corresponding author.

```
let hybrid main () = x where
  x = shm_decay (1.0, 0.0, 4.0, 0.4)
```

where **der** *x* represents \dot{x} and **der** *x'* is \ddot{x} . The node **main** instantiates the node **shm_decay** with specific initial values and k_1 and k_2 .

2 Compiling the IVP

Compiling the Zélus code requires two steps. First the hierarchy of nodes must be flattened, harvesting all the differential equations. During this process, each node instantiation expression is replaced by the body of the node where the occurrences of its parameters are replaced by the effective expressions provided at the instantiation point. This implies a recursive inlining mechanism which terminates since Zélus forbids recursive nodes.

Once the intermediate representation of the flattened system is obtained, the multiple equations have to be aggregated into a unique vector-valued function to finally generate the C++ code. Each differential equation corresponds to one dimension of the Dynlbex **Function** data structure. Initial conditions are also transformed in a vector-valued structure. During this process, Zélus expressions are compiled to C++ expressions. Since nodes are flattened, leading to a list of equations, this process mostly consists of a translation of arithmetic expressions into C++, mapping the identifiers to the appropriate vector component, and converting real constants into trivial intervals.

We extended the Zélus compiler to implement the described compilation process. This new backend operates on the intermediate representation obtained after type, causality and initialization analyses and does not interfere with the standard compilation. The code generated for the example given at the beginning of this section is shown in the following listing.

```
#define T0 (0.000000)
#define TEND (6.000000)
int main () {
  const int dim = 2;
```

```
Variable y(dim);
IntervalVector yinit(dim);
Function ydot =
  Function
  (y,
   Return
   (y[1],
    ((-Interval(4.000000)) * y[0]) -
     (Interval(0.400000) * y[1]))
  );
yinit[0] = Interval(1.000000);
yinit[1] = Interval(0.000000);
ivp_ode problem = ivp_ode(ydot, T0, yinit)
;
simulation simu =
  simulation(&problem, TEND, GL4, 1e-7);
simu.run_simulation();
simu.export_y0("export");
return 0;
}
```

In this generated code, the size of the *IVP* is 2 since we had 2 equations. The interval *y* stores the continuous state of the system. The vector *yinit* contains the initial values. Each equation is translated into an argument of the **Return** constructor. We can see that the compilation mapped the *x'* of the Zélus program to the dimension 1 of the vector-based representation, and *x* to the dimension 0. It is possible to recognize, in the **Return** clause, the translation of $-.k_1 * x - k_2 * x'$ where k_1 has been properly instantiated by 4.0 and k_2 by 0.4.

3 Experimental Results

The first experiment was to simulate the system with Zélus and with our generated code, then to compare the results. In the figure 1, the Zélus native simulation is represented by the red line and the simulation obtained using the intervals is shown by the green boxes.

Both simulations behave consistently. In particular, the results obtained with the standard integration runtime of Zélus always remain inside the boxes obtained using the interval mechanism. This suggests that the native integration runtime of Zélus is precise enough in this example to avoid inaccuracies that could be caused by float rounding errors.

Although there is not yet syntax extension of Zélus in the current implementation to spec-

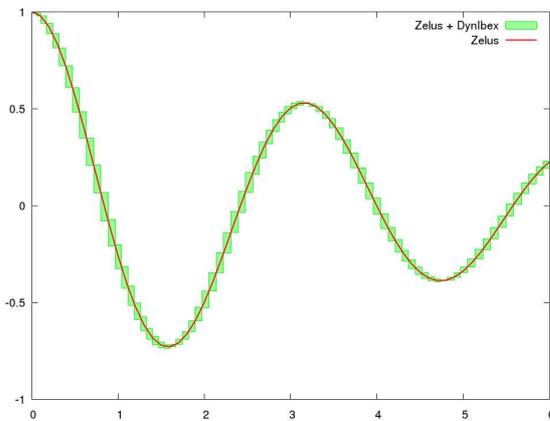


Figure 1: Simulations with/without intervals
 ify interval values, it is possible to add uncertainty on the initial value of `der x`, by manually changing the value of `yinit[0]` to `Interval(0.9, 1.0)` in the generated C++ code. The simulation obtained after this change is shown in the figure 2.

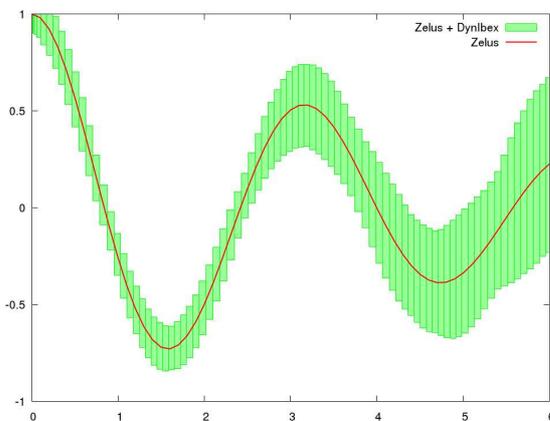


Figure 2: Simulation with initial uncertainty
 Both simulations continue to behave consistently, and we see more clearly how the uncertainty increases with time.

4 Conclusion

We presented a mechanism to compile *IVPs* described in *Zélu*s to C++ code using *Dynlbex*. This allows the simulation of programs written in a high-level programming language with interval-based validated numerical integration methods. This work has lead to a real implementation in the *Zélu*s compiler. Extensions

to handle more complex *IVPs* and to compile contracts verification on programs are in progress.

References

- [1] J. Alexandre dit Sandretto, A. Chapoutot, and O. Mullier. The dynlbex library. <http://perso.ensta-paristech.fr/~chapoutot/dynlbex/>.
- [2] O. Bouissou, S. Mimram, and A. Chapoutot. Hyson: Set-based simulation of hybrid systems. In *Proceedings of the 23rd IEEE International Symposium on Rapid System Prototyping, RSP 2012, Tampere, Finland, October 11-12, 2012*, pages 79–85, 2012.
- [3] T. Bourke and M. Pouzet. Zélu:s: A synchronous language with odes. In *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control, HSCC '13*, pages 113–118, New York, NY, USA, 2013. ACM.
- [4] L. P. Carloni, R. Passerone, A. Pinto, and A. L. Angiovanni-Vincentelli. Languages and tools for hybrid systems design. *Found. Trends Electron. Des. Autom.*, 1(1/2):1–193, Jan. 2006.
- [5] J. A. dit Sandretto and A. Chapoutot. Validated explicit and implicit Runge–Kutta methods. *Reliable Computing*, 22(1):79–103, Jul 2016.
- [6] T. A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi. Beyond hytech: Hybrid systems analysis using interval numerical methods. In N. Lynch and B. H. Krogh, editors, *Hybrid Systems: Computation and Control*, pages 130–144, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

Turnkey Solutions to PDEs in Exact Real Computation

Svetlana Selivanova* and Martin Ziegler

School of Computing, KAIST

Keywords: Exact Real Computation; Guaranteed Precision; Computable Analysis; PDEs; Difference Schemes; Analytic Solutions

$$\begin{cases} \mathbf{u}_t = L\mathbf{u} + f(t, x) \in C^p(\Omega, \mathbb{R}^n), \\ \mathbf{u}|_{t=0} = \varphi(x) \in C^q(\Omega, \mathbb{R}^n), \\ (\mathcal{L}\mathbf{u}|_{\partial\Omega \times [0, T]} = \psi(y)). \end{cases} \quad (1)$$

Introduction

We turn the rigorous but theoretical approach to computing with continuous data [8] into practice, complementing classical numerical and analytic methods for solving broad classes of initial-value and boundary-value problems (IVP and BVP) for partial differential equations (PDEs). The *Exact Real Computation* paradigm¹ allows to conveniently implement imperative algorithms involving real numbers, converging sequences, and smooth functions without the hassles of Turing machines. This approach differs from traditional Reliable Numerics in considering real numbers as exact entities (as opposed to intervals [5]) while guaranteeing output approximations up to error $1/2^n$ (as opposed to intermediate precision propagation), where n is the output error parameter. We develop a turnkey solver, including careful calculations of internal parameters (such as spatial grid and time step size) and in agreement with complexity predictions [4], in dependence on n . This is the starting point towards actual implementation.

Difference Schemes in Exact Real Computation

Consider IVP and BVP for systems of PDEs of the form

Here $\partial\Omega$ is the boundary of the compact set $\Omega \subset \mathbb{R}^m$, $x \in \Omega$, $y \in \partial\Omega \times [0, T]$, $L = \sum_{|\alpha| \leq s} A_\alpha(x, u) \frac{\partial^\alpha}{\partial x^\alpha}$. For a boundary-value problem (the Cauchy problem being stated without the last condition in the parentheses of (1)), \mathcal{L} is a linear operator.

Suppose the given IVP and BVP be *well posed* in that the classical solution $\bar{u} : [0; 1] \times \bar{\Omega} \rightarrow \mathbb{R}$ (i) exists, (ii) is unique, and (iii) depends continuously on φ . More precisely we assume that $u(t, x) \in C^2$ and its C^2 -norm is bounded linearly by C^2 -norms of the data as $\|u\|_{C^2} \leq c_u \|\varphi\|_{C^2}$ (in functional spaces guaranteeing all the required properties). Moreover suppose that the given IVP and BVP admit a (iv) stable (with stability coefficient c_{st}) and (v) approximating with at least the first order of accuracy (and approximation coefficient c_{app}) explicit *difference scheme* [2].

Then taking any (binary-rational) uniform space grid step h such that

$$h \leq 1/(c_u \cdot \|D_x^2 \varphi\| \cdot (1 + c_{st} \cdot c_{app}) \cdot 2^n)$$

and (binary-rational) time step τ meeting the Courant inequality $\tau \leq \nu h$, we can apply the standard (explicit) difference scheme iterations, treating all coefficients as exact reals. In this way we get an approximation to the solution with the precision $1/2^n$.

The coefficients c_u , c_{st} and c_{app} were explicitly expressed in [6] via (derivatives of) φ and $A_\alpha = A_\alpha^* = \text{const}$ for a particular difference scheme for symmetric hyperbolic systems.

*Corresponding author.

¹See arXiv:1608.05787v4

For the case when $A_\alpha(x, u) = A_\alpha(x)$ and $f(t, x) = 0$ it is possible to improve the bit cost of thus obtained algorithm by applying efficient matrix powering instead of step-by-step iterations [4].

Analytic PDEs in Exact Real Computation

For IVP with *analytic* A_α , f and φ in (1) we can rigorously compute solutions using analytic series, treating their coefficients as exact reals and applying iterations of [1], §4.6.3.

In the linear case $A_\alpha = A_\alpha(x)$, $f = 0$, it is more efficient to use the exponentiation series $\mathbf{u}(t, x) = \exp(tL)\varphi(x) = \sum_K t^K/K! \cdot L^K\varphi(x)$ and recursive operator powering, as suggested in [4]. More precisely, the n -th term of this power series gives approximation of the solution with precision $1/2^n$, provided that conditions of Theorem 8 of [4] hold.

Conclusion

For analytic PDEs we develop the series technique in addition to the possible application of difference schemes, because, as proved in [4] for the linear case (Theorem 3), it yields PTIME complexity bounds provided the input is PTIME computable. For the difference scheme approach the best complexity bound which we were so far able to establish for the linear case and PTIME inputs, was PSPACE (for particular examples $\#P^{\#P}$), i.e. much “worse” than PTIME.

Note also that finding the solution to the 2-dimensional Poisson equation was proved in [3] to be optimally in $\#P$ while solutions to Navier-Stokes equations were proved in [7] to be computable, but the proofs do not provide explicit algorithms.

Acknowledgement

Supported by the National Research Foundation of Korea (grant NRF-

2017R1E1A1A03071032), and by the Korean Ministry of Science and ICT (grant NRF-2016K1A3A7A03950702).

References

- [1] L. Evans. Partial differential equations. Graduate Studies in Mathematics, v. 19, American Mathematical Society, 1998.
- [2] S. Godunov and V. Ryaben’kii. *Difference Schemes: An Introduction to the Underlying Theory*. Elsevier Science Ltd, 1987.
- [3] A. Kawamura, F. Steinberg, and M. Ziegler. On the computational complexity of the Dirichlet problem for Poisson’s equation. *Mathematical Structures in Computer Science* 27:8, P. 1437–1465, 2017.
- [4] I. Koswara, S. Selivanova, and M. Ziegler. Computational complexity of real powering and improved solving linear differential equations. In *Proc. 14th International Computer Science Symposium in Russia*, volume 11532 of *LNCS*, 2019.
- [5] M. Plum. Computer-assisted proofs for semilinear elliptic boundary value problems. *Japan J. Indust. Appl. Math.*, V. 26, No 2-3, P. 419-442., 2009.
- [6] S. Selivanova and V. Selivanov. Computing solution operators of boundary-value problems for some linear hyperbolic systems of PDEs. *Logical Methods in Computer Science*, 13 (4), 2017.
- [7] S. M. Sun, N. Zhong, and M. Ziegler. On computability of Navier-Stokes’ equation. volume 9136 of *LNCS*, 2015.
- [8] K. Weihrauch. *Computable analysis*. Springer, Berlin, 2000.

Guaranteed interval integration for large initial boxes

Julien Damers¹, Luc Jaulin¹, and Simon Rohou¹

¹ENSTA-Bretagne, Lab-STICC UMR CNR 6285

Keywords: Interval analysis, Guaranteed integration, Lie groups, Symmetries

Introduction

A Lie group [4] is both an abstract and a smooth n-dimensional manifold so that multiplication and the inverse are both smooth. Lie groups have been introduced to model the continuous symmetries of differential equations. They are widely used for their resolution. The main idea of the new approach presented in this paper is to take advantage of the symmetries of the problem to extend one solution to get all other solutions.

Our main contribution is to show that the use of Lie symmetries can be combined with interval based methods to propagate uncertainties through differential equations [1]. More precisely, we will compute an enclosure of the solution to a differential equation assuming that the initial state is inside a box which may be large. The proposed method will be compared to existing methods such as CAPD [5] or DynIbex [2]. Some test-cases related to robotic applications illustrate the efficiency of our approach.

Problem

Most dynamical systems can be represented with a state equation such as $\dot{x} = f(x)$. When it is not possible to find an analytic solution for our equation we need an integration scheme to determine our solution. Conventional guaranteed interval integration libraries such as DynIbex or CAPD are able to compute a guaranteed solution without requiring much computing time when the initial condition is well described. However for some spe-

cific cases, for instance when attractors exist near the trajectory, these methods may have some difficulties. In many fields, for instance in robotics, the precision we have on our initial condition can be limited, thus the need of a tool for parameter estimation with larger initial boxes [3].

Our approach

Similarly to the variation of parameter method of Laplace, where one looks for a general solution knowing a particular solution of the differential equation, our method consists in first calculating precisely a trajectory using an integration tool such as CAPD or DynIbex applied with a given precise point as initial condition. We call this trajectory the *reference* (in red on Figure 1). Using the symmetries of the problem, we are able to calculate the solution for a different initial condition at a given time. We are able to do so without the need of applying conventional integration techniques i.e calculating each step, hence saving computation time.

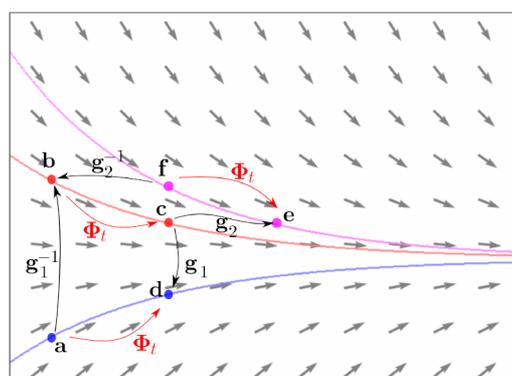


Figure 1: Integration principle illustration

Results

Consider the system following the equations below:

$$\begin{cases} \dot{x}_1 = -x_1^3 - x_1x_2^2 + x_1 - x_2 \\ \dot{x}_2 = -x_2^3 - x_1^2x_2 + x_1 + x_2 \end{cases}$$

and an initial condition $\mathbf{a}_0 = \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}$. We use a conventional guaranteed integration library to compute the trajectory associated with this system. Then we compare our approach to the result given by CAPD for an initial condition

$$[\mathbf{a}_0] = \begin{bmatrix} [0.4, 0.6] \\ [-0.1, 0.1] \end{bmatrix}$$

represented in red in Figure 2 and Figure 3.

As shown in Figure 3 we observe a bloating effect when using conventional interval integration method, in this case CAPD, the computation stops after a time $t = 0.76s$ because of the bloating effect. The approach presented in this paper is robust to the increase of the interval size as initial condition and is able to carry out the integration until the end set (here 7 seconds).

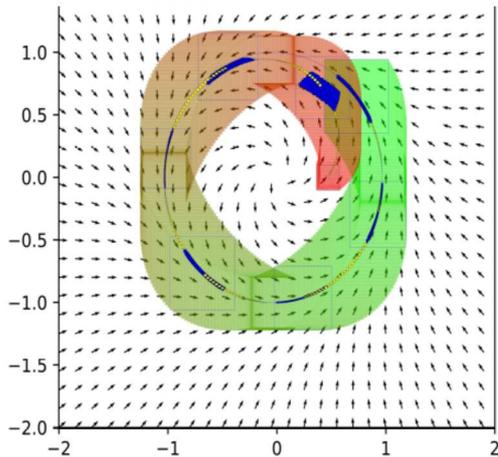


Figure 2: Integration using Lie symmetries

We also carried out the integration with 1000 particles picked randomly from the initial box $[\mathbf{a}_0]$ and calculated their positions at each second up to 7 seconds. The results obtained for both CAPD and our method is

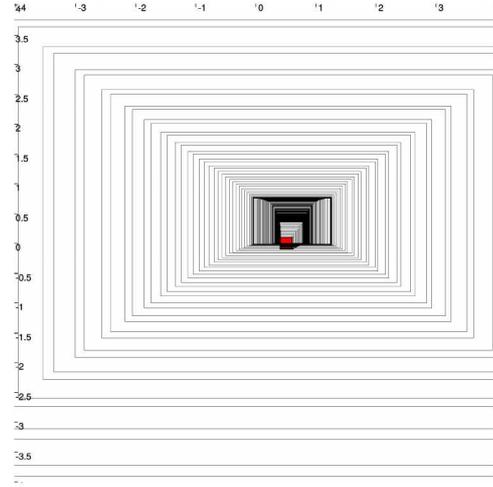


Figure 3: Integration using CAPD only

represented in blue on Figure 2. As we are only computing on points as initial condition it is possible to use CAPD for our particle cloud. We compared the computing time and as expected our approach achieves much better results on the computation time aspect compared to CAPD. It takes **297ms** with our method and **92811ms** using CAPD on an Intel Core I7-5700HQ @ 2.7GHz and 16GB of ram . This was expected as we don't need to calculate each step of the integration process but only the transformation for time step(s) enclosing our desired time slot.

Acknowledgement

This work has been supported by the *French Government Defense procurement and technology agency* (DGA). It also benefited from the support of the project CONTREDO of the French National Research Agency (ANR).

References

- [1] J. Alexandre dit Sandretto. Contraction, propagation and bisection on a validated simulation of ode. Lyon, France, 2016.
- [2] J. A. dit Sandretto and A. Chapoutot. Validated explicit and implicit Runge–Kutta

methods. *Reliable Computing*, 22(1):79–103, Jul 2016.

- [3] T. Raissi, N. Ramdani, and Y. Candau. Set membership state and parameter estimation for systems described by nonlinear differential equations. *Automatica*, 40:1771–1777, 2004.
- [4] J. Serre. *Lie Algebras and Lie Groups*. Springer, 1965.
- [5] D. Wilczak, P. Zgliczynski, P. Pilarczyk, M. Mrozek, T. Kapela, Z. Galias, J. Cyranka, and M. Capinsky. Computer assisted proofs in dynamics group, a c++ package for rigorous numerics, 2017.

Validation of a controller under state constraints

Joris Tillet¹, Luc Jaulin¹, and Fabrice Le Bars¹

¹Lab-STICC UMR CNRS 6285, ENSTA Bretagne

Keywords: Trailer control; Interval analysis; Constraints validation.

Introduction

In 1512, the boat named *La Cordelière* sunk in the *Rade de Brest*. Its wreck is still there, on the seabed or more probably under several meters of sediments. As wrecks of this time are rather rare, and the research area is huge, it leads to an interesting challenge.

Since the wreck is buried under the sand, the only sensor which is likely to detect the wreck is a magnetometer, by sensing the magnetic field perturbations of the anchors. Therefore, in order to hope to find the researched boat, a magnetometer should be dragged near to seabed in all the area of research. This is a very long mission and it is intricate to be sure that the sensor is gone everywhere.

This is why *Boatbot* was developed. *Boatbot* is a semi-rigid inflatable boat on which an electric motor was added behind the steering wheel. So, the boat can be regulated in head. Based on this robot, the objective was to develop some algorithms of control such that the magnetometer dragged by the boat properly follows the desired trajectories, while guaranteeing that some constraints are always respected.

Finding a controller

To be sure that the cable cannot be cut by the propellers of the boat, the idea was to put a kayak between the boat and the magnetometer. In this way, near to the propellers there is only a rope which stays at the surface of the water and can even at worst be cut

without losing the magnetometer. This experiment can be seen on the picture of *Boatbot* presented in Figure 1.

So the objective here is to find a controller which can control the position of the magnetometer by acting only on the direction of the boat. To address this problem, a good approach is to consider a car with a trailer, and to try to control the trailer. The trailer should follow a vector field, for instance Van der Pol vector field, like in Figure 2.

