15th Summer Workshop on Interval Methods SWIM 2024

Abstracts

Department of Advanced Computing Sciences
Maastricht University

4-6 June 2024

SWIM 2024 - Programme

Tuesday 4th June

09:00	Reception	
09:20	Opening **Mark Winands**	
09:35	**Bartłomiej Kubica**	*Fuzzy Systems and Neural Networks*
10:45	Coffee	
11:15	**Nathalie Revol**	*About 8-bit floating-point numbers for machine learning: the IEEE 3109 project*
11:50	**Pieter Collins**	*An overview of interval methods in artificial intelligence, law and taxation*
12:25	Lunch	
13:45	**Andreas Rauh**	*Interval Methods for the Identification of Initialization Functions of System Models with Delay*
14:20	**Marit Lahme**	*Systematic Approach for Parameterizing TNL Interval Observers*
14:55	**Damien Esnault**	*Estimating the success probability of a set event in a probabilistic world*
15:30	Coffee	
16:00	**Julien Alexandre Dit Sandretto**	*On Propagating Uncertainties for Estimation and Association using Validated Interval Simulation*
16:35	**Maël Godard**	*Introducing Box Chains to simplify Reachability Analysis*
17:10	**Luc Jaulin**	*Outer approximation of the occupancy set left by a mobile robot*
17:45	Dinner	

Wednesday 5th June

09:00	**Jan Bouwe van den Berg**	*Computer Assisted Proofs in Dynamical Systems*
10:10	**Lucas Si Larbi**	*Optimal Path Planner over Receding Horizon using Interval Analysis*
10:45	Coffee	
11:15	**Joris Tillet**	*Interval Methods applied to Signal Temporal Logic – Overview and Extension on Tubes*
11:50	**Damien Massé**	*Guaranteed integration on Lie groups*
12:25	**Norbert Müller**	*Exact Real Arithmetic and the Efficiency of Taylor Models*
13:00	Lunch	
16:30	Social Event	Guided Tour of the North Caves, Luikerweg 80, Maastricht
18:00	Dinner	

Thursday 6th June

09:00	**Nuwan Herath Mudiyanselage**	*Reinforced Set Projection Algorithm*
09:35	**Margarita Korovina**	*Solving non-linear constraints in CDCL-style*
10:10	**Simon Rohou**	*An aymptotic minimal contractor for non-linear equations using the Codac library*
10:45	Coffee	
11:15	**Quentin Brateau**	*Dimensioning a torpedo-like AUV using interval analysis*
11:50	**Vincent Drevelle**	*Ultra-Wideband based Smart Wheelchair Localization using Interval Analysis*
12:25	Closing	

About 8-bit floating-point numbers for machine learning: the IEEE 3109 project

Nathalie Revol

INRIA - LIP, ENS Lyon
site Monod, 46 allée d'Italie, 69007 Lyon, France
{Nathalie.Revol}@ens-lyon.fr

Keywords: Floating-Point Number, 8-Bit Format, Machine Learning

Introduction

Machine learning needs small floating-point number formats to increase the speed of computations while decreasing the memory size and the energy consumption. Constructors, such as ARM, Intel and NVidia [1], began developing 8-bit formats for floating-point numbers. As other constructors, such as GraphCore, developed other formats, the need for standardizing this format arose.

The IEEE 3109 working group for the standardization of floating-point numbers for machine learning has been created in 2022. In what follows, we will sum up the current state of the work [2].

Current status

Format

8-bit floating-point numbers are built using the same pattern as 32-bit or 64-bit ones, with some exceptions that will be detailed during the talk. The format of a floating-point number is chosen to occupy 8 bits: one bit for the sign, p bits for the significand, including the hidden bit, the remaining bits are used to store the exponent.

Rounding modes

Mandatory rounding modes are still being discussed.

Operations

The set of operations is not yet fully specified. The behavior of operations when an overflow can be either to saturate, that is, to return the largest (in absolute value) representable number, or to return an infinite value.

Conclusion and perspectives

This is an ongoing work and all of the final specifications of the future standard are not yet chosen. Remaining questions about this future standard are numerous. Will it be usable for interval arithmetic? Namely, will be directed rounding modes available? Is the machine learning community interested in interval arithmetic, for instance to guarantee some results? Will it be used by applications outside machine learning? If it is used by applications such as robotics or control theory, how will it impact computations and results?

References

[1] P. Micikevicius, D. Stosic, N. Burgess, M. Cornea. P. Dubey, R. Grisenthwaite, S. Ha, A. Heinecke, P. Judd, J. Kamalu, N. Mellempudi, S. Oberman, M. Shoeybi, M. Siu, and H. Wu. *FP8 formats for deep learning.* `https://arxiv.org/abs/2209.05433`, 2022.