A robot is represented by its state vector \mathbf{X}

$$\mathbf{X} = \begin{pmatrix} x \\ y \\ \theta \\ \theta_r \end{pmatrix},$$

and its evolution function \mathbf{f}

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}, u) = \begin{cases} \cos(\theta) \\ \sin(\theta) \\ u \\ \frac{1}{L_r} \sin(\theta - \theta_r) \end{cases}.$$

The couple of variables (x, y) represents the position of the car, θ its head and θ_r the head of the trailer. L_r is the distance between the car and the trailer (see Figure 3).

It means that a robot is a dynamic system which is modeled by a differential equation. And the job of the controller is to find the input u of the system with respect to some measurements \mathbf{Y} [3].

The error the controller should canceled is the difference between the course of the trailer and the direction given by the vector field [7]. Using feedback linearization method, a controller can be quite easily found and makes the error converge toward zero in few seconds [6].



Figure 1: Picture of *Boatbot* searching for *La Cordelière*.

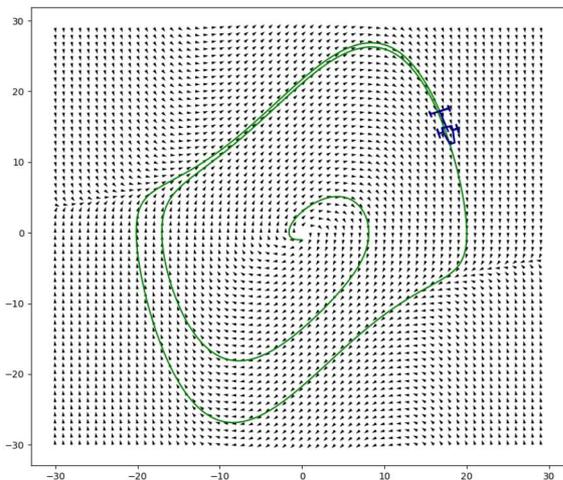


Figure 2: Simulation of a car with a trailer, where the trailer follows the Van der Pol vector field.

State constraint

Henceforth, the aim is to guarantee that some constraints are always respected. For instance, the angle between the car and the trailer should stay little to ensure that the trailer will never collide with the car itself. So here is presented a mean to show where some constraints are validate or not, supposing the controller works fine. This method relies only

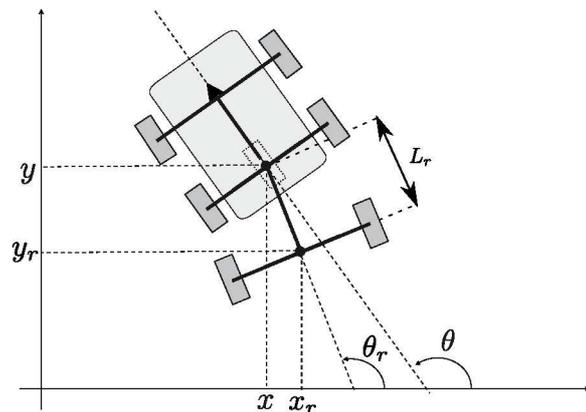


Figure 3: Model of the car with a trailer.

on the knowledge of the vector field followed by the robot, using Lie derivatives. This is why there is no need here to integrate any differential equation, like in [1, 2, 9, 10].

As the controller presented hereinabove is supposed to be perfect, we know that the robot will exactly follow the vector field, wherever it is. So the method consists in computing from any position in the vector field the theoretical state of the robot with respect to the controller, and deducing whether the constraint is respected. Interval analysis [8] is used here, helping to find the separator for a given constraint, and to use the SIVIA algo-

rithm [5] to validate the controller in a specific location [4].

An example of result is given in Figure 4. The simulation plots red circles (●) when the constraint is not respected (when there are tight curves), and its trajectory is superposed on the result of the SIVIA: we are sure that the constraint is respected when the robot is in green background areas (■), and violated in orange ones (■). The boundary is in gray, and inside we cannot be sure whether the constraint is respected (■).

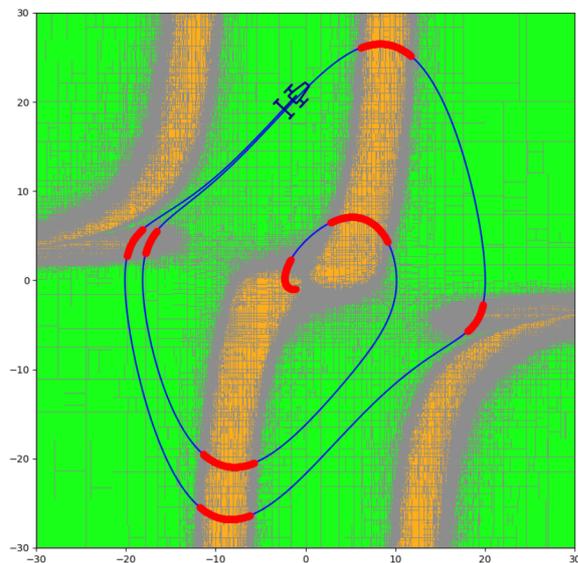


Figure 4: Example of a simulation of the car with a trailer superposed on the SIVIA result. It is certain that the constraint is respected everywhere the background is green (■) supposing the trailer follows the vector field.

Acknowledgement

This work has been supported by the *French Government Defense procurement and technology agency* (DGA). It also benefited from the support of the project CONTREDO of the French National Research Agency (ANR).

References

- [1] J. Cruz and P. Barahona. Constraint reasoning with differential equations. *Applied Numerical Analysis & Computational Mathematics*, 1:140 – 154, 03 2004.
- [2] M. Janssen, P. V. Hentenryck, and Y. Deville. A constraint satisfaction approach for enclosing solutions to parametric ordinary differential equations. *SIAM Journal on Numerical Analysis*, 40(5):1896–1939, 2002.
- [3] L. Jaulin. *Mobile robotics*. ISTE WILEY, 2015.
- [4] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag London, 2001.
- [5] L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993.
- [6] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.
- [7] M. Le Gallic, J. Tillet, F. Le Bars, and L. Jaulin. Tight Slalom Control for Sailboat Robots. In S. M. Schillai and N. Townsend, editors, *Robotic Sailing 2018*, pages 87–93, 2018.
- [8] R. E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs N. J., 1966.
- [9] S. Rohou, L. Jaulin, M. Mihaylova, F. L. Bars, and S. Veres. Guaranteed Computation of Robots Trajectories. *Robotics and Autonomous Systems*, 93:76–84, 2017.
- [10] J. A. D. Sandretto and A. Chapoutot. Dynbex: a differential constraint library for studying dynamical systems. *Conference on Hybrid Systems: Computation and Control (HSCC2016)*, 2016.

Efficient computation of the set of stabilizing controllers for an LTI System using intervals

Alexandre Lefort¹

¹ENSTA Bretagne / Sirehna, Naval Group

Keywords: Robust Control; Intervals; Stability

Introduction

Guaranteed characterisation of the set of controllers stabilizing a system is a major problem in control theory. Interval analysis gives a tool to solve this problem as described in [2]. However it is computationally expensive for high order systems (>7) with a lot of controller gains (>5) or parametric uncertainties. This work deals with author attempts on alternative approaches to improve this computation efficiency. Section 2 recalls stability criteria for systems and how they are used by an interval analysis algorithm in robust control. Section 3 gives some alternative implementations of stability criteria. Section 4 suggests a different algorithm.

Stabilizing Controllers Set Computation

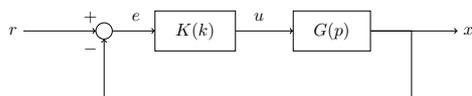


Figure 1: Closed-Loop System F

Let $G(p)$ and $K(k)$ be Linear Time Invariant Systems (LTI). G is called the regulated systems, p are the uncertain parameters of G . K is called the controller and k are the controller gains. $G(p)$ and $K(k)$ are linked in a closed-loop system $F(p, k)$ as shown in Fig.

1. $F(p, k)$ is also an LTI system. The problem stated in this work is to find the set \mathcal{K}_{stable} stabilizing F for all values of p inside a given set \mathcal{P} .

Given a state-space representation of F , $(A_F(p, k), B_F, C_F, D_F)$, An internal stability criterion for F is given by the Routh-Hurwitz criterion [2] : Given $P(p, k, s) = \det(sI - A_F(p, k))$ the characteristic polynomial of A_F and $a_i(p, k) \in \mathcal{R}^n$ its coefficients and H the Hurwitz Matrix given by:

$$H = \begin{bmatrix} a_1 & a_0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ a_3 & a_2 & a_1 & a_0 & 0 & 0 & \cdots & 0 \\ a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & a_n \end{bmatrix}$$

F is internally stable iff all the minors of H are strictly negative.

As those minors have an analytic expression, this criterion is available for a set computation via interval analysis. [2] translates the stabilizing set finding problem as a constraint satisfaction problem (CSP) and provides an algorithm to solve it by operating dichotomies on an initial box of values of k . However the complexity of this algorithm is exponential with the dimension of the interval box k and the evaluation pessimism increases dramatically with the order n of the system F for a naive implementation of the Routh-Hurwitz stability criterion.

Alternative Stability Criteria

Following the previous statement, several solutions are explored to control the computational complexity.

The first improvement is to use the Lienard-Chipart criterion [4] which is a direct derivative from the Routh-Hurwitz. However it is more efficient as it tests only half the minors of the Hurwitz matrix, giving the opportunity to not compute the minor with the highest degree, which suffer the most from evaluation pessimism. For the same computational complexity, it is possible to deal with systems with one more order.

The second tested solution is to improve the Directed Acyclic Graph (DAG) of the stability criterion expression for a more precise interval evaluation. For that, the operator $HurwStab([a_i])$ is created at a low implementation level for interval computation. This results in a significant reduction in evaluation pessimism.

The last one is to test an alternative criterion based on the Argument Principle [5] formula: given a complex function f and a complex positively oriented contour \oint_C where f never equals zero,

$$\oint_C \frac{f'(z)}{f(z)} dz = 2\pi i(Z - Q)$$

where Z and Q are respectively the number of zeros and poles of f . The idea is to replace f by the characteristic polynomial (which does not have poles) to test if it has roots on the right half plane which cause instability. It is possible with a clever contour like on Fig. 2, with a maximum radius fixed with Gershgorin circles. It provides an alternative criterion as long as the algorithm can compute integrals with interval analysis.

Despite the addition of complexity and pessimism caused by integral computation, this alternative criterion seems interesting for some problems with high order systems.

Alternative Set Computation Algorithm

Eventually, an alternative algorithm is suggested to compute the stabilizing set. It uses

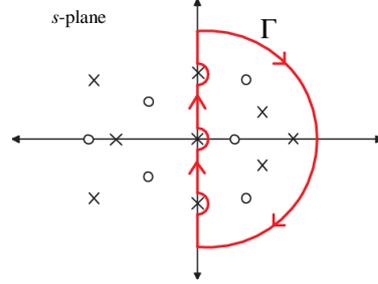


Figure 2: A Nyquist Contour Γ

Kharitonov theorem [3]. Kharitonov states that, for a characteristic polynomial with interval coefficients, it is sufficient to test only four polynomial edges to prove complete stability of all polynomials in the set.

As far as the author knows, there is no method to prove complete instability for such interval polynomials. However, Dabbene [1] gives a fast randomized algorithm to find a stable punctual polynomial inside interval polynomials. A failure of the Dabbene algorithm suggests a complete instability of the set. For a polynomial order < 14 , about 1000 iterations seem to provide a reliable result.

Here, Kharitonov and Dabbene are seen as operators taking interval polynomial coefficients and returning a Boolean (Kharitonov is true if the set is stable, Dabbene is true if it found a stable point in the set). Based on definitions given in previous sections, the author defines two operators:

- The coefficient operator :

$$\begin{aligned} (p, k) &\rightarrow a_i(p, k) \\ p \in \mathcal{P}, k \in \mathcal{K}, a_i \in \mathcal{R}^n \end{aligned} \quad (1)$$

- The Dabbene-Kharitonov (DK)-operator :

$$[a_i] \rightarrow \begin{cases} true & \text{if } Kharitonov([a_i]) \\ false & \text{if } \neg Dabbene([a_i]) \\ unknown & \text{if } Dabbene([a_i]) \end{cases} \quad (2)$$

Using those operators, the alternative Set Computation algorithm steps are as follows:

1. With a given set of values for (p,k) the

coefficient operator provides a set of polynomial coefficients $[a_i]$.

2. The DK-operator is used by a paver to provide the set of stable polynomials given by their coefficients.
3. Based on the result of 2., a Set Inversion Algorithm returns the stabilizing controller gains set \mathcal{K}_{stable} .

This algorithm is not guaranteed as the Dabbene test is not. However still it seems relevant because a failure of the Dabbene test is unlikely to occur as it is explained in the statistical analysis provided in [1]. This algorithm could be efficient insofar as the KD-operator does not introduce evaluation pessimism. It is not the case for the coefficient operator but its expression is assumed to be simple regarding Routh-Hurwitz criterion expression.

A discussion on the complete implementation of this algorithm will conclude the work.

References

- [1] F. Dabbene, B. Polyak, and R. Tempo. On the complete instability of interval polynomials. *Systems Control Letters*, 56:431–438, 06 2007.
- [2] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied interval analysis*, volume 1. Springer Science & Business Media, 2001.
- [3] V. L. Kharitonov. The asymptotic stability of the equilibrium state of a family of systems of linear differential equations. *Differentsial'nye Uravneniya*, 14(11):2086–2088, 1978.
- [4] A. Liénard and M. Chipart. Sur le signe de la partie réelle des racines d'une équation algébrique. *J. Math. Pures Appl*, 10(4):291–346, 1914.
- [5] W. Rudin. *Real and complex analysis*. Tata McGraw-hill education, 2006.

Analyzing Uncertain Dynamical Systems After State-Space Transformations Into Cooperative Forms

Julia Kersten^{*1}, Andreas Rauh¹, and Harald Aschemann¹

¹ Chair of Mechatronics, University of Rostock, Justus-von-Liebig-Weg 6
D-18059 Rostock, Germany,

{Julia.Kersten, Andreas.Rauh, Harald.Aschemann}@uni-rostock.de

Keywords: Robust Control; LMIs; State-Space Transformation; Cooperativity

Introduction

There are different reasons for the occurrence of uncertainty. It can appear due to model simplifications, approximation of nonlinearities, imprecise parameter knowledge and/or order reduction as well as physical and numerical restrictions of the system itself. Uncertainty caused by measurement noise and sensor inaccuracies are further examples. In any case, uncertainties can be treated either stochastically or as bounded quantities in terms of worst case scenarios, where the lower and upper bounds are summarized in an interval. Hence, interval arithmetic is a common tool, see [3]. Unfortunately, its use tends to lead to overestimation due to the so-called wrapping effect. To avoid this, cooperativity has already been investigated in several papers, [2, 4, 5]. A system is cooperative, if for an autonomous dynamic system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \quad , \quad \mathbf{x} \in \mathbb{R}^n \quad , \quad (1)$$

all off-diagonal elements $J_{i,j}$, $i, j \in \{1, \dots, n\}$, $i \neq j$, of the corresponding Jacobian

$$\mathbf{J} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \quad (2)$$

are strictly non-negative according to

$$J_{i,j} \geq 0 \quad , \quad i, j \in \{1, \dots, n\} \quad , \quad i \neq j \quad . \quad (3)$$

^{*}Corresponding author.

This means, that state trajectories $\mathbf{x}(t)$ starting in the positive orthant $\mathbb{R}_+^n = \{\mathbf{x} \in \mathbb{R}^n \mid x_i \geq 0, \forall i \in \{1, \dots, n\}\}$ are guaranteed to stay in this positive orthant for all $t \geq 0$ because $\dot{x}_i(t) = f_i(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \geq 0$ holds for all components $i \in \{1, \dots, n\}$ of the state vector as soon as the state variable x_i reaches the value $x_i = 0$. The advantage of cooperativity is the simplification of several tasks such as the computation of guaranteed state enclosures, the design of interval observers, forecasting worst-case bounds for selected system outputs in predictive control and the identification of unknown parameters.

Main Idea

Many system models in biological, chemical, and medical applications are naturally cooperative. However, there is also a great number of systems (typically from the fields of electric, magnetic, and mechanical applications) which do not show this property if the state equations are derived using first-principle techniques. Hence, it is often desired to transform such system models into an equivalent cooperative form. If a system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (4)$$

is linear, it can be given in the state-space representation

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{p}) \cdot \mathbf{x} + \mathbf{B}(\mathbf{p}) \cdot \mathbf{u} \quad (5)$$

with the state vector \mathbf{x} and the input \mathbf{u} considering parameter uncertainty in the elements of the system matrix $\mathbf{A}(\mathbf{p})$ as well as the input

matrix $\mathbf{B}(\mathbf{p})$. Moreover, most nonlinear systems can be reformulated into a quasi-linear state-space representation

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}) \cdot \mathbf{x} + \mathbf{B}(\mathbf{x}) \cdot \mathbf{u} , \quad (6)$$

where the uncertainty lies in the state dependencies due to nonlinear expressions in the right-hand sides of (4). Both representations (5) and (6) describe uncertain systems, which can be transformed into cooperative forms by means of Eqs. (1)–(3). If the system model is controllable (or at least stabilizable) and the desired operating state is set to $\mathbf{x} = \mathbf{x}_s = \mathbf{0}$ without loss of generality for the steady-state input $\mathbf{u} = \mathbf{u}_s = \mathbf{0}$, a feedback controller is introduced in Eqs. (5) and (6) according to $\mathbf{u} = -\mathbf{K}(\mathbf{p}) \cdot \mathbf{x}$ or $\mathbf{u} = -\mathbf{K}(\mathbf{x}) \cdot \mathbf{x}$, respectively, leading to the following state-space representations:

$$\dot{\mathbf{x}} = (\mathbf{A}(\mathbf{p}) - \mathbf{B}(\mathbf{p})\mathbf{K}(\mathbf{p})) \cdot \mathbf{x} = \mathbf{A}_C(\mathbf{p}) \cdot \mathbf{x} \quad (7)$$

$$\dot{\mathbf{x}} = (\mathbf{A}(\mathbf{x}) - \mathbf{B}(\mathbf{x})\mathbf{K}(\mathbf{x})) \cdot \mathbf{x} = \mathbf{A}_C(\mathbf{x}) \cdot \mathbf{x}. \quad (8)$$

For the transformation into an equivalent cooperative form, we make use of a method developed in [2] for linear systems with crisp parameterization. This approach has been extended to uncertain systems in [5] and generalized in [4] to cover real-life applications in an efficient manner. It was shown that one needs to distinguish between systems with purely real and conjugate complex eigenvalues. For the presented paper, we will concentrate on the former. It was assumed that the uncertain system matrix can be expressed by the element-wise defined inequality

$$\mathbf{Z}_a - \mathbf{\Delta} \leq \mathbf{Z} := \mathbf{A}_C \leq \mathbf{Z}_a + \mathbf{\Delta} , \quad (9)$$

where $\mathbf{\Delta}$ consists of the (symmetric) worst-case bounds of all entries in $[\mathbf{A}_C]$. Note, the midpoint matrix $\mathbf{Z}_a = \mathbf{Z}_a^T$ in Eq. (9) is assumed to be symmetric in what follows. A Metzler matrix $\mathbf{R} = \mu \mathbf{E}_n - \mathbf{\Gamma}$ is searched for, which has the same eigenvalues as \mathbf{Z}_a , with a constant $\mu \in \mathbb{R}$ and a diagonal matrix $\mathbf{\Gamma} \in \mathbb{R}^{n \times n}$; $\mathbf{E}_n \in \mathbb{R}^{n \times n}$ is a matrix with

all elements equal to 1 and $\mathbf{\Gamma} = \rho \mathbf{I}_n$ with $\rho > \mu$ and the identity matrix \mathbf{I} of order n . If $\text{eig}(\mathbf{R}) = \text{eig}(\mathbf{Z}_a)$, according to [2], there exists an orthogonal matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ such that $\mathbf{S}^T \mathbf{Z} \mathbf{S}$, respectively, is Metzler provided that $\mu > n \|\mathbf{\Delta}\|_{\max}$, where $\|\mathbf{\Delta}\|_{\max}$ denotes the maximum absolute value of $\mathbf{\Delta}$. However, in several practical cases finding the transformation matrix \mathbf{S} is not trivial. Thus, this approach was converted into a computationally feasible optimization problem formulated with linear matrix inequality (LMI) constraints [1]. This is done with the main goal of a generalization to cover both possible uncertainties of Eqs. (5) and (6). Both types of system models with time- and state-dependent parameter uncertainties are investigated for real-life electric RLC-circuits.

References

- [1] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, Philadelphia, 1994.
- [2] D. Efimov, T. Raïssi, S. Chebotarev, and A. Zolghadri. Interval State Observer for Nonlinear Time Varying Systems. *Automatica*, 49(1):200–205, 2013.
- [3] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter. *Applied Interval Analysis*. Springer-Verlag, London, 2001.
- [4] J. Kersten, A. Rauh, and H. Aschemann. State-Space Transformations of Uncertain Systems With Purely Real and Conjugate-Complex Eigenvalues Into a Cooperative Form. In *Proc. of 23rd International Conference on Methods and Models in Automation and Robotics 2018*, Miedzyzdroje, Poland, 2018.
- [5] T. Raïssi, D. Efimov, and A. Zolghadri. Interval State Estimation for a Class of Nonlinear Systems. *IEEE Trans. Automat. Contr.*, 57:260–265, 2012.

Guaranteed Polynesian Navigation

Thibaut Nico¹, Luc Jaulin ^{*2}, and Benoît Zerr²

¹ECA, Lab-STICC UMR CNR 6285

²ENSTA-Bretagne, Lab-STICC UMR CNRS 6285

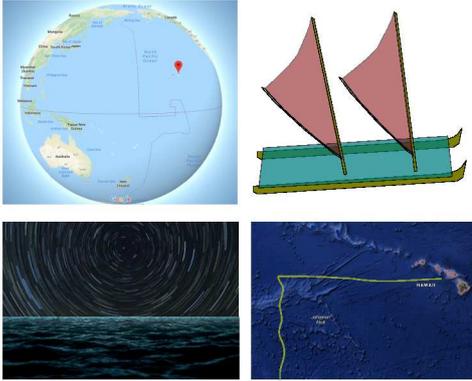


Figure 1: Polynesian navigation

Keywords: Navigation; Intervals, Contractors, No-lost zone

Introduction

The Polynesian navigation problem asks to move from islands to other islands without being lost. The navigation should be performed without GPS, compass and clocks. The difficulty of the navigation is illustrated by Figure 1: the ocean is huge, the islands are small, the boats have a dynamic which is more or less uncertain.

Among the techniques used by Polynesians, the observation of the stars (see Figure 2) are useful to get the heading, but also to detect if the boat is on the route which leads to the desired island. The approach we will follow to guarantee that we can reach an island from another island, uses guaranteed integration [7], tube programming [6], constraint programming [8], localization [3], contractors [2] and interval analysis [5].

^{*}Corresponding author.

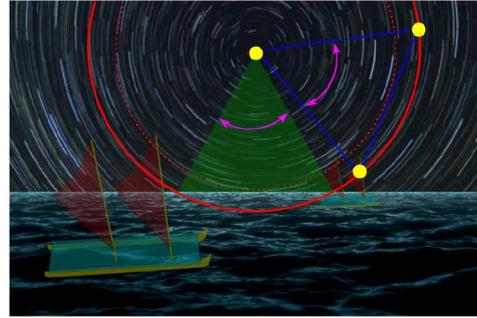


Figure 2: Pair of stars technique: the boat is on the right route if the bottom star rises when the right star sets

Formalisation

Given a set of geo-localized islands $\mathbf{m}_i, i \geq 0$, the i th coastal area is:

$$\mathbb{C}_i = \{\mathbf{x} \mid c_i(\mathbf{x}) \leq 0\}.$$

A robot has to move in this environment without being lost. Figure 3 represents a set of 4 islands with the associated coastal zones $\mathbb{C}_1, \mathbb{C}_2, \mathbb{C}_3, \mathbb{C}_4$ (painted blue).

We assume that (i) the coastal areas are small compare to the offshore area, (ii) in the coastal area, the robot knows its state, (iii) offshore, the robot is blind and has an open loop strategy, such as for instance go North and (iv) the robot is described by blind state equations

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), & \mathbf{u}(\cdot) \in [\mathbf{u}](t) \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases}$$

where the input $\mathbf{u}(t)$ belongs to the uncertainty box $[\mathbf{u}](t)$.

We define the *set flow* $\Phi : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathcal{P}(\mathbb{R})$ as:

$$\Phi(t_1, \mathbf{x}_0) = \left\{ \begin{array}{l} \mathbf{a} \mid \exists \mathbf{u}(\cdot) \in [\mathbf{u}](t), \mathbf{a} = \mathbf{x}(t_1), \\ \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \mathbf{x}(0) = \mathbf{x}_0 \end{array} \right\}$$

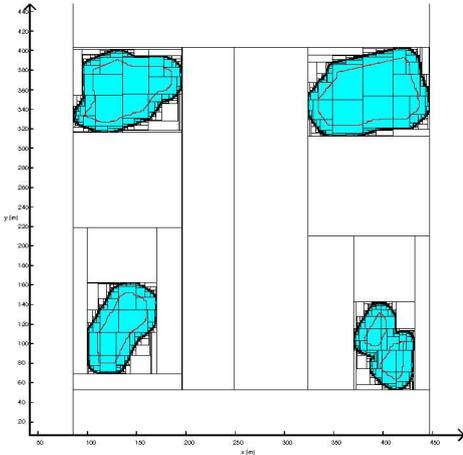


Figure 3: Islands and coastal zones

Given the set \mathbb{A} (for instance a coastal area), the *backward reach set* [1] is defined by

$$\text{Back}(\mathbb{A}) = \left\{ \mathbf{x} \mid \forall \varphi \in \Phi, \exists t \geq 0, \varphi(t, \mathbf{x}) \in \mathbb{A} \right\}$$

Interval analysis is often used to compute backward reach sets in the case where the robot is nonlinear [4]. We have

$$\text{Back}(\mathbb{A} \cup \mathbb{B}) \supset \text{Back}(\mathbb{A}) \cup \text{Back}(\mathbb{B}) .$$

This is the *Archipelago effect* which tells us that finding an Archipelago ($\mathbb{A} \cup \mathbb{B}$) is easier than finding individual islands.

Moving between coastal zones

Assume that we have m coastal sets $\mathbb{C}_1, \mathbb{C}_2, \dots, i \in \{1, 2, \dots\}$ and open loop control strategies $\mathbf{u}_j, j \in \{1, 2, \dots\}$ or equivalently, we have set flows $\Phi_j(t, \mathbf{x}_0)$. Moreover, we assume that the control strategy cannot change offshore. As a consequence, (i) from \mathbb{C}_1 we can reach \mathbb{C}_2 with the j th control strategy if $\mathbb{C}_1 \cap \text{Back}(j, \mathbb{C}_2) \neq \emptyset$. (ii) From \mathbb{C}_1 we can reach \mathbb{C}_2 with at least one control strategy if $\mathbb{C}_1 \cap \bigcup_j \text{Back}(j, \mathbb{C}_2) \neq \emptyset$. (iii) From \mathbb{C}_1 we can reach $\mathbb{C}_2 \cup \mathbb{C}_3$ with at least one control strategy if $\mathbb{C}_1 \cap \bigcup_j \text{Back}(j, \mathbb{C}_2 \cup \mathbb{C}_3) \neq \emptyset$.

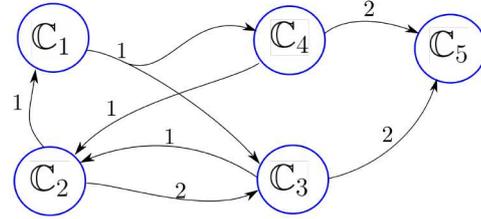


Figure 4: Reach an island from another island using a 'Go-East' strategy

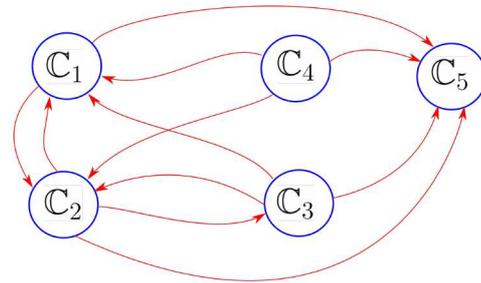


Figure 5: Reachability graph

Therefore, we define the *reachability* relation \hookrightarrow as:

- $\mathbb{C}_a \hookrightarrow \mathbb{C}_b$ if from \mathbb{C}_a we can reach \mathbb{C}_b with at least one control strategy j .
- \hookrightarrow is the smallest transitive relation which satisfies

$$\begin{cases} \forall k \in \mathbb{K}, \mathbb{C}_{i_k} \hookrightarrow \mathbb{C}_b \\ \exists j, \mathbb{C}_a \cap \text{Back}(j, \bigcup_{k \in \mathbb{K}} \mathbb{C}_{i_k}) \neq \emptyset \\ \Rightarrow \mathbb{C}_a \hookrightarrow \mathbb{C}_b \end{cases}$$

Consider for instance, the hyper-graph of Figure 4 where the relation $\mathbb{A} \xrightarrow{j} \mathbb{B}, \mathbb{C}$ means that from \mathbb{A} the robot can reach either \mathbb{B} or \mathbb{C} using the j th strategy. For instance, in our graph

$$\mathbb{C}_1 \cap \text{Back}(1, \mathbb{C}_3 \cup \mathbb{C}_4) \neq \emptyset \Rightarrow \mathbb{C}_1 \xrightarrow{1} (\mathbb{C}_3, \mathbb{C}_4)$$

Thus, the associated reachability graph (corresponding to \hookrightarrow) is given by Figure 5.

In a similar way, we can also define the forward reach set.

No lost zone

We define the no-lost zone as the set \mathbb{S} of all states that we may visit from a coastal area

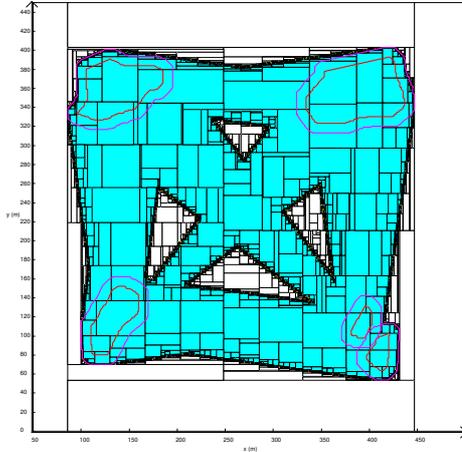


Figure 6: No-lost zone associated with the 5 islands

without being lost with the available control strategies. Define the *index set* associated with the strategy \mathcal{I}_j as

$$\mathcal{I}_j = \{k | \mathbb{C}_k \cap \text{Back}(j, \bigcup_{i \neq k} \mathbb{C}_i) \neq \emptyset\}.$$

If we start from $\mathbb{C}_k, k \in \mathcal{I}_j$, then we will reach at least another coastal area with the control strategy j . We have

$$\begin{cases} \mathbf{x} \in \text{Back}(j, \bigcup_i \mathbb{C}_i) \\ \mathbf{x} \in \text{Forw}(j, \mathbb{C}_k), k \in \mathcal{I}_j \end{cases} \Rightarrow \mathbf{x} \in \mathbb{S}$$

Thus

$$\mathbb{S} \subset \bigcup_j \bigcup_{k \in \mathcal{I}_j} \text{Forw}(j, \mathbb{C}_k) \cap \text{Back}(j, \bigcup_i \mathbb{C}_i).$$

This property will allow us to have an inner approximation of the no-lost zone, which is our main contribution. This is illustrated by Figure 6 with 8 strategies: North, East, South, West, North-East, East-South, South-West, West-North.

Acknowledgement

This work has been supported by the French Government Defense procurement and technology agency (DGA). It also benefited from

the support of the project CONTREDO of the French National Research Agency (ANR).

References

- [1] F. Blanchini and S. Miani. *Set-Theoretic Methods in Control*. Springer Science & Business Media, Oct. 2007.
- [2] G. Chabert and L. Jaulin. Contractor Programming. *Artificial Intelligence*, 173:1079–1100, 2009.
- [3] V. Drevelle, L. Jaulin, and B. Zerr. Guaranteed characterization of the explored space of a mobile robot by using sub-pavings. In *Proc. Symp. Nonlinear Control Systems (NOLCOS'13)*, Toulouse, 2013.
- [4] T. L. Mézo, L. Jaulin, and B. Zerr. An interval approach to compute invariant sets. *IEEE Transaction on Automatic Control*, 62:4236–4243, 2017.
- [5] R. E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966.
- [6] S. Rohou, L. Jaulin, M. Mihaylova, F. L. Bars, and S. Veres. Guaranteed Computation of Robots Trajectories. *Robotics and Autonomous Systems*, 93:76–84, 2017.
- [7] J. A. D. Sandretto and A. Chapoutot. Validated simulation of differential algebraic equations with runge-kutta methods. *Reliable Computing*, 22, 2016.
- [8] G. Trombettoni and G. Chabert. Constructive Interval Disjunction. In *Proc. CP, Constraint Programming*, pages 635–650, LNCS 4741, 2007.

Zonotopic set-membership state estimation applied to an octorotor model

Dory Merhy^{*1}, Cristina Stoica Maniu¹, Teodoro Alamo², Eduardo F. Camacho²,
Thomas Chevet¹, Maria Makarov¹, and Israel Hinostroza³

¹Laboratoire des Signaux et Systèmes (L2S), CentraleSupélec-CNRS-Université Paris-Sud, Université Paris-Saclay, 3 rue Joliot Curie, F-91192, Gif-sur-Yvette, France

²Department of Ingeniería de Sistemas y Automática, Universidad de Sevilla, Camino de los Descubrimientos, 41092 Sevilla, Spain

³SONDRA Lab, CentraleSupélec, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette, France

Keywords: Set-membership state estimation, Octorotor, Linear Matrix Inequality

Introduction

In control systems, state estimators are mainly used to filter redundant data, to eliminate erroneous measurements and to produce reliable state estimations in the presence of measurement noises and perturbations. In 1960, Kalman set the ground for a new class of state estimation techniques by introducing his famous powerful yet simple filter, that considers known (Gaussian) distributions of measurement noises and state perturbations. Sometimes, the assumptions that the classical filter uses are not too realistic. Therefore, as an alternative, the deterministic approaches arose by considering unknown but bounded perturbations and measurement noises. Among this family, a particular interesting approach is the set-membership state estimation, where different sets can be used. The choice of the considered set mainly depends on the application and on the trade-off between accuracy and simplicity. However, despite the precision and the low complexity that some set-membership state estimation techniques can offer, there is still a gap between theory and practice in this field. In this context, few set-membership state estimators were tested on new technologies, in particu-

lar on Unmanned Aerial Vehicles (UAVs) [2], [6] and robots [3], or extended to incorporate physical state constraints [5]. In this work, a zonotopic set-membership state estimation technique is applied to the position estimation of an octorotor model used for radar applications. The model complexity and the perturbations coming from different sources make the state estimation of the drone a challenging problem. In this case, an accurate position estimation of the UAV is needed for the radar to provide high resolution images.

Zonotopic set-membership state estimation technique

Consider the following detectable discrete-time linear time invariant system:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{E}\boldsymbol{\omega}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{F}\boldsymbol{\omega}_k \end{aligned} \quad (1)$$

with $\mathbf{x}_k \in \mathbb{R}^{n_x}$, $\mathbf{u}_k \in \mathbb{R}^{n_u}$, $\mathbf{y}_k \in \mathbb{R}^{n_y}$, and $\boldsymbol{\omega}_k$ belonging to the unitary box $\mathbb{B}^{n_x+n_y}$.

Theorem 1. (based on [7]) Consider \mathbf{x}_0 and assume that the state \mathbf{x}_k belongs to the zonotope $\mathcal{Z}(\mathbf{p}_k, \mathbf{H}_k) = \mathbf{p}_k \oplus \mathbf{H}_k \mathbb{B}^m$. Given a scalar $\beta \in (0, 1)$, if there exist a positive definite matrix $\mathbf{P} = \mathbf{P}^\top \succ 0$ in $\mathbb{R}^{n_x \times n_x}$ and a matrix $\mathbf{Y} \in \mathbb{R}^{n_x \times n_y}$ for which the following linear matrix inequality (LMI) holds

$$\begin{bmatrix} \beta \mathbf{P} & 0 & \mathbf{A}^\top \mathbf{P} - \mathbf{C}^\top \mathbf{Y}^\top \\ * & \mathbf{T}^\top \mathbf{T} & \mathbf{E}^\top \mathbf{P} - \mathbf{F}^\top \mathbf{Y}^\top \\ * & * & \mathbf{P} \end{bmatrix} \succeq 0 \quad (2)$$

^{*}Corresponding author.

then it is guaranteed that $\mathbf{x}_{k+1} \in \mathcal{Z}(\bar{\mathbf{x}}_{k+1}, \mathbf{H}_{k+1})$, $\forall \boldsymbol{\omega}_k \in \mathbb{B}^{n_x+n_y}$, where:

$$\bar{\mathbf{x}}_{k+1} = \mathbf{A}\bar{\mathbf{x}}_k + \mathbf{B}\mathbf{u}_k + \mathbf{L}(\mathbf{y}_k - \mathbf{C}\bar{\mathbf{x}}_k) \quad (3)$$

$$\mathbf{H}_{k+1} = [\mathbf{A}_L \mathbf{H}_k \quad \boldsymbol{\eta}] \quad (4)$$

with $\mathbf{Y} = \mathbf{P}\mathbf{L}$, $\mathbf{T} = [\mathbf{E}^\top \quad \mathbf{F}^\top]^\top$, $\mathbf{A}_L = \mathbf{A} - \mathbf{L}\mathbf{C}$ and $\boldsymbol{\eta} = \mathbf{E} - \mathbf{L}\mathbf{F}$.

Sketch of proof: The error $\mathbf{z}_k = \mathbf{x}_k - \bar{\mathbf{x}}_k$ between the real state and the nominal estimated state at time k belongs to the centered zonotope $\mathbf{H}_k \mathbb{B}^m$. At time $k+1$, one has $\mathbf{z}_{k+1} = \mathbf{A}_L \mathbf{z}_k + \boldsymbol{\eta} \boldsymbol{\omega}_k \in \mathbf{H}_{k+1} \mathbb{B}^{m+n_x+n_y}$.

The non increase of the P-radius [4] of the zonotopic error can be expressed such that $\max_{\hat{\mathbf{z}}} \|\mathbf{H}_{k+1} \hat{\mathbf{z}}\|_{\mathbf{P}}^2 \leq \beta \max_{\mathbf{z}} \|\mathbf{H}_k \mathbf{z}\|_{\mathbf{P}}^2 + \max_{\mathbf{t}} \|\mathbf{T}\mathbf{t}\|_2^2$ with the notations $\hat{\mathbf{z}} = [\mathbf{z}^\top \quad \mathbf{t}^\top]^\top \in \mathbb{B}^{m+n_x+n_y}$, $\mathbf{z} \in \mathbb{B}^m$ and $\mathbf{t} \in \mathbb{B}^{n_x+n_y}$.

Using the reverse triangle inequality leads to a sufficient condition for $\max_{\hat{\mathbf{z}}} (\|\mathbf{H}_{k+1} \hat{\mathbf{z}}\|_{\mathbf{P}}^2 - \beta \|\mathbf{H}_k \mathbf{z}\|_{\mathbf{P}}^2 - \|\mathbf{T}\mathbf{t}\|_2^2) \leq 0$. Extensively, $\forall \mathbf{z}, \mathbf{t}$, the next expression is verified

$$\hat{\mathbf{z}}^\top \mathbf{H}_{k+1}^\top \mathbf{P} \mathbf{H}_{k+1} \hat{\mathbf{z}} - \beta \mathbf{z}^\top \mathbf{H}_k \mathbf{P} \mathbf{H}_k \mathbf{z} - \mathbf{t}^\top \mathbf{T}^\top \mathbf{T} \mathbf{t} \leq 0 \quad (5)$$

Replacing $\mathbf{H}_{k+1} \hat{\mathbf{z}} = (\mathbf{A} - \mathbf{L}\mathbf{C})\mathbf{H}_k \mathbf{z} + (\mathbf{E} - \mathbf{L}\mathbf{F})\mathbf{t}$ in Eq. (5) and using the Schur complement lead us to the LMI (2).

Octorotor modeling

The Mikrokopter ARF Okto-XL is equipped with a micro-controller that provides fused and filtered information from the sensors about the drone's position. A non-linear dynamical model together with a linearized model around the static hovering equilibrium with null translational and rotational velocities and null roll, pitch and yaw angles exist [1]. The linearized model [1] can be decoupled into three double integrator subsystems and then discretized with a sampling period T_s . However, for linear position estimation problems, we only need the two subsystems describing the longitudinal and the altitude dynamics, respectively:

$$\begin{aligned} \mathbf{x}_{1_{k+1}} &= \mathbf{A}\mathbf{x}_{1_k} + \mathbf{B}_1 \mathbf{u}_{1_k} + \mathbf{E}_1 \boldsymbol{\omega}_k \\ \mathbf{y}_{1_k} &= \mathbf{C}\mathbf{x}_{1_k} + \mathbf{F}_1 \boldsymbol{\omega}_k \end{aligned} \quad (6)$$

$$\begin{aligned} \mathbf{x}_{3_{k+1}} &= \mathbf{A}\mathbf{x}_{3_k} + \mathbf{B}_3 \mathbf{u}_{3_k} + \mathbf{E}_3 \boldsymbol{\omega}_k \\ \mathbf{y}_{3_k} &= \mathbf{C}\mathbf{x}_{3_k} + \mathbf{F}_3 \boldsymbol{\omega}_k \end{aligned} \quad (7)$$

with $\mathbf{x}_{1_k} = [z_k \quad \psi_k \quad V_{z_k} \quad \omega_{z_k}]^\top$, $\mathbf{x}_{3_k} = [x_k \quad y_k \quad V_{x_k} \quad V_{y_k}]^\top$, $\mathbf{u}_{1_k} = [F_{z_k}^R \quad \tau_{z_k}^R]^\top$, $\mathbf{u}_{3_k} = [F_{x_k}^R \quad F_{y_k}^R]^\top$, $\mathbf{y}_{1_k} = [z_k \quad \psi_k]^\top$, $\mathbf{y}_{3_k} = [x_k \quad y_k]^\top$, $\mathbf{A} = \begin{bmatrix} I_2 & T_s I_2 \\ 0_2 & I_2 \end{bmatrix}$, $\mathbf{B}_1 =$

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{T_s}{m} & 0 \\ 0 & \frac{T_s}{I_{zz}} \end{bmatrix}, \mathbf{B}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{T_s}{m} & 0 \\ 0 & \frac{T_s}{m} \end{bmatrix}, \mathbf{C} = [I_2 \quad 0_2].$$

The notations and parameter values are detailed in [1]. Furthermore, the perturbations and the measurement noises $\boldsymbol{\omega}_k$ are bounded by the unitary box \mathbb{B}^6 . Additionally, $\mathbf{E}_i = \epsilon_i \cdot [I_4 \quad 0_{4 \times 2}]$, $\mathbf{F}_i = \gamma_i \cdot [0_4 \quad I_{4 \times 2}]$, for $i \in \{1, 3\}$, with ϵ_i and γ_i two scalars representing the accuracy provided by the drone sensors. The control inputs F_x^R , F_y^R and F_z^R are the components of the resulting propeller's force, whereas τ_z^R is the component of the resulting propeller's torque expressed in the drone's frame denoted by the superscript R .