[2] IEEE 3109 interim report (February 2024). `https://github.com/P3109/Public/blob/main/Shared%20Reports/P3109%20WG%20Interim%20Report.pdf`

An Overview of Interval Methods for Artificial Intelligence, Law and Taxation

Pieter Collins

Department of Advanced Computing Sciences
Maastricht University
pieter.collins@maastrichtuniversity.nl

Keywords: Interval methods, Artificial intelligence, Law, Finance

Introduction

Interval methods have been widely used in control engineering, notably for verification of safety-critical systems or handling non-stochestic uncertainties. However, many other domains could benefit from interval methods and other rigorous numerical techniques to provide reliable computational analyses, especially when questions of legal compliance are involved. In particular, given the opaque behaviour of many artificial intelligence (AI) systems based on machine-learning (ML), legislation is increasingly demanding more insight into such systems, notably regarding the explainablity of recommendations made.

In this talk, I will give an overview of some problems in AI and the legal domain, and approaches involving interval analysis for their solution, including a project for the Dutch Tax Service in model-checking well-definedness of their computations.

Querying Grey-Box Models

Many AI sytems make a prediction or recommendation of an output variable y based on input variables x. Such a system is a *black-box* model: one can merely see the output returned for a given input. This makes gaining insight into the internal workings very difficult. Using polymorphism, we can implement an AI system to accept any *applicative functor* [6]; We call such a system a *grey-box model* [3]. Since applicative functors include intervals, differentials and expressions, we can make extended queries of grey-box models. Applying the system to intervals (or more general sets) can allow us to prove properties of the form *"for all inputs* $x \in [x]$*, the output* $f(x)$ *lies in* $[y]$*"*.

Verifying Neural Networks

Neural networks are a cornerstone model in modern artificial intelligence. Networks with a ReLU activation function $\sigma(x) = \max(0, x)$ are piecewise-affine functions, and can be verified by the ReLUplex algorithm [5], which is a branch-and-bound approach built on top of the simplex algorithm, Smooth activation functions can be accurately handled using Taylor polynomial models [4]. We have compared these approaches in [7] using these techniques.

Correctness legal algorithms

Software for implementing algorithms described in legislation has high standards of correctness, including that of numerical computations. In [2], in which the authors consider European regulations for drivers' rest-periods, and the use of tachographs. This work illustrates subtle logical and numerical artificts in the interpretation of these laws.

Model-checking tax rules

RgelSpraak [1] is a controlled natural language (CNL) for defining the rules by which tax is computed. It is used in the Agile Law Execution Factory (ALEF) which automatically generates software computing the tax due. However, these rules might not be well-defined in all circumstances, for example, in a rule involving division, the divisor should never be allowed to be zero, The Dutch Tax Office (Belastingdienst) is interested in statically "model checking" the rules to find and prevent such ill-formed conditions. Mathematically, this problem reduces to computing the ranges of functions, possibly over unbounded domains, and showing that a partial function such as $\div$ or $\sqrt{}$ can never be given an argument outside its domain. Mathematically, this reduces to is a constraint satisfaction problem, possibly over an unbounded domain, which can be rigorously solved by interval mathods.

References

[1] Mischa Corsius, Stijn Hoppenbrouwers, Mariette Lokin, Elian Baars, Gertrude Sangers-Van Cappellen, and Ilona Wilmont. RegelSpraak: a CNL for executable tax rules specification. In *Proceedings of the Workshop on Controlled Natural Language*, 2021.

[2] Ana de Almeida Borges, Juan José Conejero Rodríguez, David Fernández-Duque, Mireia González Bedmar, and Joost J. Joosten. To drive or not to drive: A logical and computational analysis of European transport regulations. *Information and Computation*, 280, 2021.

[3] Tom Donney, Haridasyam Koushik, Bartlomiej Bitner, Zhuoer Ma, and Xinqian Qian. Queryable systems for explanations. Master's thesis, Department of Data Science and Knowledge Engineering, Maastricht University, 2020.

[4] Chao Huang, Jiameng Fan, Xin Chen, Wenchao Li, and Qi Zhu. POLAR: A polynomial arithmetic framework for verifying neural-network controlled systems. In *Automated Technology for Verification and Analysis*, pages 414–430, 2022.

[5] Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. ReLuPlex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, 2017.

[6] Conor McBride and Ross Paterson. Applicative programming with effects. *Journal of Functional Programming*, 18(1):1–13, 2008.

[7] Borislava Sandeva, Lukas Padolevicius, Vasileios Chaidas, Philip Salomons, and Dheemantha Bhat. Verification of machine learning. Technical report, Department of Data Science and Knowledge Engineering, Maastricht University, 2022. Master Project Report.