Simulation results

The highest sampling period $T_s = 0.02$ s of all sensors is considered. The systems are fully controllable and observable. Based on the GPS, altimeter and gyroscope information, the following values are considered for $\gamma_1 = \gamma_3 = 1$ and $\epsilon_1 = \epsilon_3 = 10^{-3}$. The UAV mass is 3.69kg and the inertia component I_{zz} w.r.t. to the z-axis is 0.0869kg·m². The drone's behavior was tested using a Matlab/Simulink simulator implementing the non-linear model with a linear quadratic integral (LQI) controller [1] for which the nominal control inputs are then fed into the linear designed system. A linear trajectory is simulated to validate the efficiency of the zonotopic set-membership esti-

mation technique. It corresponds to a take-off to an altitude of 50m and then to a flight on the x-axis with a linear constant speed. The flight duration is 235s.

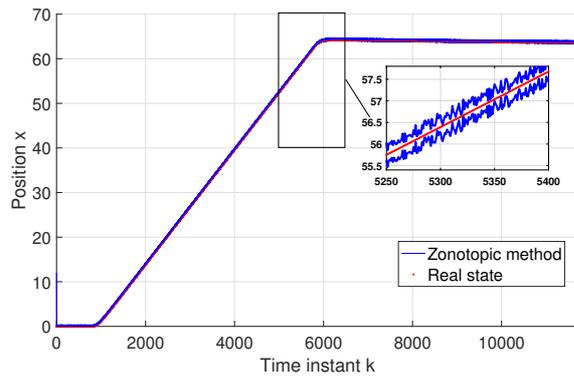


Figure 1: Bounds of the linear position x

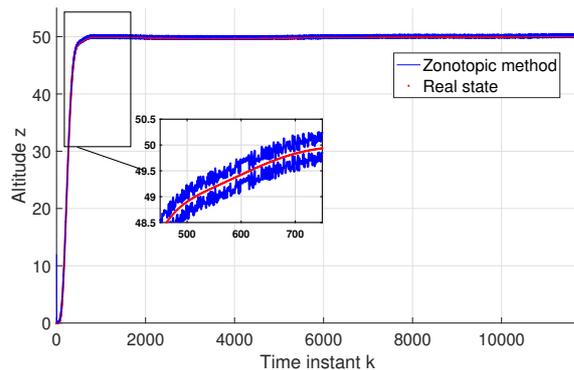


Figure 2: Bounds of the altitude z

Figure 1 shows the zonotopic bounds (in blue) of the linear position x of the drone, whereas Figure 2 presents the guaranteed estimation bounds (in blue) of the altitude z . The real state (in red) in both cases lies inside the bounds despite of the considered measurement noises and state perturbations.

Conclusion

A guaranteed zonotopic set-membership state estimation technique has been considered to compute the guaranteed linear position bounds of an octorotor model.

Acknowledgement

The authors acknowledge MINERCO, FEDER funds, and EU Programme H2020 for funding projects DPI2016-76493-C3-1-R and SI-1838/24/2018.

References

- [1] T. Chevet, M. Makarov, C. Stoica Maniu, I. Hinostroza, and P. Tarascon. State estimation of an octorotor with unknown inputs. Application to radar imaging. *21st ICSTCC*, 2017.
- [2] R. A. Garcia, G. V. Raffo, M. G. Ortega, and F. R. Rubio. Guaranteed quadrotor position estimation based on GPS refreshing measurements. *1st IFAC Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles*, 48(9), 2015.
- [3] L. Jaulin. Robust set-membership state estimation; application to underwater robotics. *Automatica*, 45, 2009.
- [4] V. T. H. Le, C. Stoica, T. Alamo, E. Camacho, and D. Dumur. Zonotopic guaranteed state estimation for uncertain systems. *Automatica*, 49, 2013.
- [5] D. Merhy, T. Alamo, C. Stoica Maniu, and E. F. Camacho. Zonotopic constrained Kalman filter based on a dual formulation. In *IEEE CDC*, 2018.
- [6] G. Rousseau, C. Stoica Maniu, S. Tebbani, M. Babel, and N. Martin. Minimum-time B-spline trajectories with corridor constraints. Application to cinematographic quadrotor flight plans. *Control Engineering Practice*, 89, 2019.
- [7] Y. Wang, V. Puig, and G. Cembrano. Set-membership approach and Kalman observer based on zonotopes for discrete-time descriptor systems. *Automatica*, 93, 2018.

A Polytopic Box Particle Filter for state estimation of Non Linear Discrete-Time Systems

Thomas Gatto^{*1}, Luc Meyer¹, and H el ene Piet-Lahanier¹

¹D epartement Traitement de l'Information et Syst emes, ONERA, Universit e Paris Saclay

Keywords: Particle Filter, Intervals, Orthotopes, Ellipsoids, Estimation, Bounded noise, Set-Membership uncertainty, Uncertain dynamic systems

Introduction

State estimation of dynamic systems is commonly addressed by modelling the uncertainty as a stochastic variable, usually assumed Gaussian. For linear or non-linear systems, such problems are solved by using a classical (KF), an extended (EKF) or an unscented (UKF) Kalman Filter. For non-linear systems, particle filters have been developed to tackle non-Gaussian noise distributions. However, stochastic representation of errors is not immune to criticism as the probability density function is seldom known a priori. In set-membership estimation, process and measurement uncertainties are only assumed to vary within known bounds which makes this type of approach very robust to lack of probabilistic information. Various set structures have been used to characterize the variation domain of the system states, given the model structure and bounds. However, this results often in a pessimistic estimation, especially for multi-modal distributions. A more recent alternative method, first introduced by [1] consists in combining the versatility of the particle representation with the robustness of set-membership method. This translates in replacing the point particle by a box which results in reducing significantly the number of particles and the adverse effects of non-linearity. Box Particle Filter (BPF) esti-

matoms have already been applied in Simultaneous Localization and Mapping (SLAM) or mobile localization [1, 7]. However, the BPF provides a rather pessimistic solution due to the fact that the intervals have to be aligned along the state axis which result in losing potential dependencies between the resulting estimate components. To address this issue, an improvement of the box description could be to combine this description with a more precise set characterization using either ellipsoidal [2, 4, 5] or polyhedral boundaries [6]. The aim of the present work is to build a new box particle filter based partially on polytopic description.

Problem Statement

Consider the following non linear discrete-time system:

$$\begin{cases} x_{k+1} &= f(x_k) + w_k \\ y_k &= h(x_k) + v_k \end{cases}, \quad (1)$$

where $x_k \in \mathbb{R}^{n_x}$ is the state vector, $y_k \in \mathbb{R}^{n_y}$ the measurement vector, $f : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ a non-linear function and w_k , a process noise vector. We denote by n_x, n_w , respectively, the dimensions of the state and process noise vectors. The function $h : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ is a non-linear function and v_k a measurement noise vector. Dimensions of the measurement and measurement noise vectors are respectively n_y and n_v .

Assumption 1. The disturbance terms w_k and v_k are assumed to be unknown but bounded (UBB) noises:

$$|w_{k,i}| \leq \varepsilon_{k,i}^w, i = 1, \dots, n_w \iff \|w_k\|_{\infty}^w \leq 1, \quad (2)$$

^{*}Corresponding author.

$$|v_{k,i}| \leq \varepsilon_{k,i}^v, i = 1, \dots, n_v \iff \|v_k\|_{\infty}^{\varepsilon_k^v} \leq 1. \quad (3)$$

Definition 1. A real interval, denoted $[x]$, is defined as a closed and connected subset of \mathbb{R} and a box $[X]$ of \mathbb{R}^{n_x} as a Cartesian product of n_x intervals: $[X] = [x_1] \times [x_2] \times \dots \times [x_{n_x}] = \times_{i=1}^{n_x} [x_i]$.

Definition 2. An n -dimensional polyhedron P is defined as a set of n_p vertices $\mathbb{V}_i, i = 1, \dots, n_p$ and n_h supporting hyper-planes \mathbb{H}_j .

Each of the n_h hyper-planes is defined by $\{x \in \mathbb{R}^n | a_i x = b_i\}$, where $a_i^T \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$. Therefore, a n -dimensional polyhedron P supporting n_h hyper-planes is defined by :

$$\{x \in \mathbb{R}^n | Ax \leq b\}, \quad (4)$$

where $A \in \mathbb{R}^{n_h \times n}$, a_i the i -th row of A , $b \in \mathbb{R}^{n_h}$ and b_i the i -th component of b .

Proposed algorithm

The algorithm is based on the BPF algorithm. The main originality consists in modifying the update step of the BPF by replacing the measurement boxes by polytopes to improve accuracy.

Initialization

As in the BPF, the initialization consists in creating N_p box particles from the initial box with minimum intersection and equivalent weights.

Prediction

In this step, each state predicted particle is computed based on the previous state estimated particle, via a classical interval propagation.

Measurement update

The observation function h is linearized at the center \hat{x}_k of the predicted box:

$$h(x_k) = h(\hat{x}_k) + C_k(x_k - \hat{x}_k) + o_k, \quad (5)$$

where $C_k = \frac{\partial h(\hat{x}_k)}{\partial x}$; o_k is the linearization error. The measurement bounds $[m_k]$ are obtained as $[o_k] + [v_k]$. For each measurement y_k , two bounding hyperplanes are defined as $h(\hat{x}_k) + C_k(x_k - \hat{x}_k) = y_k + \min([m_k])$ and $h(\hat{x}_k) + C_k(x_k - \hat{x}_k) = y_k + \max([m_k])$.

Using the approach described in [6], the measurement update step consists in computing the feasible set for each particle by intersecting the predicted box particles with the two half spaces associated with each of the bounding hyperplanes. The volumes of the resulting polytopes are computed as in [3], and will be used as weight for each polyhedron particle.

Estimation

At the k -th step, the state is usually approximated using the weighted particles, as $\hat{x}_k = \sum_{i=1}^{N_p} w_k^i x_k^i$. In the case of box particles, so on the BPF, the state is actually computed as $\hat{x}_k = \sum_{i=1}^{N_p} w_k^i C_k^i$, where C_k^i is the center of the box particle i . However, in our proposed filter, the new estimated state is computed as the center of the polytope i which is obtained as $C_k^i = \frac{1}{n_p} \sum_{j=1}^{n_p} \mathbb{V}_{k,j}$ where $\mathbb{V}_{k,j}$ is the j -th vertice of the polytope i at time k .

Similarly to the BPF, the associated covariance matrix is given by $\hat{P}_k = \sum_{i=1}^{N_p} w_k^i (\hat{x}_k - x_k^i)(\hat{x}_k - x_k^i)^T$.

Resampling

The resampling phase consists in eliminating polytopes associated with the lowest weights, and in dividing the polytopes associated with the highest weights. These weights are obtained by computing the volume of each polytope. After selection of the polytopes to be kept, each of those is approximated by the smallest box containing it. Figure 1 illustrates the measurement update phase. It can be seen that the polyhedral update (in green) makes the resulting estimation uncertainty less pessimistic than with the classical Box Resampling (in black).

Several examples of non linear model estimation have been tested to evaluate the average precision improvement resulting from the use of the new method.

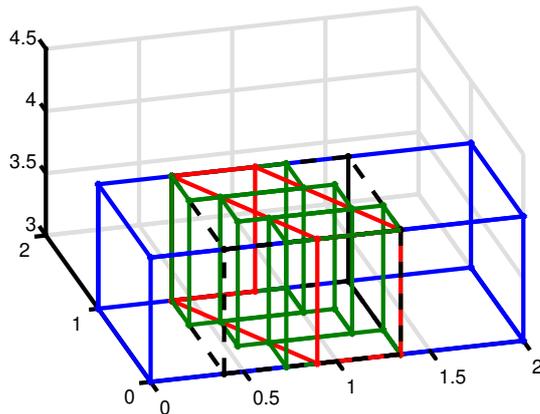


Figure 1: Illustration of the measurement update using polyhedrons. Blue: predicted box. Red: half spaces associated with each of the bounding hyperplans. Green: set of new boxes after resampling. Black: set that would be obtained with classical Box Particle Filter.

Conclusion

In this paper, improvement of box particle filter based on polytopic measurement updating is proposed. Different examples of application have been compared with the BPF and the results are promising. The estimate is more precise, especially if all the variables are measured. However, for now, the computing time is still uncertain because it depends on the dimensions of the state and the measure. Future work includes analysis of the computation of the bounds on measurements allowing the best compromise between reliability and precision. Evaluation of weights depending not only on the volume of the resulting polytopes is also under study.

References

- [1] F. Abdallah, A. Gning, and P. Bonnifait. Box particle filtering for nonlinear state estimation using interval analysis. *Automatica*, 44(3):807–815, 2008.
- [2] Z. Bo, Q. Kun, M. Xu-Dong, and D. Xian-Zhong. A new nonlinear set membership filter based on guaranteed bounding ellipsoid algorithm. *Acta Automatica Sinica*, 39(2):146–154, 2013.
- [3] J. B. Lasserre. An analytical expression and an algorithm for the volume of a convex polyhedron in n . *Journal of optimization theory and applications*, 39(3):363–377, 1983.
- [4] B. T. Polyak, S. A. Nazin, C. Durieu, and E. Walter. Ellipsoidal parameter or state estimation under model uncertainty. *Automatica*, 40(7):1171–1179, 2004.
- [5] E. Scholte and M. E. Campbell. A nonlinear set-membership filter for on-line applications. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 13(15):1337–1358, 2003.
- [6] E. Walter and H. Piet-Lahanier. Exact recursive polyhedral description of the feasible parameter set for bounded-error models. *IEEE Transactions on Automatic Control*, 34(8):911–915, 1989.
- [7] P. Wang, P. Xu, P. Bonnifait, and J. Jiang. Box particle filtering for slam with bounded errors. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1032–1038. IEEE, 2018.

Rigorous bounds for ill-posed linear programming problems

Marko Lange

Institute for Reliable Computing, Hamburg University of Technology

Keywords: Ill-posedness; Linear programming; Verification; Facial reduction; NETLIB LP library

Preliminaries

We are concerned with linear programming problems of the form

$$\begin{aligned} \inf_{x_f, x_l} \quad & c_f^T x_f + c_l^T x_l \\ & A_f x_f + A_l x_l = b \\ & x_l \geq 0, \end{aligned} \quad (\text{LP})$$

Motivation

In [8], Ordóñez and Freund have shown that 71% of the instances from the Netlib test suite [7] - a benchmark suite containing difficult but practically relevant linear programming problems - have infinite condition measure. Due to their practical background and possible inaccuracies in the input data, these problem instances are very interesting for the application of verification methods.

However, the solvability principle of verification methods states that verification methods solve well-posed problems [10]: it is typically not possible to compute rigorous inclusions for the optimal value of an ill-posed problem using floating-point arithmetic because even the slightest perturbation may change its feasibility status. Without incorporating additional knowledge about the ill-posed instances from the NETLIB test suite, verification tools are not able to compute verified bounds for these problems (cf. [5]).

We are proposing an error free preprocessing procedure to replace a given ill-posed linear programming problem with an equivalent well-posed problem. We demonstrate the applicability of our procedure by computing new verified bounds for a large number of ill-posed problem instances from the NETLIB linear programming library.

where $x_f \in \mathbb{R}^{n_f}, x_l \in \mathbb{R}^{n_l}$ are free and non-negative decision variables, respectively, $A_\diamond \in \mathbb{R}^{m \times n_\diamond}, b \in \mathbb{R}^m$ and $c_\diamond \in \mathbb{R}^{n_\diamond}$ for $\diamond \in \{f, l\}$.

In accordance with Renegar's definition in [9], a feasible instance of (LP) is ill-posed if infinitesimal small perturbations can render the problem infeasible. This is precisely the case if the sets $\{A_f x_f + A_l x_l \mid x_l \geq 0\}$ and $\{b\}$ are separable by a hyperplane, but not strictly so.

If $u \in \mathbb{R}^m$ is a normal vector to such a hyperplane, then

$$(A_f x_f + A_l x_l)^T u \leq b^T u \quad (1)$$

is satisfied for all $x_f \in \mathbb{R}^{n_f}, 0 \leq x_l \in \mathbb{R}^{n_l}$. It is straightforward to show that (1) is equivalent to the conditions

$$(A_f x_f)^T u = 0, \quad (A_l x_l)^T u \leq 0, \quad b^T u \geq 0.$$

Since the separation is not strict, there exist x_f and a non-negative vector x_l such that (1) is satisfied with equality and therefore $b^T u = 0$. Summarizing, one can prove the following crucial equivalence.

Proposition. *A feasible instance of (LP) is ill-posed if, and only if, the conditions*

$$A_f^T u = 0, \quad A_l^T u \leq 0, \quad b^T u = 0 \quad (2)$$

are satisfied for non-trivial vectors $u \in \mathbb{R}^m$.

Reduction procedure

The conditions in (2) lead to another linear programming problem which can be used to compute suitable vectors u . If $u = 0$ is the only feasible solution to this problem, then (LP) is well-posed. Otherwise we found an indicator for ill-posedness.

Moreover, (2) can be used not only to detect ill-posedness but also for its removal. For any u satisfying (2) and every feasible point (x_f, x_l) of (LP), we have

$$0 = \underbrace{(A_f x_f + A_l x_l - b)^T}_{\equiv 0} u = \underbrace{x_l^T}_{\geq 0} \underbrace{A_l^T u}_{\leq 0}$$

and therefore

$$\forall 1 \leq i \leq n_l: (A_l^T u)_i \cdot (x_l)_i = 0. \quad (3)$$

If the equality constraints of (LP) are linearly independent, by which $u \neq 0$ implies $[A_l, A_f, b]^T u \neq 0$, then

$$\exists i: (A_l^T u)_i \neq 0, (x_l)_i = 0.$$

In this case, (LP) can be reduced to a smaller equivalent linear programming problem by eliminating the respective entries $(x_l)_i$. On the other hand, if there are linearly dependent equality constraints, we may remove them and again obtain a reduced equivalent problem. This can be repeated up to the point where we derive a well-posed problem.

Following, one may use some verification tool for linear programming problems to compute verified bounds also for the original (LP).

Verification

The reduction approach is not exactly a new concept. The set described by the conditions in (2) is strongly related to the set of dual recession directions [2, 3] and our reduction approach is a specific form of facial reduction [1, 2, 6].

The actual difficulty lies in the verification of the reduction procedure. How can we compute rigorously an interval U that contains an

Problem	only SDPT3	VSDP with (PP)
25FV47	$1.50 \times 10^{+3}$	1.94×10^{-8}
CZPROB	1.67×10^{-1}	1.14×10^{-8}
MODSZK1	$7.67 \times 10^{+8}$	1.02×10^{-2}
SCFXM1	1.42×10^{-10}	3.69×10^{-9}
SHIP12S	$8.70 \times 10^{+1}$	9.48×10^{-10}

Table 1: Duality gaps for selected LPs

actual solution to (2) but not the trivial vector of all zeros?

After using an auxiliary linear programming problem to compute an approximate solution to (2) and deciding which entries of u are nonzero via a simple threshold approach, the task reduces to determining and verifying linearly dependent equality constraints. By exploiting a line-up of presuppositions that are satisfied for most instances from the NETLIB linear programming library [7], it is actual possible to reduce the respective set of equality constraints to a linearly independent basis in a rigorous manner.

To demonstrate the applicability of our approach, in Table 1, we present relative duality gaps for some ill-posed instances from the NETLIB library. We chose problems for which VSDP [4] fails to compute rigorous bounds without prior preprocessing (PP). The used solver is SDPT3 [11].

References

- [1] J. M. Borwein and H. Wolkowicz. Facial reduction for a cone-convex programming problem. *J. Austral. Math. Soc. Ser. A*, 30(3):369–380, 1981.
- [2] V. Y.-L. Cheung. *Preprocessing and Reduction for Semidefinite Programming via Facial Reduction: Theory and Practice*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 2013.
- [3] V. Y.-L. Cheung and H. Wolkowicz. Sensitivity analysis of semidefinite programs

- without strong duality. Technical report, 2014.
- [4] V. Härter, C. Jansson, and M. Lange. VSDP: A Matlab toolbox for verified semidefinite-quadratic-linear programming. <http://www.ti3.tuhh.de/jansson/vsdp/>, 2012.
 - [5] C. Keil and C. Jansson. Computational experience with rigorous error bounds for the netlib linear programming library. *Reliab. Comput.*, 12(4):303–321, 2006.
 - [6] B. F. Lourenço, M. Muramatsu, and T. Tsuchiya. Solving SDP completely with an interior point oracle. *ArXiv e-prints*, 2015.
 - [7] NETLIB linear programming library. A collection of linear programming problems, <http://www.netlib.org/lp/>.
 - [8] F. Ordóñez and R. M. Freund. Computational experience and the explanatory value of condition measures for linear optimization. *SIAM J. Optim.*, 14(2):307–333, 2003.
 - [9] J. Renegar. Incorporating condition measures into the complexity theory of linear programming. *SIAM J. Optim.*, 5(3):506–524, 1995.
 - [10] S. M. Rump. Verification methods: Rigorous results using floating-point arithmetic. *Acta Numer.*, 19:287–449, 2010.
 - [11] K.-C. Toh, M. J. Todd, and R. H. Tuntuncu. On the implementation and usage of SDPT3 – a Matlab software package for semidefinite-quadratic-linear programming, version 4.0. Technical report, 2006.

Computation of integrals with interval endpoints

Olivier Mullier*¹ and Julien Alexandre dit Sandretto¹

¹ENSTA ParisTech, 828 Boulevard des maréchaux, 91120 Palaiseau

¹{mullier, alexandre}@ensta.fr

Keywords: Integral; Set-membership Computation; Interval Methods.

Introduction

Numerical integration is one of the fundamental tool of scientific computation. Providing a reliable result to such problem is important for validated simulation [1] or for global optimization with a continuous objective function [3]. An important work on inclusion methods for integral equations can be found in [2]. In our presentation, we propose an efficient guaranteed method for the computation of the integral of a nonlinear continuous function f between two interval endpoints $[x_1]$ and $[x_2]$, we call interval integrals:

Definition 1 (Interval integral). Let $f : \mathbb{R} \rightarrow \mathbb{R}$, a continuous function and $[x_1], [x_2] \in \mathbb{IR}$ two intervals. The interval integral of f with $[x_1]$ and $[x_2]$ as endpoints is denoted $\int_{[x_1]}^{[x_2]} f(x)dx$ and corresponds to the set

$$\int_{[x_1]}^{[x_2]} f(x)dx = \left\{ \int_{x_1}^{x_2} f(x)dx \mid \begin{array}{l} x_1 \in [x_1] \\ x_2 \in [x_2] \end{array} \right\}. \quad (1)$$

This set considers all the integrals with the endpoints taken in the intervals $[x_1]$ and $[x_2]$. Three cases can occur whether the interval endpoints $[x_1]$ and $[x_2]$ are disjoint, intersect or one is included in the other.

The endpoints are disjoint As introduced in [2], an interval integral as defined in

Definition 1 where the endpoints are disjoint can be decomposed as follows

$$\int_{[x_1]}^{[x_2]} f(x)dx = \int_{[x_1]}^{\overline{x_1}} f(x)dx + \int_{\underline{x_2}}^{\overline{x_2}} f(x)dx + \int_{\underline{x_2}}^{[x_2]} f(x)dx. \quad (2)$$

The endpoints intersect The interval integral in Eq. (1) can be subdivided with

$$\int_{[x_1]}^{[x_2]} f(x)dx = \int_{[x_1, x_2]} f(x)dx + \int_{[\underline{x_2}, \overline{x_1}]} f(x)dx + \int_{[\underline{x_2}, \overline{x_1}]} f(x)dx. \quad (3)$$

The first and the last interval integrals in the right member of Eq. (3) are of the same type as the one where endpoints are disjoint except that the integral can be equal to 0 when taking both the same endpoints.

One endpoint is included in the other

When $[x_1] \subseteq [x_2]$, we have $\underline{x_2} \leq \underline{x_1} \leq \overline{x_1} \leq \overline{x_2}$ and the same decomposition as in Eq. (3) is possible:

$$\int_{[x_1]}^{[x_2]} f(x)dx = \int_{[x_1]}^{[\underline{x_2}, \overline{x_1}]} f(x)dx, + \int_{[x_1]}^{[x_1]} f(x)dx, + \int_{[\overline{x_1}, \overline{x_2}]} f(x)dx \quad (4)$$

so we go back to the already treated kind of interval integral that occurred in the previous cases.

We see that in all cases, only three interval integrals occur:

$$\int_{[x]}^{\overline{x}} f(x)dx; \int_{\underline{x}}^{[x]} f(x)dx; \int_{[x]}^{[x]} f(x)dx. \quad (5)$$

*Corresponding author.

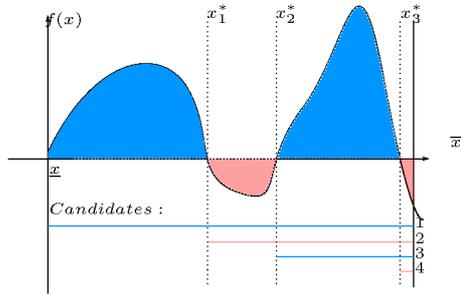


Figure 1: Example of computation of $\int_{[x]}^{\bar{x}} f(x)dx$ for $\mathcal{X}^* = \{x_1^*, x_2^*, x_3^*\}$ (blue: maximum; red: minimum).

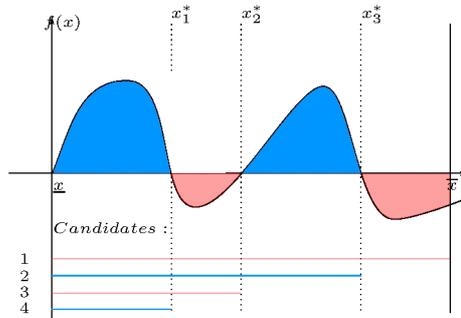


Figure 2: Example of computation of $\int_{\underline{x}}^{[x]} f(x)dx$ for $\mathcal{X}_2^* = \{x_1^*, x_2^*, x_3^*\}$.

Producing the minimum and the maximum of these interval integrals requires the parts where sub-integrals are positive and parts where they are negative. The change between positiveness and negativeness of the integral occurs at x such that $f(x) = 0$. Computing the minimum and maximum then requires to produce the set $\mathcal{X}^* = \{x \in [x] : f(x) = 0\}$. The minimum and the maximum candidates for all the interval integrals in Eq (5) can be defined using \mathcal{X}^* . When the arity of \mathcal{X}^* is finite, the set of candidate to consider is then finite as well. Figure 1 provides an illustration of the candidates for $\int_{[x]}^{\bar{x}} f(x)dx$. In this case, we only have to consider 4 integral candidates to be the minimum and the maximum. In Figure 2, an illustration of $\int_{\underline{x}}^{[x]} f(x)dx$ is illustrated. The method for the computation of an

interval integral consists in the computation of the set \mathcal{X}^* and to find all the candidates to be the minimum and the maximum of the set described in Eq. (1). The method then provides the interval outer approximation of this set and also the endpoints at play.

Example We consider the computation of the interval integral $\int_0^{[0,1]} \frac{dx}{1+x^2}$. The result is:

$$\left[\int_0^0 \frac{dx}{1+x^2}, \int_0^1 \frac{dx}{1+x^2} \right] \subset [0, 0.78543] \quad (6)$$

The implementation of the computation of any interval integral is linear on the arity of \mathcal{X}^* for $\int_{\underline{x}}^{[x]} f(x)dx$ and $\int_{[x]}^{\bar{x}} f(x)dx$. For $\int_{[x]}^{[x]} f(x)dx$, we need to consider the backward integrals as well since the first endpoint can be greater than the last one. The proposed algorithm is factorial on the arity of \mathcal{X}^* . This will be detailed in the presentation.

Acknowledgment

This research was supported of the “Chair Complex Systems Engineering - Ecole Polytechnique, THALES, DGA, FX, DASSAULT AVIATION, DCNS Research, ENSTA Paris-Tech, Télécom ParisTech, Fondation Paris-Tech and FDO ENSTA”. This research is also partially funded by DGA MRIS.

References

- [1] J. Alexandre dit Sandretto and A. Chapoutot. Validated explicit and implicit runge-kutta methods. *Reliable Computing electronic edition*, 22, 2016.
- [2] G. F. Corliss. Computing narrow inclusions for definite integrals. *Computer arithmetic*, pages 150–179, 1987.
- [3] E. Hansen. Global optimization using interval analysis—the multi-dimensional case. *Numerische Mathematik*, 34(3):247–270, 1980.

Identification of Multi-Faults in GNSS Signals using RSIVIA under Dual Constellation

Shuchen Liu^{*1}, Jan-Jöran Gehrt¹, Dirk Abel¹, and René Zweigel¹

¹Institute of Automatic Control, RWTH Aachen University

Keywords: RSIVIA, GPS, Galileo, Integrity, Fault Identification, Navigation Filter

Introduction

As described in the Market Report from European Global Navigation Satellite Systems Agency (GSA), satellite-based navigation will substantially contribute to the future innovation of self-driving vehicles (see [1]). In autonomous applications, especially in safety-critical scenarios, a false estimation of vehicle state can result in catastrophic accidents, which requires the high accuracy and integrity of the navigation solution. To maintain the integrity of a global navigation satellite system (GNSS)-based navigation system, the faulty GNSS observations caused by signal interferences and other possible reasons shall be detected, identified and excluded. Since the open service of the newly developed EU satellite navigation system Galileo is in operation, the combination of GPS and Galileo provides the modern navigation systems more available satellites in view. However, a higher number of satellites also increases the possibility that satellite observations contain a fault or even multi-faults. Therefore, identification of multi-faults becomes a crucial and challenging task to maintain the integrity of GNSS-based navigation systems.

The previous work [2] presents the development of a fault detection and exclusion (FDE) algorithm of GNSS measurements. The approach is an extension of an existing tightly-coupled navigation filter, which integrates the measurements from GNSS and an inertial

measurement unit (IMU). In [2], FDE bases on the receiver integrity monitoring (RAIM) approach, which is a pure statistical method. RAIM predicts pseudorange residual, which is based on estimated reference vehicle state using least square method, and uses the residual to detect and identify pseudorange faults. This method might not be adequate, when many of the measurements are faulty, since it is originally developed under single fault assumption. This work concentrates on multi-faults identification, when the conventional statistic based approach cannot provide a correct identification solution certainly.

In recent years, an alternative localization method, Set Inversion via Interval Analysis (SIVIA), is developed under such concern in [3] and applied to realize robot localization in [4]. SIVIA guarantees integrity and estimates a trust region of the antenna position fulfilling a predefined confidence level. Further, robust SIVIA (RSIVIA) approach is applied for satellite positioning in [5], which allows to estimate the trust region under the assumption of erroneous pseudorange measurements. Hereby, it is possible to identify outliers in the GNSS observations by checking the compatibility of each GNSS measurement and the estimated trust region. This trust region is calculated as a wrap of several sub-pavings, which makes it less sensitive with the pseudorange errors, i.e. the pseudorange error should be relatively big to be identified. Another drawback of this approach is its computational load, because RSIVIA begins with an initial guess of an arbitrary big box, bisects it into small boxes and operates on them separately and iteratively.

The present research applies RSIVIA for fault detection and identification in a dual-

^{*}Corresponding author.

constellation based-navigation system, where RSIVIA is executed in an iterative process: it starts with the assumption that no fault exists in the observed measurement space. Whenever an empty trust region is returned, RSIVIA assumes one more fault existing in the measurements. This iterative process continues until a non-empty trust region is estimated. Instead of checking the compatibility of each GNSS measurement with the estimated trust region [5], which is a wrap of several sub-pavings, the compatibility of these measurements with each existing sub-paving is checked to identify the faults. To reduce the computational load, RSIVIA does not start with an arbitrary big box, instead, the middle point of the initial box is the estimated states from the navigation filter and the size is calculated as a function of maximum dynamic from the experimental vehicle.

This extended abstract contains the following contents: First, the background of current work was introduced, including the motivation and previous work. Then, the approach applied in this work was described briefly. Further, the next section describes the measurement setup that is used to validate the approach and provides the first experimental results. Finally, the last section draws the conclusion and provides an outlook for the ongoing and future developments.

Experimental Validation

Measurement Setup

The test trajectory in the experimental validation is defined and simulated in a NCS TITAN GNSS simulator from IFEN GmbH. The TITAN generates GNSS observables and supports all existing GNSS systems and provides up to 256 signal channels. Further, the inertial measurements are also simulated by TITAN, with respect to the defined trajectories, the virtual vehicle characteristic and the noise level of a real LORD MicroStrain 3DM-GX4-25 industrial-class IMU-sensor. Compared to

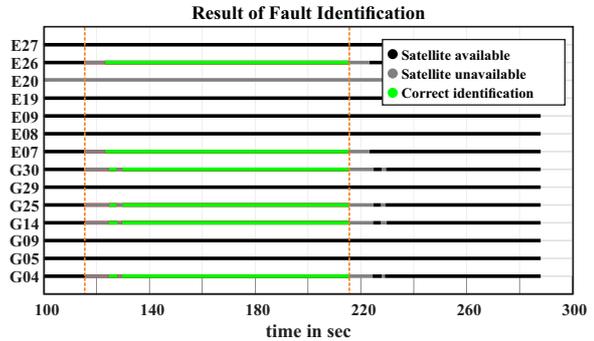


Figure 1: Identified erroneous satellite signals

real world measurement campaign, the TITAN can generate the GNSS signals with feared event, which is the scenario containing pseudorange errors, and record it, which can be used as the reference of the fault identification outputs.

The generated GNSS signal is received and decoded by a Septentrio AstRx3 HDC receiver at a rate of 10 Hz. The navigation filter is implemented on a 900 MHz single core Rapid Control Prototyping (RCP) unit, called MicroAutoBox from the manufacture dSPACE. The communication between the receiver and the RCP unit is achieved via serial interface. The receiver provides a pulse per second (PPS). Using the PPS, the communication and processing delays of the receiver are compensated (see [6]). All the GNSS and IMU measurements are recorded and the test scenario with RSIVIA is reproduced in a post-processing environment.

Experimental Results

Figure 1 shows the first experimental results of the fault identification of multi-faults in GNSS measurements. In Figure 1, the label of y-axis shows all satellites that are simulated in this test, where 'G' stands for GPS and 'E' for Galileo satellites. In total, seven GPS L1/L2 satellites and seven Galileo E1/E5a satellites are simulated. In this figure, the black parts of the lines show the epochs, when a satellite is available for the RSIVIA, otherwise, it is in

gray. Satellite E20 is simulated during the entire test, but blocked by the navigation filter because of its low elevation angle. The area between the two orange dashed lines shows when the feared events occurs. As recorded, six of the 14 simulated satellites have pseudo-range errors with a constant amplitude of 50 meters. At the beginning of the feared events and shortly after the feared events, when the receiver observed a rapid change in the pseudoranges, it stops transmitting those suspicious signals. As a result, the recorded feared events are identified correctly using RSIVIA, which is showed in green. In this experiment, missed detection or false identification does not occur, with a proper choice of the maximum acceptable size of sub-pavings, which is 5 meters.

Conclusion

The present work applies RSIVIA in a GNSS-based navigation system for the fault detection and identification, which concentrates on a more reliable identification of multi-faults in GNSS signals. The approach was validated in a test campaign with a GNSS simulator. The result shows that, all six satellites with feared events was successfully identified by RSIVIA, which makes it a promising alternative for conventional integrity methods like RAIM.

The present extended abstract only provides the first results of the on-going work, which concentrates on comparing RSIVIA with conventional RAIM and research a concept to combine them both.

Acknowledgement

The joint research project GALILEOnautic 2 is supported by the German Federal Ministry for Economic Affairs and Energy (grant 50NA1808). Basis for the support is a decision by the German Bundestag.

References

- [1] European GNSS Agency, *GNSS market report*. 2017.
- [2] S. Liu, J.-J. Gehrt, D. Abel, and R. Zweigel, “Dual-constellation aided high integrity and high accuracy navigation filter for maritime applications,” *Proceeding of the 2019 International Technical Meeting of The Institute of Navigation (ION ITM)*, pp. 762–774, 2019.
- [3] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*. Springer, 2001.
- [4] L. Jaulin, M. Kieffer, E. Walter, and D. Meizel, “Guaranteed robust nonlinear estimation with application to robot localization,” *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 32, no. 4, pp. 374–381, 2002.
- [5] V. Drevelle and P. Bonnifait, “A set-membership approach for high integrity height-aided satellite positioning,” *J. of GPS Solutions*, vol. 15, no. 4, pp. 357–368, 2011.
- [6] J.-J. Gehrt, R. Zweigel, T. Konrad, and D. Abel, “Dvl-aided navigation filter for maritime applications,” *11th IFAC Conference on Control Applications in Marine Systems, Robotics and Vehicles (IFAC CAMS 2018)*, pp. 418–423, 2018.

Interval based Fault Detection and Exclusion for GNSS

Hani Dbouk*¹ and Steffen Schön²

^{1,2}Institut für Erdmessung, Leibniz Universität Hannover,
dbouk,schoen@ife.uni-hannover.de

Keywords: Intervals; Zonotope; Polytope; GNSS

Introduction

Guaranteed protection levels of the Global Navigation Satellite System (GNSS) are of great importance, especially for the safety critical application such as: landing approach and navigation of autonomous vehicles. In order to guarantee the computed protection levels, reliable outlier detection and exclusion algorithms must be apply. In the past 30 years, different algorithms have been investigated based on statistical hypothesis testing. Thanks to their out-performance, the residual based test statistics and the solution separation have gained most interest compared to other algorithms. However, statistic-based fault detection and exclusion algorithms do not guarantee a safe navigation when the underlying assumptions on error probability density functions may not be fulfilled. In addition, interval-based fault detection techniques have been investigated in literature *e.g.*: q-relax intersection. This technique guarantees the protection level but it suffers from low accuracy in the multiple fault situation.

In this work, we propose a fault detection and exclusion technique based on deterministic observation intervals. The inconsistency of the positioning problem is indicated by the size of the polytope obtained from the intersection of the observation intervals. For the optimal case of no observation noise and no outliers, the polytope is a zonotope. Thus, we will use the normalized relative volume between a nominal polytope (zonotope) and the

actual non-regular polytope as indicator to derive an outlier detection.

The observation intervals are determined from sensitivity analysis of the correction models and the expert knowledge of the size of remaining errors. Applying the interval bounds on both direction of the observation transforms the navigation problem from a single point position to a solution set represented by non-regular polytope. If the observation intervals contain the actual observations, then the solution set guarantees to contain the true position. If biases occur for some observations, different situations can happen. For large biases, the solution set is empty which indicates the bias and can serve as detection criteria. When small biases occur, the solution set is not guaranteed and also not empty. In order to detect those type of biases, a threshold is proposed and applied on the relative volume which indicate the level of inconsistency between the intervals and the actual observations. Monte Carlo simulations are performed on different GNSS positioning scenarios for a better understanding of the inconsistency behavior.

Methodology

The non-linear GNSS navigation equation Eq. (1), is linearized via Taylor expansion at an approximate initial position. As a result, we get the system of equations represented in Eq. (2). Then applying the interval bounds on the observed minus computed values, we get a system of inequalities Eq. (3) which can be interpreted as a hyperplan representation of a polytope.

*Corresponding author.

$$l = \sqrt{(x_{sv} - x_u)^2 + (y_{sv} - y_u)^2 + (z_{sv} - z_u)^2} + c \cdot (dt_u - dt_{sv}) \quad (1)$$

$$\mathbf{A}d\hat{\mathbf{x}} = d\mathbf{l} \quad (2)$$

$$d\mathbf{l} - \Delta \leq \mathbf{A}d\hat{\mathbf{x}} \leq d\mathbf{l} + \Delta \quad (3)$$

where sv indicates the space vehicle and u the user, l is the pseudorange measurements, cdt_u and cdt_{sv} the receiver and satellite clock offset, respectively, \mathbf{A} the design matrix, $d\hat{\mathbf{x}}$ the estimated state vector, and Δ the interval error bound of the observations. Then a primal dual polytope algorithm is used to transform the hyperplane representation into a vertex representation of the polytope.

The shape, volume and position of the polytope depend on the observation errors, interval bounds, and satellites geometry. As the observation errors increase, the volume of the polytope decreases till it becomes empty for large outliers *i.e.*: the true observation is outside the interval bounds. To measure this inconsistency in the observations, the nominal polytope (zonotope) is computed and compared to the regular polytope, Eq. (4).

$$Inconsistency = V_r = \frac{V_Z - V_P}{V_Z} \quad (4)$$

Results and Discussion

We perform a Monte Carlo simulation to understand the behavior of the inconsistency measures in terms of geometry, Δ and biases. Fig. 1 shows the results of the simulation, where 7 different scenarios with different number of satellites in view and geometrical dilution of precision GDOP (Table 1) have been analyzed. 1000 epochs have been simulated for each run, where a ramp bias is introduced starting from epoch 100 and ending at epoch 500. We simulate GPS code measurements with white noise ($0, \sigma = 1m$) and a clock error with linear drift and white noise ($0, 1m$).

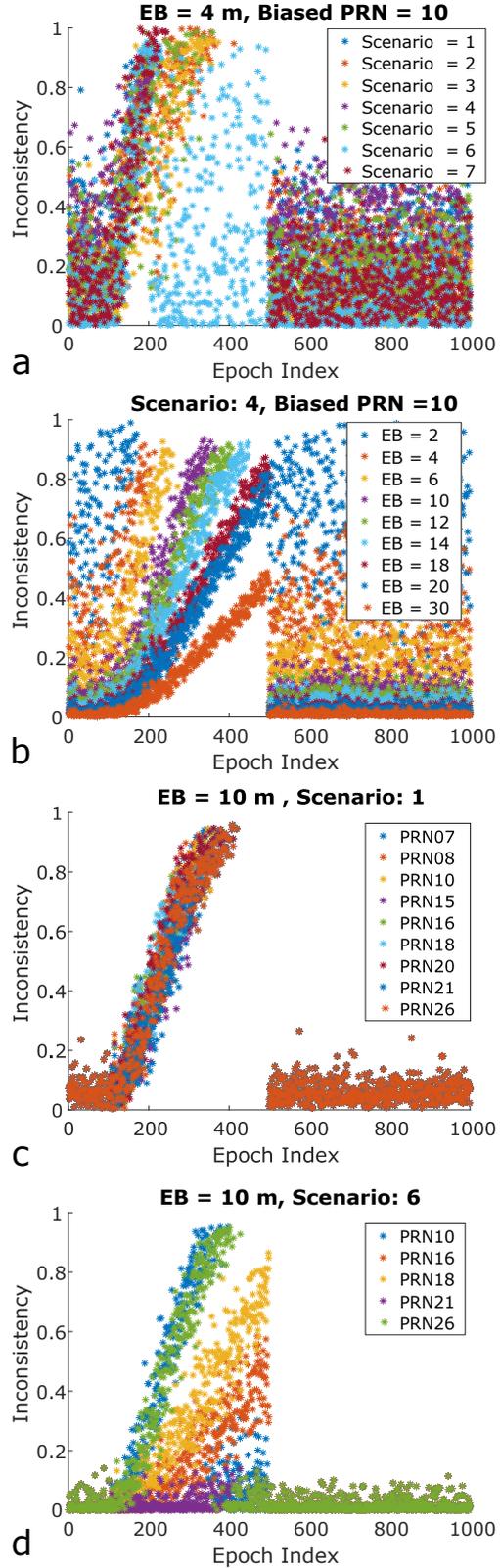


Figure 1: Inconsistency measures: a. different scenarios, b. different applied error bounds, c and d. different biased measurement.