Interval Methods for the Identification of Initialization Functions of System Models with Delay

Andreas Rauh and Marit Lahme

Carl von Ossietzky Universität Oldenburg
Distributed Control in Interconnected Systems
26111 Oldenburg, Germany
{Andreas.Rauh, Marit.Lahme}@uni-oldenburg.de

Keywords: System models with delay, delay-differential equations, fractional-order differential equations, initialization functions

Introduction

Many real-life systems are characterized by delay phenomena [3, 4, 8]. Among others, these are caused by physical transport processes over long spatial domains or by communication via networks including the associated processing times. When modeling such systems in a continuous-time form, either delay-differential equations or fractional-order differential equations are employed. These system models are characterized by the fact that not only the initial condition at a specific time instant and the corresponding external input signals after that point need to be known to make predictions about the future behavior but also knowledge about the temporal evolution of the system state for the (entire) past is required [6, 7].

A discretization in time leads to sets of difference equations which again include not only the mapping of state vectors $\mathbf{x}(t)$ from one time instant $t = t_k$ to a subsequent point t_{k+1} but which also dependent on the predecesssor states $\mathbf{x}(t_{k-1}), \mathbf{x}(t_{k-2}), \mathbf{x}(t_{k-3}), \ldots$.

Identification of Past (Pseudo) State Evolutions

To distinugish the memory phenomenon mentioned before from the notation of the system state in classical ordinary differential equations, $\mathbf{x}(t)$ is subsequently denoted as a pseudo state if it refers to system models with delay. In this contribution, the approach of identifying initial conditions with an observability matrix-based procedure [5] is transferred to the identification of the pseudo state initialization.

For that purpose, consider first a linear discrete-time difference equation

$$\mathbf{x}(t_{k+1}) = \mathbf{A} \cdot \mathbf{x}(t_k) + \mathbf{B} \cdot \mathbf{u}(t_k) \ , \quad \mathbf{x}(t_k) \in \mathbb{R}^n \tag{1}$$

with the measured output $y(t_k) = \mathbf{c}^T \cdot \mathbf{x}(t_k) \in \mathbb{R}$.

The evaluation of the so-called observability mapping [1,9] yields

$$\begin{aligned} \mathbf{q} = \begin{bmatrix} y(t_0) \\ y(t_1) \\ \vdots \\ y(t_{n-1}) \end{bmatrix} &= \begin{bmatrix} \mathbf{c}^T \\ \mathbf{c}^T\mathbf{A} \\ \vdots \\ \mathbf{c}^T\mathbf{A}^{n-1} \end{bmatrix} \cdot \mathbf{x}(t_0) \\ &+ \begin{bmatrix} 0 \\ \mathbf{c}^T\mathbf{B} \cdot \mathbf{u}(t_0) \\ \vdots \\ \mathbf{c}^T \cdot \big(\mathbf{B} \cdot \mathbf{u}(t_{n-2}) + \mathbf{A}\mathbf{B} \cdot \mathbf{u}(t_{n-3}) + \ldots \mathbf{A}^{n-2}\mathbf{B} \cdot \mathbf{u}(t_0)\big) \end{bmatrix} \\ &= \mathbf{Q}_\mathrm{O} \cdot \mathbf{x}(t_0) + \mathbf{\Delta} \ , \end{aligned} \tag{2}$$

where $\mathbf{Q}_\mathrm{O}$ is the observability matrix. In the case that the system is fully observable, coinciding with a full rank of $\mathbf{Q}_\mathrm{O}$, the initial state $\mathbf{x}(t_0)$ can be uniquely reconstructed by an analytic solution of (2) under the assumption of perfectly known system matrices and control inputs $\mathbf{A}$, $\mathbf{B}$, $\mathbf{c}^T$, $\mathbf{u}(t_0)$, ...

In recent work, cf. [5], this idea has been generalized to a set-based estimation approach that also allows for handling bounded uncertainty in the measured outputs as well as in the system parameters.

During this presentation, we aim at further generalizing this approach towards a set-based contraction procedure that allows for tightening a-priori bounds on the initialization function, given in the form of the intervals $[\mathbf{x}](t_k)$ with $k \in \{\dots, -3, -2, -1, 0\}$.

Illustrating Example

Consider the discrete-time difference equation $x(t_{k+1}) = 0.9x(t_k) + 0.1x(t_{k-1})$ with the direct state measurement $y(t_k) = x(t_k) + \upsilon_k$ and the bounded measurement uncertainty $\upsilon_k \in [\upsilon_k] = [\underline{\upsilon}_k \ ; \ \overline{\upsilon}_k]$. Denote the corresponding measurement intervals by $[y](t_k)$ for which $x(t_k) \in [y](t_k)$ holds with $0 \in [\upsilon_k]$.