Scen	1	2	3	4	5	6	7
N_{SV}	9	8	6	10	7	5	6
GDOP	2.3	2.4	3.3	2.0	3.3	11	9

Table 1: GDOP and number of satellites for each scenario in the Monte Carlo simulation.

Fig. 1.a shows the results when the Δ are fixed to 4 meters and the biased satellite is the same for all scenarios. It is clear that different scenarios behave in a different way to the same biased satellite. In figure 1.b the scenario (4) and the biased satellite (PRN 10) is fixed while varying the error bounds Δ . The inconsistency behave in different way with and without biases. As the error bound increases the mean value of the inconsistency decreases and the slope of the inconsistency in the biased region decreases and number of empty sets decreases.

Fig. 1.c and 1.d display the different biased satellites where the geometry and the Δ were kept fixed. Fig. 1.c depicts a good geometry situation (scenario 1; GDOP = 2.3) and reveals the same effect for different biased satellites, while Fig. 1.d depicts the bad geometry (scenario 6; GDOP = 11) and implies different behavior of the inconsistency for different biased satellites. This behavior depends on the line-of-sight direction of the biased satellites and the geometry of the other satellites in view. A good explanation and demonstration is presented in [1].

The Monte Carlo simulation reveals the complexity of the inconsistency measures derived from the relative volume of polytopes. For simplicity and to test the algorithm on a real data, we apply a simple threshold test based on the mean value and the standard deviation of the relative volume. Fig. 2 shows the cumulative frequency of the 3D position error with and without applying the fault detection and exclusion on the inconsistency. The test shows a 25.6 % improvement in position where the root mean square of the po-

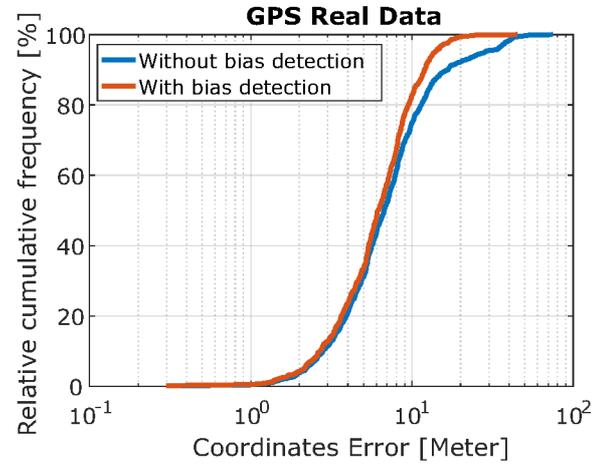


Figure 2: Cumulative frequency of the 3D coordinates error.

sition error decreases from 9 m to 6.7 m.

Conclusions

In this study, a new fault detection and exclusion method is developed and tested with simulated and real data. First results show around 25 % improvement in the positioning error with simple threshold test. However, the behavior of the inconsistency measures is more complicated and varies from situation to another as the Monte Carlo simulation suggest. In the future more sophisticated test will be investigated.

Acknowledgement

This work was supported by the German Research Foundation (DFG) as a part of the Research Training Group i.c.sens [GRK2159].

References

- [1] H. Dbouk and S. Schön. Reliability and integrity measures of gps positioning via geometrical constraints. pages 730–743. Proceedings of the 2019 International Technical Meeting of The Institute of Navigation, Jan. 2019.

Extrinsic Calibration Between a 3D Laser Scanner and a Camera Under Interval Uncertainty

Raphael Voges* and Bernardo Wagner

Leibniz Universität Hannover, Real Time Systems Group (RTS)
 Appelstraße 9A, D-30167 Hannover, Germany
 {voges,wagner}@rts.uni-hannover.de

Keywords: Extrinsic calibration; Camera; 3D LiDAR; Interval analysis; Contractors

Introduction

For navigation in the absence of GPS, mobile robots often fuse information from both laser scanner and camera to take advantage of both sensor modalities [1]. While the laser scanner allows to measure accurate distances to the environment, camera images can be employed to re-identify salient features over time and space. However, to fuse data from both sensors, the extrinsic transformation - i.e. the rotation and translation between the sensor coordinate systems - has to be known.

Generally, all approaches for the extrinsic calibration between camera and laser scanner can be divided into two categories: target-based (e.g. using a checkerboard) or target-less (by relying on natural image features). Since the target-less extrinsic calibration is usually less accurate due to the problem of accurately identifying the same features in both laser scan and camera data, we omit it and focus on the target-based calibration. Unnikrishnan and Hebert employ a checkerboard for which they extract the plane parameters from the data of both sensors [2]. Since their approach requires corresponding plane parameters from at least three different checkerboard poses, Zhou et al. aim to reduce this number by incorporating additional features into their non-linear optimization [3]. Instead of only extracting the checkerboard's plane, they

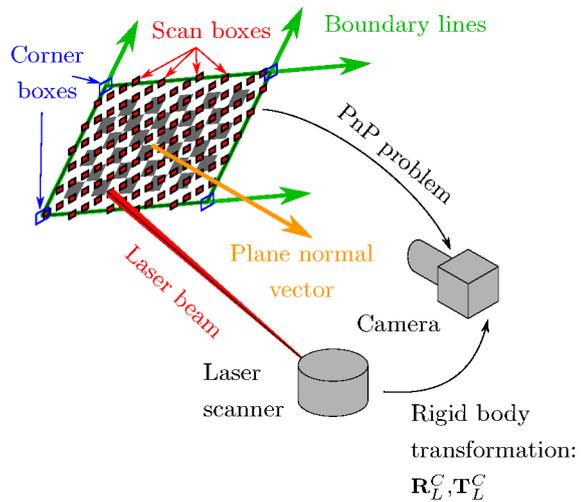


Figure 1: To find the rigid body transformation between camera and laser scanner, consisting of the rotation \mathbf{R}_L^C and the translation \mathbf{T}_L^C , we extract plane, line and point features from the data of both sensors.

also identify the border lines in both the camera image and the laser scan. Consequently, the authors integrate these line features as additional constraints, and can thus reduce the number of required checkerboard poses while also increasing the calibration accuracy.

However, both presented approaches neglect the underlying uncertainties and their accuracies can only be assessed by comparing the resulting transformation to ground truth information. This poses a problem since ground truth is generally not available and the results from Zhou et al. show that the accuracy varies significantly with the number and diversity of chosen checkerboard poses [3].

Furthermore, both approaches assume zero-

*Corresponding author.

mean noise for the sensors. However, prior to the extrinsic laser-camera calibration, an intrinsic camera calibration needs to be performed. Here, imperfections leading to biased intrinsic camera parameters can occur. Subsequently, these parameters are employed to establish a connection between image pixels and real world coordinates, resulting in biased (non-zero-mean) features for the following extrinsic calibration. Likewise, the laser scanner's distance and angular measurements can be biased due to an imperfect calibration, leading to systematic errors for the extracted plane and line features.

Thus, we propose to assign an unknown but bounded error to the sensor measurements and use interval analysis to propagate the error from input sources to the final calibration result. On the one hand, this allows us to model unknown systematic errors for both sensors. On the other hand, we can immediately assess the extrinsic calibration accuracies by inspecting the corresponding interval widths. Similar work - although for a different application - has been done by Sandretto et al., who introduce a method to calibrate a cable-driven robot [4].

Extrinsic Calibration

We extract the same plane and line features as proposed by Zhou et al. [3], but constrain the rigid body transformation even further by computing the checkerboard's corner points. Figure 1 shows the general idea. All computations are performed in a bounded-error context, meaning that we start by modeling the sensor errors with intervals and extract interval domains for the desired features.

To identify the aforementioned features in the camera image, we solve the Perspective-n-Point problem under interval uncertainty [5]. This allows us to establish a connection between the camera coordinate system and the checkerboard coordinate system. By taking advantage of the fact that the dimensions of

the checkerboard are known, we can immediately compute a box enclosing the corner points and derive the boundary lines as well as the plane parameters accordingly.

To find the plane parameters in the laser scan data, we force the corresponding interval boxes on a common plane by employing a forward-backward contractor in combination with branching. Subsequently, we find boundary points residing on the checkerboard's border and fit a line through them to determine the boundary lines. Afterwards, a box enclosing the corner points can be computed by intersecting adjacent boundary lines.

In the following, we introduce the variables required to establish the constraint satisfaction problem (CSP). Generally, a right superscript C or L indicates that the particular feature is given in the camera or laser scanner coordinate system, respectively.

- \mathbf{n}^L and \mathbf{n}^C are the unit checkerboard plane normal vectors.
- \mathbf{d}_i^L and \mathbf{d}_i^C are unit direction vectors describing the same checkerboard boundary line $i \in \{1, \dots, 4\}$.
- \mathbf{Q}_{ij}^L and \mathbf{Q}_{ik}^C are points on the line i with $j \in \{1, \dots, N_i\}$ and $k \in \{1, 2\}$. N_i is the total number of points on the line i which we extract from laser scan data. In contrast, we determine only two points on every line i for the camera - namely the two adjacent corner points.
- \mathbf{P}_l^L are scan points on the checkerboard with $l \in \{1, \dots, N_p\}$. N_p is the total number of scan points on the plane.
- d^C is the distance from the camera coordinate system's origin to the plane.
- \mathbf{C}_m^L and \mathbf{C}_m^C are corresponding checkerboard corner points, $m \in \{1, \dots, 4\}$.

Finally, we are able to formulate the CSP that employs the extracted features to constrain the rigid body transformation, which consists of the rotation matrix \mathbf{R}_L^C and the translation vector \mathbf{T}_L^C .

x	ϕ_L^C (°)	θ_L^C (°)	ψ_L^C (°)	xT_L^C (cm)	yT_L^C (cm)	zT_L^C (cm)
x^*	90.0	0.0	0.0	-27.0	15.0	-12.0
$[x]$	[89.6, 90.3]	[-0.4, 0.3]	[-0.1, 0.3]	[-28.8, -25.0]	[13.1, 16.7]	[-13.1, -11.0]
$w([x])$	0.7	0.7	0.4	3.8	3.6	2.1

Table 1: Results from simulation. The rotation matrix \mathbf{R}_L^C is expressed using the three Euler angles θ_L^C , ψ_L^C and ϕ_L^C . Besides, $\mathbf{T}_L^C = ({}_xT_L^C \quad {}_yT_L^C \quad {}_zT_L^C)^\top$. We depict the true transformation parameters x^* , the computed intervals $[x]$ and the corresponding interval widths $w([x])$.

Variables: $\mathbf{R}_L^C, \mathbf{T}_L^C, \mathbf{n}^L, \mathbf{n}^C, \mathbf{d}_i^L, \mathbf{d}_i^C,$
 $\mathbf{Q}_{ij}^L, \mathbf{Q}_{ik}^C, \mathbf{P}_l^L, d^C, \mathbf{C}_m^L, \mathbf{C}_m^C$

Constraints:

1. $\mathbf{R}_L^C \mathbf{n}^L = \mathbf{n}^C$
2. $\mathbf{R}_L^C \mathbf{d}_i^L = \mathbf{d}_i^C$
3. $(\mathbf{I} - \mathbf{d}_i^C (\mathbf{d}_i^C)^\top) (\mathbf{R}_L^C \mathbf{Q}_{ij}^L + \mathbf{T}_L^C - \mathbf{Q}_{ik}^C) = \mathbf{0}$
4. $\mathbf{n}^C \cdot (\mathbf{R}_L^C \mathbf{P}_l^L + \mathbf{T}_L^C) + d^C = 0$
5. $\mathbf{R}_L^C \mathbf{C}_m^L + \mathbf{T}_L^C = \mathbf{C}_m^C$

Domains: $[\mathbf{R}_L^C], [\mathbf{T}_L^C], [\mathbf{n}^L], [\mathbf{n}^C], [\mathbf{d}_i^L], [\mathbf{d}_i^C],$
 $[\mathbf{Q}_{ij}^L], [\mathbf{Q}_{ik}^C], [\mathbf{P}_l^L], [d^C], [\mathbf{C}_m^L], [\mathbf{C}_m^C]$

To solve the CSP, we build a forward-backward contractor for all constraints. In principle, one corresponding laser scan and camera image suffices to compute the transformation. However, by combining the contractors built from different checkerboard poses, the accuracy can be increased.

Results

Our results for simulated data show that we are able to reliably enclose the true transformation parameters for different transformations. The results indicate that our method can cope with outliers by employing a q-relaxed intersection and performs accurately for different error intervals. Table 1 depicts exemplary results.

Moreover, we collected data using a typical laser scanner and camera setup to demonstrate the applicability of our approach to real data. The resulting intervals are consistent

with the parameters computed using the approach of Zhou et al. [3]. Unlike their method, however, our approach allows a direct assessment of the accuracy.

Acknowledgement

This work was supported by the German Research Foundation (DFG) as part of the Research Training Group i.c.sens [RTG 2159].

References

- [1] R. Voges, C. S. Wiegardt, and B. Wagner, “Finding timestamp offsets for a multi-sensor system using sensor observations”, *Photogrammetric Engineering & Remote Sensing*, vol. 84, no. 6, pp. 357–366, 2018.
- [2] R. Unnikrishnan and M. Hebert, “Fast Extrinsic Calibration of a Laser Rangefinder to a Camera”, 2005.
- [3] L. Zhou, Z. Li, and M. Kaess, “Automatic Extrinsic Calibration of a Camera and a 3D LiDAR Using Line and Plane Correspondences”, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018.
- [4] J. A. D. Sandretto, G. Trombettoni, D. Daney, and G. Chabert, “Certified Calibration of a Cable-Driven Robot Using Interval Contractor Programming”, in *Computational Kinematics*, Springer Netherlands, Oct. 2013, pp. 209–217.
- [5] R. Voges and B. Wagner, “Timestamp Offset Calibration for an IMU-Camera System Under Interval Uncertainty”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018.

Toward the Development of Iteration Procedures for the Interval-Based Simulation of Fractional-Order Systems

Andreas Rauh^{*1} and Julia Kersten¹

¹ Chair of Mechatronics, University of Rostock, Justus-von-Liebig-Weg 6
D-18059 Rostock, Germany, {Andreas.Rauh, Julia.Kersten}@uni-rostock.de

Keywords: Exponential enclosure techniques; Interval analysis; Fractional-order systems; Initial value problems (IVPs)

Introduction

In recent years, numerous interval-based simulation techniques have been developed which allow for a verified computation of outer interval enclosures for the sets of reachable states of dynamic systems represented by finite-dimensional sets of ordinary differential equations (ODEs). Here, especially the evaluation of IVPs is of interest, when both the systems' initial conditions and parameters can only be defined by finitely large domains, often represented by interval boxes. Suitable simulation techniques make use of series expansions of the solutions of IVPs with respect to time and (possibly) the uncertain initial conditions as well as of verified Runge-Kutta techniques. Solution sets are then typically represented by means of multi-dimensional intervals, zonotopes, ellipsoids, or Taylor models, cf. [5].

In most of these approaches, variants of the Picard iteration [1] are involved, which either determine the sets of possible solutions or at least worst-case outer enclosures with which time discretization errors are quantified. An example for a solution routine based entirely on this iteration is the exponential enclosure technique published in [9] and the references therein. It is applicable to systems with non-oscillatory and oscillatory behavior if the solution of the IVP of interest shows an asymptotically stable behavior. For non-oscillatory

dynamics, the solution is determined by a real-valued iteration, while complex-valued interval analysis [7] is employed when eigenvalues with non-zero imaginary parts arise after a linearization of the state equations.

Although such enclosure techniques are well studied for IVPs of integer-order ODEs, the analysis of fractional-order differential equations (FDEs) has not yet received the same amount of interest. FDEs can be used efficiently in many engineering applications if the frequency response of a dynamic system is not characterized by variations of the amplitudes that consist of multiples of the slope ± 20 dB per frequency decade. The same holds for changes of the phase responses which do not coincide with integer multiples of $\pm \frac{\pi}{2}$, cf. [6,8]. In such cases, FDEs (for real-life applications often of the so-called Caputo type) can be used to significantly enhance modeling accuracy in comparison with integer-order ODEs. First extensions of the Picard iteration for determining interval enclosures to IVPs for FDEs were published in [4]. Practical applications where FDEs have significant advantages over classical ODEs can be found exemplarily in the field of modeling and state estimation for battery systems [10].

In the following, a brief summary of an extension of the exponential enclosure technique for FDEs published in [9] is given.

Interval Methods for FDEs

Consider the commensurate-order FDE system of Caputo type [6,8]

$$\mathbf{x}^{(\nu)}(t) = \mathbf{A}(\mathbf{x}(t)) \cdot \mathbf{x}(t), \quad \mathbf{x}(t) \in \mathbb{R}^n, \quad (1)$$

^{*}Corresponding author.

with $0 < \nu < 1$, where initial conditions for $\mathbf{x}(t)$ at $t = 0$ are defined in analogy to classical IVPs for integer-order ODEs. Then, for the iteration step κ , (complex-valued) parameters $\lambda_i^{(\kappa)}$ with $[\mathbf{\Lambda}]^{(\kappa)} := \text{diag} \{ [\lambda_i]^{(\kappa)} \}$, $i \in \{1, \dots, n\}$, are determined in the ansatz

$$\mathbf{x}(t) \in [\mathbf{x}_e]^{(\kappa)}(t) := \mathbf{E}_{\nu,1}([\mathbf{\Lambda}]^{(\kappa)} \cdot t^\nu) \cdot [\mathbf{x}_e](0), \quad (2)$$

where $\mathbf{E}_{\nu,1}([\mathbf{\Lambda}]^{(\kappa)} \cdot t^\nu)$ is a diagonal matrix with the element-wise evaluation of the Mittag-Leffler function [3] $E_{\alpha,\beta}(\zeta) = \sum_{i=0}^{\infty} \frac{\zeta^i}{\Gamma(\alpha i + \beta)}$. According to [9], the parameters of the enclosures (2) are computed by

$$\begin{aligned} [\lambda_i]^{(\kappa+1)} &:= a_{ii}([\mathbf{x}_e]^{(\kappa)}([t])) \\ &+ \sum_{\substack{j=1 \\ i \neq j}}^n \left\{ a_{ij}([\mathbf{x}_e]^{(\kappa)}([t])) \right. \\ &\left. \cdot \frac{E_{\nu,1}([\lambda_j]^{(\kappa)} \cdot [t]^\nu)}{E_{\nu,1}([\lambda_i]^{(\kappa)} \cdot [t]^\nu)} \cdot \frac{[x_{e,j}](0)}{[x_{e,i}](0)} \right\}. \end{aligned} \quad (3)$$

In this contribution, simulation results for verified IVP solutions for FDEs are presented in comparison with corresponding analytic solutions given, for example, in [2].

Finally, current research directions will be pointed out together with possible solution approaches for yet open problems resulting from the fact that FDE models represent memory effects over infinitely long time horizons.

References

- [1] Y. Deville, M. Janssen, and P. van Henenryck. Consistency Techniques for Ordinary Differential Equations. *Constraint*, 7(3–4):289–315, 2002.
- [2] K. Dorjgotov, H. Ochiai, and U. Zunderiya. On Solutions of Linear Fractional Differential Equations and Systems Theoref. 2018. arXiv:1803.09063.
- [3] H. J. Haubold, A. M. Mathai, and R. K. Saxena. Mittag-Leffler Functions and Their Applications. *Journal of Applied Mathematics*, 2011:51 pages, 2011. <https://doi.org/10.1155/2011/298628>.
- [4] R. Lyons, A. Vatsala, and R. Chiquet. Picard’s Iterative Method for Caputo Fractional Differential Equations with Numerical Results. *Mathematics*, 5(4), 2017.
- [5] N. S. Nedialkov. Interval Tools for ODEs and DAEs. In *CD-Proc. of 12th GAMM-IMACS Intl. Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN 2006*, Duisburg, Germany, 2007. IEEE Computer Society.
- [6] A. Oustaloup. *La Dérivation Non Entière: Théorie, Synthèse et Applications*. Hermès, Paris, 1995. In French.
- [7] M. Petković and L. Petković. *Complex Interval Arithmetic and Its Applications*. Wiley-VCH Verlag GmbH, Berlin, 1998.
- [8] I. Podlubny. *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*. Mathematics in Science and Engineering. Academic Press, London, 1999.
- [9] A. Rauh, J. Kersten, and H. Aschemann. Techniques for Verified Reachability Analysis of Quasi-Linear Continuous-Time Systems. In *Proc. of 24th Intl. Conference on Methods and Models in Automation and Robotics*, 2019.
- [10] B. Wang, Z. Liu, S. Li, S. Moura, and H. Peng. State-of-Charge Estimation for Lithium-Ion Batteries Based on a Nonlinear Fractional Model. *IEEE Trans. on Control Systems Technology*, 25(1):3–11, 2017.

Confidence-based Contractor, Propagation and Potential Cloud for Differential Equations

Julien Alexandre dit Sandretto*¹

¹ENSTA ParisTech, Palaiseau, France

Keywords: Interval Analysis, Confidence Level, Potential Cloud, Reachability for ODEs

Introduction

An interval aims to bound all the values of an uncertain variable, for example provided by a measurement device [5]. This approach is highly effective for every safety, verification or validation procedures because the intervals are conservative. The major inconvenience is that intervals are sometimes too pessimistic. Otherwise, it is conceivable that a measure can be associated to guaranteed bounds, an interval, and also a confidence level coming from past observations.

A novel contractor is proposed to filter an interval following a confidence level given on the associated quantity. This confidence level is an input of the contractor, the “new” information, while the probability distribution of the considered variable is a characteristic of the associated random variable.

Combining intervals and a probability has been already proposed in numerous papers using techniques such as p-boxes [8], fuzzy sets [4], box-particles [1] and potential cloud [7].

We are particularly interested in Ordinary Differential Equations (ODEs) and validated methods to compute their reachable sets via validated simulation [6, 3]. In the case of Initial Value Problems (IVPs) with ODEs, the initial state is primordial. An uncertain initial

state is generally bounded in a box. As experimentation, we propose to consider in addition to this initial box some confidence levels, and we apply the presented approach. It allows us to describe the reachable set by a cloud. This richer result can then be used in different control problems, parameter synthesis, verification, etc.

Preliminaries

When focusing on symmetric distributions such as the normal distribution, one can define:

Confidence interval is a set \mathcal{S} for which the probability of the given random variable to be in this set is equal to the given probability P .

Probability density function is most commonly associated with absolutely continuous univariate distributions. A random variable X has density f_X , where f_X is a non-negative Lebesgue-integrable function, if:

$$P = \Pr[a \leq X \leq b] = \int_a^b f_X(x) dx.$$

Confidence level (e.g. $CL = 95\%$) allows to define the corresponding confidence interval (e.g. $C_{95\%}$). This interval can be obtained by observation (statistical approach) or with the help of a known distribution (probability approach). A new

*Corresponding author.

measure \hat{x} coming from the (same) experiment will be in the associated confidence interval such that:

$$\hat{x} \in C_{95\%} \quad 95\% \text{ of the time.}$$

Confidence-based Contractor

We propose the following confidence-based contractor:

$$\begin{aligned} Cbc([x]|f_X, cc) : \mathbb{R} &\mapsto \mathbb{R} \\ [x] &\rightarrow [x] \cap [y] \end{aligned}$$

with $[y]$ the confidence interval defined such that

$$\Pr[x \in [y]] = \int_{[y]} f_X(x) dx = cc,$$

cc being the confidence coefficient ($0 \leq cc \leq 1$). For example, one can use the parameter assignment $cc = 0.68$ for a confidence level of 68%.

Figure 1 illustrates the effect of the confidence-based contractor applied to the following example. Let X be a random variable with a normal distribution, such that

$$f_X(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

with $\mu = 1.0$ and $\sigma = 1.0$. The quantity X is observed and one measure is obtained: $[x] = [0.7, 2.1]$. A confidence level of 68.27% is given on X , that is to say that we are confident on the accuracy of the observations, so X stays close to its mean. Our method computes the contraction such that:

$$\begin{aligned} Cbc([0.7, 2.1] | f_X, 0.6827) &= [0.7, 2.1] \cap [0, 2] \\ &= [0.7, 2.0] \end{aligned}$$

So upper bound is reduced with respect to the confidence level. The pessimism induced by interval approach is then limited.

Two special cases can be described:

- $\forall [x], Cbc([x]|f_X, 0) = \emptyset$ (annihilating element)

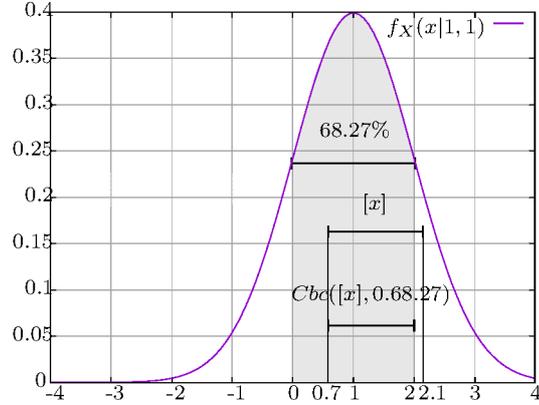


Figure 1: Illustration of confidence-based contraction.

- $\forall [x], Cbc([x]|f_X, 1) = [x]$ (identity element)

For two different confidence coefficients cc_1 and cc_2 such that $cc_1 < cc_2$, the following order holds:

$$\forall [x], Cbc([x]|f_X, cc_1) \subset Cbc([x]|f_X, cc_2)$$

The contractor Cbc can be composed with other contractors or with itself.

The confidence-based contractor presented in this paper needs the computation of the confidence interval associated to a given confidence level. Three cases can be detailed:

Case 1: a well known probability distribution and a particular confidence level with known confidence interval. For example, a normal distribution with a 95% confidence level gives a confidence interval $[\mu - 2\sigma, \mu + 2\sigma]$.

Case 2: a probability distribution with a known inverse function, such as the inverse of error function for Gaussian density function (*i.e.* erf^{-1}).

Case 3: the general case without any particular value. For this case, a predictor-corrector algorithm exploiting the symmetry of the distribution is designed.

Reachability and Potential Clouds

Computing the reachable set of an initial value problem defined as follows:

$$\begin{cases} \dot{\mathbf{y}}(t) = g(t, \mathbf{y}(t)) \\ \mathbf{y}(0) \in [\mathbf{y}_0] \subseteq \mathbb{R}^n. \end{cases} \quad (1)$$

can be performed with validated simulation tools. It provides a tube enclosing $\mathbf{y}(t; [\mathbf{y}_0])$ for $t \in [0, T]$. The confidence contractor presented here can be used to filter the initial states w.r.t. different confidence levels. The contraction can be propagated along the tube[2] to build a potential cloud on the final state. This approach is useful in the case of differential constraint while a constraint can be verified for some confidence levels but not for the global initial set ($\mathbf{y}(0) \in [0, 2]$). An example is given in Fig. 2. With a confidence level of 50% on initial states, the constraint $[y(5; [y_0])] \subset [1.5, 1.6]$ is verified.

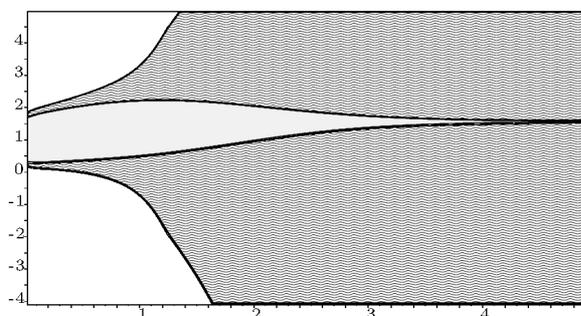


Figure 2: Validated trajectories for 60% (in grey waves) and for 50% (in light grey) confidence contraction.

Conclusion

A novel interval contractor based on the confidence assigned to a random variable was proposed. It makes possible to consider at the same time an interval in which the quantity is guaranteed to be, and a confidence level to reduce the pessimism induced by interval approach. As application, we proposed to com-

pute the reachable set of an ordinary differential equation under the form of a potential cloud, with respect to confidence levels on initial value.

References

- [1] F. Abdallah, A. Gning, and P. Bonnifait. Box particle filtering for nonlinear state estimation using interval analysis. *Automatica*, 44(3):807 – 815, 2008.
- [2] J. Alexandre dit Sandretto and A. Chapoutot. Contraction, propagation and bisection on a validated simulation of ODE. In *Small Workshop on Interval Methods*, 2016.
- [3] J. Alexandre dit Sandretto and A. Chapoutot. Validated explicit and implicit Runge-Kutta methods. *Reliable Computing*, 22:79–103, 2016.
- [4] D. Dubois, E. Kerre, R. Mesiar, and H. Prade. *Fuzzy Interval Analysis*, pages 483–581. Springer US, Boston, MA, 2000.
- [5] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer, 2001.
- [6] R. J. Lohner. Enclosing the solutions of ordinary initial and boundary value problems. *Computer Arithmetic*, page 255–286, 1987.
- [7] A. Neumaier. Clouds, fuzzy sets, and probability intervals. *Reliable computing*, 10(4):249–272, 2004.
- [8] R. C. Williamson and T. Downs. Probabilistic arithmetic. i. numerical methods for calculating convolutions and dependency bounds. *International Journal of Approximate Reasoning*, 4(2):89 – 158, 1990.

Bounded-error victim localization for UAV-based search and rescue operations

Vincent Drevelle*

Univ Rennes, IRISA, Inria, CNRS

Keywords: Localization, Drone, Search and Rescue, Vision, Digital Elevation Model, Intervals

Introduction

Small unmanned aerial vehicles (UAVs), are currently used by firefighters teams for search and rescue operations, leveraging the use of thermal imaging and zoom cameras for finding and identifying lost people. The victim is detected and tracked in the image, but the rescue team on the ground needs georeferenced coordinates to intervene. The classical operational way is to describe the victims position relative to surrounding landmarks, and to fly the drone above the victim to take down its geographical coordinates (latitude and longitude).

UAVs are equipped with GPS and inertial navigation systems, which enables to know their position and orientation in a geographical reference frame. Assuming that the onboard camera calibration parameters are known, it is possible to cast each image measurement into a *ray* in the real world. Locating a victim can thus be done either by intersecting the rays obtained from two different views, or by intersecting the ray from a single observation with the ground surface obtained from a digital elevation model (DEM).

Interval methods have successfully been used for vision-based localization [2] or reconstruction [3], and also for DEM-aided positioning [1]. Assuming uncertain camera calibration and drone pose (position and orientation), this work aims to compute a bounding

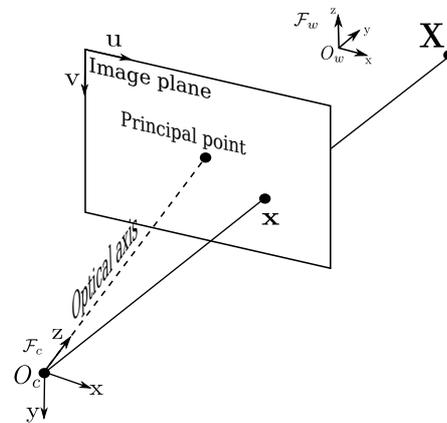


Figure 1: Pinhole camera model

domain of the possible position of a victim localized in the drone image. Localization from a single measurement is enabled by using a DEM, whose accuracy is also taken into account.

Camera measurement

Image observations are described by the pinhole camera model (Fig. 1). A point in the image plane corresponds to a ray in the world. The camera ray uncertainty can be divided into two components: origin and direction.

A first source of ray direction error is expressed in the camera image plane (in pixels). It is related to pointing/tracking accuracy. The camera intrinsic parameters ma-

*Corresponding author. vincent.drevelle@irisa.fr

trix \mathbf{K} (obtained from calibration) enables to relate the measured image plane coordinates $\bar{\mathbf{x}} = (u, v, 1)^T$ to the ray coordinates $\mathbf{x} = (x, y, 1)^T$ in the normalized space:

$$\mathbf{x} = \mathbf{K}^{-1}\bar{\mathbf{x}}, \quad \mathbf{K} = \begin{pmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

The second source of ray direction error is the camera orientation ${}^c\mathbf{q}$ in the global frame. The pitch and roll components of the error are generally small (a few tenths of a degree if the camera gimbal is well calibrated and the drone is not accelerating). The yaw component can be subject to larger errors (in the order of a few degrees), since it is estimated by a magnetic compass onboard the drone.

The ray origin corresponds to the camera position ${}^c\mathbf{p}$, measured by the GPS of the UAV. The width of the position error domain can vary from several meters if using standalone GPS, to tenths of centimeters if using differential techniques (DGPS, RTK).

The constraint from a single image observation defines a domain corresponding to the “uncertain direction ray” (cone) originating from the camera center, and dilated by the uncertainty box of the GPS position. The set of 3-D world points ${}^w\mathbf{X}$ compatible with the image observation is given by:

$$\mathbb{S}_{\text{camera}} = \{ {}^w\mathbf{X} \mid \bar{\mathbf{x}} = \mathbf{K} \mathbf{\Pi} {}^c\mathbf{T}_w({}^c\mathbf{p}, {}^c\mathbf{q}) {}^w\mathbf{X}, \\ \bar{\mathbf{x}} \in ([u], [v])^T, {}^c\mathbf{p} \in [{}^c\mathbf{p}], {}^c\mathbf{q} \in [{}^c\mathbf{q}] \}$$

where ${}^c\mathbf{T}_w$ is the rigid transform from the world reference frame to the camera frame, and $\mathbf{\Pi}$ is the perspective projection. The intervals $[u], [v]$ represent bounded-error measurements in the image plane. The boxes $[{}^c\mathbf{p}]$ and $[{}^c\mathbf{q}]$ are respectively the position and orientation uncertainty domains of the camera.

Digital elevation model

The digital elevation model provides a useful additional constraint for locating people, assuming they are on the ground. A DEM is

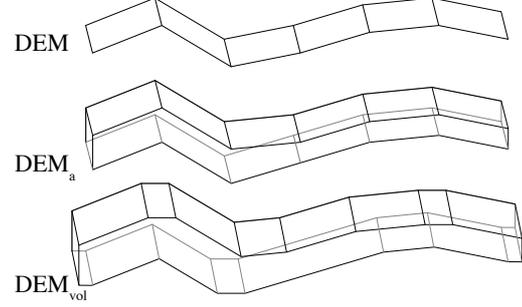


Figure 2: “Thick DEM” construction

classically a regular grid of altitudes. The accuracy of the terrain model has to be taken into account, since it can greatly vary depending on the data source. Particularly, DEM precision tends to be worse in mountain areas, which is where most of the SAR operations occur. The DEM precision is described by two components: altimetric accuracy and planimetric accuracy.

From the ground surface defined by the DEM mesh, we define a “thick DEM” as the domain of the possible locations of the ground surface, taking accuracy figures into account. Thickening the DEM is done in two steps (Fig. 2). Firstly, the punctual altitude measurements are converted to intervals accordingly to altimetric accuracy. This leads to a first volume DEM_a . Then, we compute the Minkowski sum of the obtained domain DEM_a with a square $[\pm e_x; \pm e_y; 0]$ representing the planimetric accuracy, to obtain the final “thick DEM” DEM_{vol} .

Victim localization

Assuming bounded errors, the DEM and the image tracking measurements define sets that are guaranteed to contain the victim’s position. A bounding domain of the victim’s position can thus be obtained by intersecting the “thick DEM” with the “uncertain rays” corresponding to each visual observation.

References

- [1] V. Drevelle and P. Bonnifait. A set-membership approach for high integrity height-aided satellite positioning. *GPS solutions*, 15(4):357–368, 2011. (document)
- [2] I.-F. Kenmogne, V. Drevelle, and E. Marchand. Image-based UAV localization using interval methods. In *IROS 2017 - IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 5285–5291, Vancouver, Canada, Sept. 2017. (document)
- [3] B. Telle, M. J. Aldon, and N. Ramdani. Guaranteed 3d visual sensing based on interval analysis. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1566–1571 vol.2, Oct 2003. (document)

Towards an interval fingerprinting approach for indoor localization.

Nacim Ramdani^{*1}, Demetrios Zeinalipour-Yazti², Michalis Karamousadakis³, and
Andreas Panayides²

¹Univ. Orléans, INSA CVL, PRISME EA 4229, F45072, Orléans, France

²Department of Computer Science, University of Cyprus, 1678 Nicosia, Cyprus

³SingularLogic, Achaias 3 & Troizinias, Nea Kifisia 14564, Greece

Keywords: Indoor, Location, Interval Analysis, Modeling, Hausdorff distance

Introduction

The omni-present availability of sensor-rich smartphones along with the fact that people spend 80-90% of their time in indoor environments has recently boosted an interest around the so called *Internet-based Indoor Navigation (IIN)* [12]. These comprise of indoor models, such as floor-maps and points-of-interest, along with Internet of Things (IoT)-based raw data, such as wireless, light and magnetic signals, used to localize and track mobile users and targets.

There is a large variety of localisation methods that exhibit diverse quality performance levels regarding precision, accuracy, cost, reliability, scalability, energy efficiency and robustness [1, 9]. One reason behind low performance usually observed in localization accuracy or robustness is the noisy nature of the IoT raw data used. For instance, the WiFi received signal strength (RSS), which is most commonly used by indoor localisation techniques, is in fact susceptible to multipath effects and interference, hence shows high variability over time. These variations may naturally introduce errors and jolts in reconstructed locations. To smoothen the location estimates and improve consistency, state-of-the-art localisation techniques work either with averaged signal data, or rely on more ad-

vanced probabilistic or Bayesian approaches [6, 11].

Other approaches use as well hybrid approaches combining RSSi-fingerprinting with inertial tracking systems as in [7] where the WiFi-based and the IMU-based location estimates, along with the associated uncertainties are provided as inputs to a data fusion module that implements the hybridization scheme by means of a particle filter. In practice however, the true probability distribution to use as Likelihood or a priori in the Bayesian methods are often unknown hence need be approximated using Gauss or uniform distributions. It is therefore appealing to consider an alternative description of the errors and disturbances acting on the measurements.

In this note we will report on a preliminary investigation on alternative methods to deal with the uncertainty in the measured signals by working directly with interval data, i.e. data ranges or data sets computed from the raw data with no assumption of the probability distribution within the interval. We will discuss a new method for IIN based on interval fingerprinting [10].

Interval fingerprinting

In this section, we discuss ways to exploit the interval measurements data in Wi-Fi Radiomap-based indoor localization techniques. We focus on techniques such as the ones implemented in Anyplace software [12].

Anyplace uses Wi-Fi Radiomap-based in-

^{*}Corresponding author.

door localization, which stores radio signals from Wi-Fi APs in a database at a high density. The localization subsystem of Anyplace utilizes the following routine:

In an offline phase, a logging application records the so called *Wi-Fi fingerprints*, which comprise of *Received Signal Strength (RSS)* indicators of Wi-Fi Access Points (APs) at certain locations (x, y) pin-pointed on a building floor map (e.g., every few meters).

Subsequently, in a second offline phase, the Wi-Fi fingerprints are joint into a $N \times M$ matrix, coined the *Wi-Fi RadioMap*, where N is the number of unique (x, y) fingerprints and M the total number of APs.

Finally, in the online phase, a user can compare its currently observed RSS fingerprint against the RadioMap in order to find the best match, using known algorithms such as K-nearest neighbour (KNN) or weighted KNN (WKNN) [6].

Contrariwise to standard approaches, we further assume that at each unique location l , $l = 1, \dots, N$, the range of variation of the signal intensity is captured, e.g. by sampling data during short time windows. The *RadioMap (RM)* now contains *interval fingerprint* $[\vec{v}_l]$ measured at location l . The actual coordinates of location l may also be subject to bounded uncertainty, i.e. $\vec{p}_l \in [\vec{p}_l] = ([x_l], [y_l])$. The thus obtained *interval-RM* is stored in a database, where each entry T_l has the form

$$T_l = ([\vec{p}_l]; [\vec{v}_l]) \quad (1)$$

Finally, the observed RSS fingerprint during online phase is taken as an interval vector $[\vec{v}_o]$.

Classification with interval data

Since both the radiomap data and the signal measured by the mobile unit are interval data, we need to extend the WKNN approach to measure dissimilarities between interval vectors.

This idea is not new. A KNN classification method using interval data is proposed in [8],

where the method mainly relies on identifying possible and necessary neighbours using the *partial* orders induced by some distance metrics computed with intervals. By construction, the method yields ambiguous decisions.

To the contrary, other authors addressed the issue using *total* orders for clustering interval data in [4, 2] and also in the framework of fuzzy sets in [3]. In these works, the distance metric used for comparing two interval vectors was the Hausdorff distance, associated with the Chebyshev metric as it seems explicit formulas were readily available for on-line computation. However, when other metrics were used, the distance used was not the most appropriate to interval data. The reason it seems, is that the authors of these works did not find explicit formulas for computing the Hausdorff distance associated with the other metrics.

Actually, Jahn [5] gives explicit formulas that allow online computation of the Hausdorff distance h_p associated with the Minkowski norms (2)

$$d_p(x, y) = \left(\sum_i |x_i - y_i|^p \right)^{1/p}. \quad (2)$$

Classical WKNN can then be used to estimate the location of the mobile unit using the k nearest neighbours

$$[\hat{p}_o] = \frac{\sum_{i=1}^k [\vec{p}_i]/d_i}{\sum_{i=1}^k 1/d_i}, \quad (3)$$

where the distance are given by

$$d_i = h_p([\vec{v}_l], [\vec{v}_o]). \quad (4)$$

Concluding remarks

The preliminary evaluation of our new method on an actual interval radio map containing $N = 52$ interval fingerprints obtained at positions covered by $M = 206$ unique AP, shows that first, it can work directly with interval data, and second, the estimates it provides are smoother and more consistent than the ones provided by state-of-the-art methods.

Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 823887.

H2020 RISE MSCA ENDORSE project
www.endorse-project.eu.

References

- [1] A. Basiri, E. S. Lohan, T. Moore, A. Winstanley, P. Peltola, C. Hill, P. Amirian, and P. F. e Silva. Indoor location based services challenges, requirements and usability of current solutions. *Computer Science Review*, 24:1 – 12, 2017.
- [2] M. Chavent. A Hausdorff distance between hyper-rectangles for clustering interval data. In D. B. D., F. McMorris, P. A. P., and W. Gaul, editors, *Classification, Clustering, and Data Mining Applications*. Springer, Berlin, Heidelberg, 2004.
- [3] F. d. A. de Carvalho and E. C. Simes. Fuzzy clustering of interval-valued data with city-block and hausdorff distances. *Neurocomputing*, 266:659 – 673, 2017.
- [4] R. M. de Souza and F. de A.T. de Carvalho. Clustering of interval data based on city-block distances. *Pattern Recognition Letters*, 25(3):353–365, 2004.
- [5] K.-U. Jahn. Evaluation of hausdorff distances in interval mathematics. *Computing*, 45(1):69–77, Mar 1990.
- [6] B. Li, J. Salter, A. G. Dempster, and C. Rizos. Indoor positioning techniques based on wireless LAN. In *1st IEEE Int Conf on Wireless Broadband and Ultra Wideband Communications*, pages 13–16, 2006.
- [7] C.-L. Li, C. Laoudias, G. Larkou, Y.-K. Tsai, D. Zeinalipour-Yazti, and C. G. Panayiotou. Demo: Indoor geolocation on multi-sensor smartphones. In *Proceedings of the 11th International Conference on Mobile Systems, Applications and Services*, Mobisys'13, pages 503–504, Taipei, Taiwan, June 25 - 28, 2013.
- [8] V.-L. Nguyen, S. Destercke, and M.-H. Masson. K-Nearest Neighbour Classification for Interval-Valued Data. In *11th International Conference on Scalable Uncertainty Management (SUM 2017)*, number 10564 in Lecture Notes in Computer Science, pages 93–106, Oct. 2017.
- [9] G. Oguntala, R. Abd-Alhameed, S. Jones, J. Noras, M. Patwary, and J. Rodriguez. Indoor location identification technologies for real-time iot-based applications: An inclusive survey. *Computer Science Review*, 30:55 – 79, 2018.
- [10] N. Ramdani, D. Zeinalipour-Yazti, M. Karamousadakis, and A. Panayides. Towards robust methods for indoor localization using interval data. In *Proceedings of the 20th IEEE International Conference on Mobile Data Management*, pages 403–408, 2019.
- [11] F. Zafari, A. Gkelias, and K. K. Leung. A survey of indoor localization systems and technologies. *CoRR*, abs/1709.01015, 2017.
- [12] D. Zeinalipour-Yazti and C. Laoudias. The anatomy of the anyplace indoor navigation service. In *ACM SIGSPATIAL Special*, volume 9 of *SIGSPATIAL'17*, pages 3–10. ACM Press, 2017.

Verifying the existence of loops in robot trajectories

Simon Rohou^{*1}, Peter Franek², Clément Aubry³, and Luc Jaulin¹

¹ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France

²IST Austria, Klosterneuburg, Austria

³ISEN Brest, France

Keywords: Mobile Robotics, SLAM, Loop Detection, Interval Analysis, Topological Degree, Tubes

Introduction

We present a reliable method to verify the existence of loops along the uncertain trajectory of a robot, based on proprioceptive measurements only [1], within a bounded-error context.

The loop closure detection is one of the key points in Simultaneous Localization And Mapping (SLAM) methods, especially in homogeneous environments with difficult scenes recognitions.

The approach we propose [3] is fast, reliable and could be coupled with conventional SLAM algorithms to reliably reduce their computing burden, thus improving the localization and mapping processes in the most challenging environments such as unexplored underwater extents.

Loops

An example of loop is given in Figure 1, with a mobile robot that came back at time t_2 to a previous position reached at t_1 . In this work, for a given trajectory, a loop is defined temporally as a 2d vector $\mathbf{t} = (t_1, t_2)^\top$ such that $\mathbf{f}(\mathbf{t}) = \mathbf{0}$ with

$$\mathbf{f}(\mathbf{t}) = \int_{t_1}^{t_2} \mathbf{v}(\tau) d\tau,$$

a function describing robot's move from t_1 to t_2 , based on its absolute velocities $\mathbf{v}(t) \in \mathbb{R}^2$.

^{*}Corresponding author.

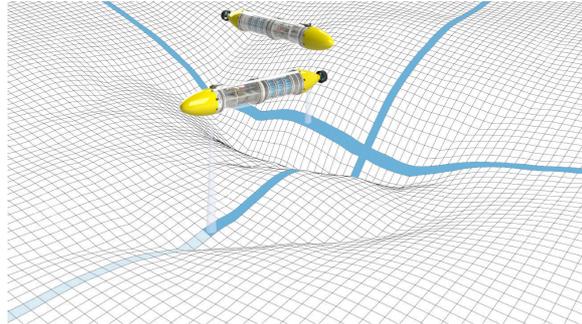


Figure 1: An underwater robot exploring its environment, before and after performing a loop. The robot trajectory is projected in blue on the sea-floor.

Detection vs. verification

A distinction has to be made between the *detection* and the *verification* of a loop \mathbf{t} . Considering a set of feasible trajectories, some of them may cross themselves at some point; this will lead to a *detection*. In addition, when we verify that all the feasible trajectories are looped, then we can speak about a *loop proof* since a loop occurs whatever the considered uncertainties coming from the sensors. Figure 2 provides an illustration of this distinction.

In a reliable context, any feasible trajectory has to be considered, based on the uncertainties coming from the measurements of $\mathbf{v}(t)$. Tubes are used for this purpose.

The set-membership method we propose stands on tubes $[\mathbf{x}](\cdot)$, see Figure 3, that are intervals of trajectories $\mathbf{x}^-(\cdot)$ and $\mathbf{x}^+(\cdot)$ such that $\mathbf{x}^-(t) \leq \mathbf{x}^+(t) \forall t$. Most of the classical mathematics operations we know on intervals can be extended to tubes. In this work, in-

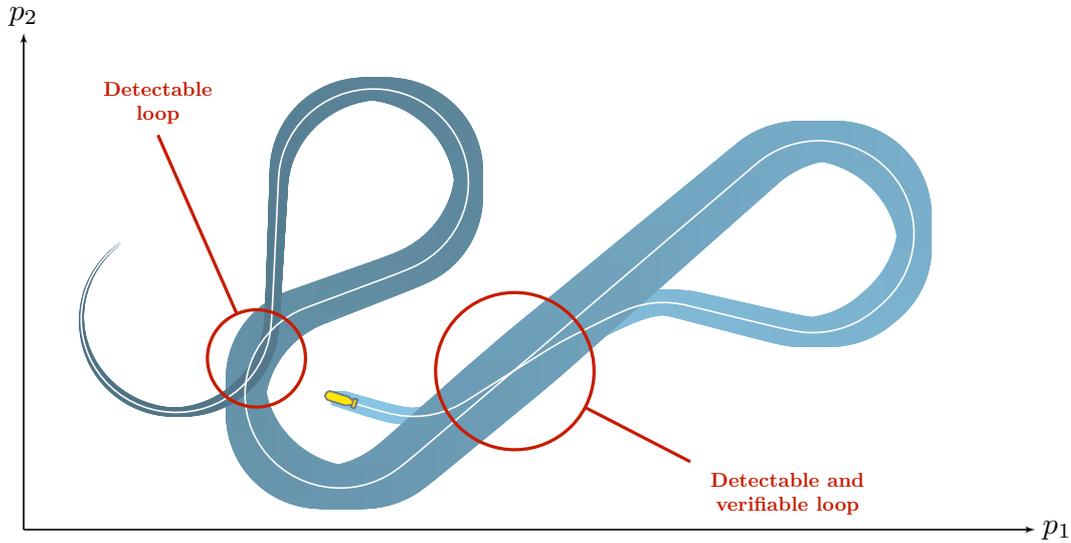


Figure 2: Only one loop can be *verified* in this set of trajectories, while at least two feasible loops are *detected*. Indeed, there exist trajectories that loop only once.

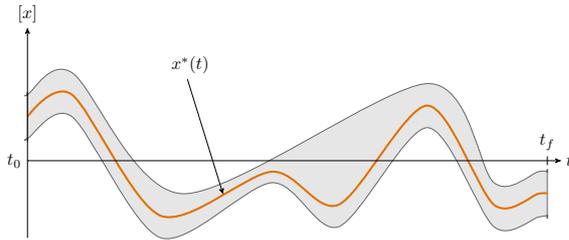


Figure 3: A one-dimensional tube enclosing an uncertain trajectory.

tegral computations of tubes will allow to approximate all feasible loops \mathbf{t} : so-called *loop sets* denoted by \mathbb{T} . From tubes, we can compute reliable inclusion functions $[\mathbf{f}]$ of \mathbf{f} . Then:

$$\mathbb{T} = \{\mathbf{t} \mid \mathbf{0} \in [\mathbf{f}](\mathbf{t})\}.$$

Formally, we want to verify that $\forall \mathbf{f} \in [\mathbf{f}], \exists \mathbf{t} \in \mathbb{T}$ such that $\mathbf{f}(\mathbf{t}) = \mathbf{0}$, which is equivalent to verifying a zero of an unknown function $\mathbf{f} \in [\mathbf{f}]$ on \mathbb{T} .

Topological degree

For this zero verification, we employ the notion of *topological degree* that originates in the

field of differential topology. An algorithm exists [2] to verify a zero of an uncertain function $\mathbf{f} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ known to belong to an inclusion function $[\mathbf{f}] : \mathbb{R}^2 \rightarrow \mathbb{R}^2$.

We will show its use as a powerful verification tool for proving robot loops. This will be demonstrated on actual datasets from real missions involving autonomous underwater vehicles at sea.

References

- [1] C. Aubry, R. Desmare, and L. Jaulin. Loop detection of mobile robots using interval analysis. *Automatica*, 49(2):463–470, Feb. 2013.
- [2] P. Franek and S. Ratschan. Effective topological degree computation based on interval arithmetic. *Mathematics of Computation*, 84(293):1265–1290, Sept. 2014.
- [3] S. Rohou, P. Franek, C. Aubry, and L. Jaulin. Proving the existence of loops in robot trajectories. *The International Journal of Robotics Research*, 37(12):1500–1516, 2018.

Guaranteed Trajectory Tracking using Flatness

Amit Kumar^{*1} and Olivier Mullier²

¹École Centrale de Nantes

²ENSTA ParisTech

Keywords: Trajectory tracking, Model predictive control, Set-membership computation, Differential flatness

Introduction

In this talk we are interested in tracking the trajectory of differential systems when the system is time continuous. Its implementation can be greatly simplified when the system checks the flatness property. The proposed method then makes use of non-linear model-based predictive control [1] (a control-law strategy to direct the state of a cyber-physical system along a given trajectory predict). BoxRRT [6] is an algorithm based on RRT (Rapidly exploring Random Tree) Algorithm combined with interval analysis tools (e.g., guaranteed numerical integration). It computes an outer approximation of the states at each time interval k . It takes into account the model of the studied system and a map of static obstacles.

NMPC Among the control methods capable of tracking a reference trajectory, Nonlinear Model Predictive Control (NMPC) is well-adapted, especially in the presence of constraints on state and/or input variables [1]. The aim of NMPC is to determine a sequence of controls by solving a constrained optimization problem at time k over a prediction horizon n_p .

Differential flatness The idea of differential flatness was first introduced by Fliess et al. in 1995 [2]. A system is differentially flat

if there exists a set of independent variables (equal in number to the dimension of inputs) referred to as the flat output such that all states and inputs of the system can be expressed in terms of this flat output and a finite number of its successive time derivative (resp. advances) for continuous-time (resp. discrete-time, see [4]) systems.

We consider the class of non-linear continuous systems described by:

$$\begin{aligned} \dot{x} &= f(x, u) \\ y &= h(x) \end{aligned} \quad (1)$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$ are respectively the state, the input and the output of the system. The functions $f : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ and $h : \mathbb{R}^n \mapsto \mathbb{R}^p$ are nonlinear vector functions.

The system is said differentially flat if there exist a particular output (named flat output) $z \in \mathbb{R}^m$ which is a projection on m coordinates of the state x such that the state and the output of the system can be described using z and a particular number of its successive time derivatives:

$$\begin{aligned} z &= \varphi(x, u, \dot{u}, \dots, u^m) \\ x &= \varphi_0(z, \dot{z}, \dots, z^{(r)}) \\ u &= \varphi_1(z, \dot{z}, \dots, z^{(r+1)}) \\ \dot{\varphi} &= f(\varphi_0, \varphi_1) \end{aligned} \quad (2)$$

with r corresponding to the relative degree of the system. Then, the knowledge of the value of z and its successive time derivatives over time allows to characterize the state and the output of the system.

In [5], the computation of a guaranteed inner approximation of the set of the admissible controls using the flatness and NMPC has been addressed in the case of discrete time systems. The goal of the presentation is to extend this result in the case of continuous time

^{*}Corresponding author.

system.

The method

Thanks to boxRRT, we are able to get a set of states values for which the system is guaranteed to reach a defined goal without colliding with static obstacles. When the system is flat, this result can be used to produce a set of values for the flat output and its time derivative and then allow to provide a set of admissible value for the input of the system. Indeed, if we consider $u(t)$ as piece-wise constant, we have

$$\begin{aligned} [\dot{x}_k] &= f([x_k], u_k) \\ [\ddot{x}_k] &= \frac{\partial f}{\partial x}([x_k], u_k) \cdot [\dot{x}_k] \\ &\vdots \end{aligned}$$

Using Eq. (2), we are then able to produce a set of controls that can be applied at time k . A particular control is then applied to the system (the midpoint of the characterized input set for instance).

Experiment

The method is illustrated using the Dubin's car model (see [3]) which is given by:

$$\begin{aligned} \dot{x} &= u \cos \theta \\ \dot{y} &= u \sin \theta \\ \dot{\theta} &= v \end{aligned} \quad (3)$$

In [3], it has already been shown that this system is flat with the flat output $z = (x, y)$ and

$$u = \sqrt{\dot{x}^2 + \dot{y}^2} \quad (4)$$

$$v = \frac{\ddot{y}\dot{x} - \ddot{x}\dot{y}}{\dot{x}^2 + \dot{y}^2} \quad (5)$$

All the system variables x, y, θ, u, v are thus expressed as function of $x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}$. Using boxRRT, we are able to provide a set of trajectories avoiding any obstacles and reaching a given goal (see Figure 1). This result is then used to define a control synthesis using NMPC.

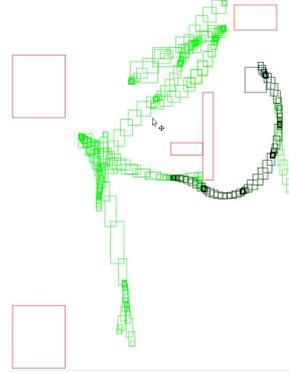


Figure 1: Example of result (in the phase space) of the boxRRT algorithm (red: obstacles; blue: goal; green: candidates paths; black: solution).

References

- [1] F. Allgöwer and A. Zheng. Nonlinear model predictive control. volume 26, 2012.
- [2] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: Introductory theory and examples. *Journal of Control*, 61(6):1327–1361, 1995.
- [3] J. Lévine. *Analysis and Control of Nonlinear Systems*. Springer Berlin Heidelberg, 2009.
- [4] J. B. Mare and J. A. De Donà. A case study in explicit solutions to constrained nonlinear MPC. *In Conference on Nonlinear Model Predictive Control for Fast Systems*, 2006.
- [5] O. Mullier and E. Courtial. Set-membership coputation of admissible controls for the trajectory tracking. *Reliable Computing*, 2017.
- [6] A. Panchea, A. Chapoutot, and D. Filiat. Extended reliable robust motion planners. *56th IEEE Conference on Decision and Control*, Dec 2017.

Prospects on the application of necessary optimality conditions on the resolution of the Goddard problem with unknown bounded parameters using interval arithmetics

Etienne Bertin^{*1,2}, Elliot Brendel², Bruno Hérissé², Alexandre Chapoutot¹, and Julien Alexandre dit Sandretto¹

¹ENSTA ParisTech, {firstName.lastName@ensta-paristech.fr}

²DTIS, ONERA, Université Paris Saclay, F-91123 Palaiseau, France, {firstName.lastName@onera.fr}

Keywords: Intervals; Optimal control; Command; Ordinary differential equations; Validated simulation

Introduction

Optimal control of aerospace systems is performed by modelling the considered system by dynamics depending on multiple uncertain parameters (for example, aerodynamic coefficients and maximal thrust). Usually, the optimal control problem is solved for the nominal values of these parameters and the robustness of the solution is demonstrated by dispersing the parameters around nominal values with Monte Carlo simulations. In addition to parameter uncertainties, the problem-solving method often introduces numerical approximation (for example the numerical solver of the ordinary differential equation representing the dynamics of the system or the optimization algorithm solving the optimal control problem).

Interval Arithmetics has shown its ability to address several control problems, providing validated solutions while dealing with method uncertainties (numerical approximations) as well as with model uncertainties (unknown parameters). The Pontryagin Maximum Principle [3] (PMP) provides necessary optimality conditions for the resolution of optimal control problems by transforming an optimal

control problem into a zero-finding problem : the dynamics of the system is extended with a co-state and necessary conditions are given by the PMP on that co-state, and the initial value of the co-state vector is the unknown to be found by the zero-finding algorithm. This method has proven its efficiency and its precision compared to direct methods [3], but its convergence depends strongly on its initialization and a prior knowledge of the solution structure is needed.

Our goal is to address the return version of the Goddard problem, which consists in performing the landing of the first stage of a rocket while minimizing its fuel consumption, combining interval arithmetics and the necessary optimality conditions given by the application of the PMP. Although this goal has not been reached yet, this paper presents preliminaries results on simplified problems, exposes the challenges encountered and suggests further developments. The optimal re-entry trajectory for the Goddard problem is presented in Figure 1 with the ballistic phases in blue, and in red the first boost for the inversion of the speed vector, the intermediate boost for the dynamic pressure constraint and the landing boost. The evolution of the dispersions along the complete trajectory (launch and landing) of a rocket are described in [1].

As a first step, a very simplified version of the Goddard problem is studied, namely a double integrator where $[k] = [\underline{k}, \bar{k}]$ is an interval parameter and u is the control in-

*Corresponding author.

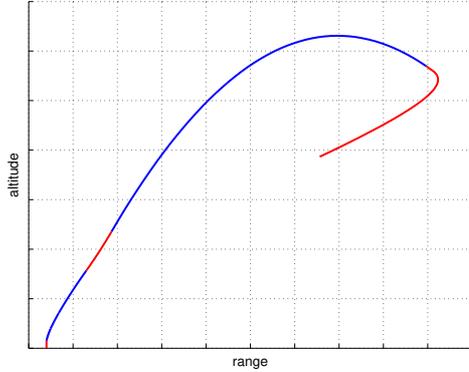


Figure 1: Optimal trajectory for the re-entry Goddard problem

put. This double integrator fits in with a Goddard problem without gravity and aerodynamic forces, and with a constant mass. The interval parameter $[k]$ represents the uncertainty on the maximal thrust force. Hence, the optimal control problem is

$$\min \int_0^T |u| dt, \begin{cases} \dot{r} = v, \dot{v} = [k]u, \\ r(0) = 0, v(0) = 0, \\ r(T) = r_T, v(T) \text{ is free,} \\ T \text{ is fixed.} \end{cases}$$

Two ways of combining interval arithmetics with the PMP are investigated : an *open loop* approach providing an enclosure on the system trajectory which can be used to assess robustness (regarding the interval parameter $[k]$), and a *closed loop* approach providing a closer enclosure of the optimal trajectories and can be used to initialize a non-interval algorithm.

Once the double integrator optimal control problem is solved, multiple approaches such as validated continuation methods are considered to solve the Goddard problem using this simplified version.

Open-loop approach

In this approach, the goal is to find the smallest initial co-state interval that contains every

admissible co-state by combining numerical integration tools such as DynIbex¹ [2] and an algorithm to solve the zero-finding problem. Many algorithms are considered, for example Krawczyk method, forward-backward operators and branch algorithms. The enclosure of the solution of the zero-finding problem provides a validated initialization for the co-state vector. Issues like discontinuous control input, control saturation, pure state constraints and mixed constraints are to be studied in order to solve a practical optimal control problem like the Goddard problem.

Closed-loop approach

In this approach, dynamic programming is used to find a finer enclosure of the optimal trajectories : if the system measures its interval state vector at a certain time, a new optimal control problem is solved from this interval, providing a better enclosure of the solution. Due to its algorithmic cost, this approach is irrelevant for solving practical cases online, but it can improve the enclosure of the solution when it is applied offline and therefore provides useful information for the online guidance algorithm.

References

- [1] L. Blackmore. Autonomous precision landing of space rockets. *The Bridge*, 46:15–20, Jan 2016.
- [2] J. A. dit Sandretto and A. Chapoutot. Validated explicit and implicit Runge–Kutta methods. *Reliable Computing*, 22(1):79–103, Jul 2016.
- [3] E. Trélat. Optimal control and applications to aerospace: some results and challenges. *Journal of Optimization Theory and Applications*, 154(3):713–758, 2012.

¹<http://perso.ensta-paristech.fr/~chapoutot/dynibex/>

List of Authors

Abel, D., 55
Adam, S.P., 5
Alamo, T., 41
Alexandre dit Sandretto, J., 53, 69, 85
Aschemann, H., 35
Aubry, C., 81

Benet, L., 15
Bertin, É., 85
Brendel, E., 85
Brown, J., 17

Camacho, E. F., 41
Casado, L. G., 5
Chapoutot, A., 85
Chevet, T., 41

Damers, J., 23
Dbouk, H., 59
Drevelle, V., 73

Forets, M., 15
Franek, P., 81

Gatto, T., 45
Gehrt, J.-J., 55

Hérissé, B., 85

Jaulin, L., 23, 27, 37, 81

Karamousadakis, M., 77
Kersten, J., 35, 67
Kumar, A., 83

Lange, M., 49
Le Bars, F., 27
Lefort, A., 31

Liu, S., 55
López, D., 5

Makarov, M., 41
Merhy, D., 41
Meyer, L., 45
Mullier, O., 53, 83

Nasiotis, K. A., 5
Nico, T., 37

Panayides, A., 77
Pessaux, F., 17
Piet-Lahanier, H., 45

Ramdani, N., 77
Rauh, A., 35, 67
Revol, N., 9
Rohou, S., 23, 81

Sanders, D.P., 15
Schilling, C., 15
Schön, S., 59
Selivanova, S., 21
Stoica Maniu, C., 41

Tillet, J., 27

Voges, R., 63
Vorontsova, E., 11

Wagner, B., 63

Zeinalipour-Yazti, D., 77
Zerr, B., 37
Ziegler, M., 21
Zweigel, R., 55