Using this information, an interval-based contractor [2] can be implemented that accounts for the following model:

$$\begin{aligned} x(t_1) &= 0.9x(t_0) + 0.1x(t_{-1}) \in [y](t_1), \\ x(t_2) &= 0.9x(t_1) + 0.1x(t_0) \in [y](t_2), \dots, \\ x(t_5) &= 0.9x(t_4) + 0.1x(t_3) \in [y](t_5) \ . \end{aligned} \tag{3}$$

The solution approach now firstly starts with the most recent measurement $[y](t_5)$ and tries to exploit this information to enhance the a-priori enclosures for $x(t_1) \in [y](t_1)$ and $x(t_2) \in [y](t_2)$. This information is then temporally propagated backward to obtain feasible bounds for $[x](t_{-1})$ and $[x](t_0)$, see Fig. 1. For system models with an infinite memory of previous states, the solution approach is combined with the finite-memory technique presented in [7], which is essentially based on a change of coordinates to circumvent the necessity to account for infinitely many previous time instants.

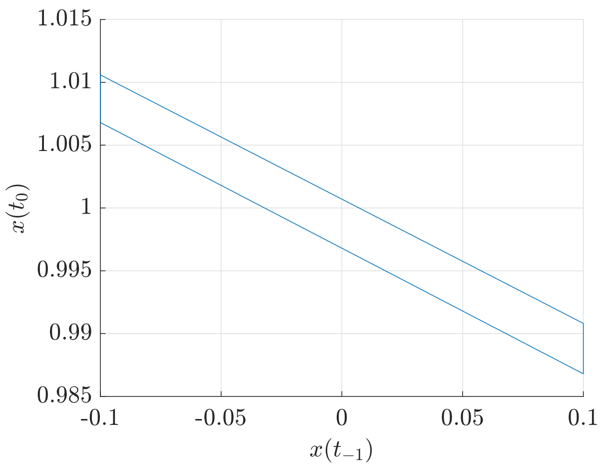


Figure 1: Identification of the dependency between $x(t_{-1})$ and $x(t_0)$, represented as a constrained zonotope with $[\upsilon_k] = [-0.05 \ ; \ 0.05]$.

References

[1] R. Hermann and A. J. Krener. Nonlinear Controllability and Observability. *IEEE Transactions on Automatic Control*, 22(5):728–740, 1977.

[2] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter. *Applied Interval Analysis*. Springer–Verlag, London, 2001.

[3] Y. Kyrychko and S. Hogan. On the Use of Delay Equations in Engineering Applications. *Journal of Vibration and Control*, 16, 07 2010.

[4] A. Martins, P. Laranjeira, M. Dias, and J. C. Lopes. Mass Transport Modelling in Porous Media Using Delay Differential Equations. *Defect and Diffusion Forum*, 258–260:586–591, 01 2006.

[5] N. Meslem and N. Ramdani. A New Approach to Design Set-Membership State Estimators for Discrete-Time Linear Systems Based on the Observability Matrix. *International Journal of Control*, 93(11):2541–2550, 2020.

[6] A. Rauh and E. Auer. Verified Integration of Differential Equations with Discrete Delay. *Acta Cybernetica*, 25(3):677–702, 2022.

[7] A. Rauh and M. Lahme. A Finite Memory Approach for Verified Pseudo State Estimation of Fractional Models of Lithium-Ion Batteries. In *Proceedings of 12th IFAC Conference on Fractional Differentiation and its Applications*, Bordeaux, France, 2024.

[8] A. Seuret, L. Hetel, J. Daafouz, and K. H. Johansson. *Delays and Networked Control Systems.* Advances in Delays and Dynamics. Springer International Publishing, Switzerland, 2016.

[9] E. D. Sontag. *Mathematical Control Theory — Deterministic Finite Dimensional Systems.* Springer–Verlag, New York, 1998.

Systematic Approach for Parameterizing TNL Interval Observers

Marit Lahme and Andreas Rauh

Carl von Ossietzky Universität Oldenburg
Distributed Control in Interconnected Systems
26111 Oldenburg, Germany
{Marit.Lahme, Andreas.Rauh}@uni-oldenburg.de

Keywords: State Estimation, LMI, Uncertain Dynamic Systems

Introduction

Reliable state estimation is essential for practical control applications, because oftentimes not all state variables can be measured. It is therefore necessary to estimate the corresponding state variables based on available measurements, for example with the help of state observers. During the observer design and parameterization, uncertainties have to be taken into account when dealing with real systems, because they are always affected by uncertainties which include for instance measurement noise, process noise or parametric uncertainty. A common approach for state estimation in systems with bounded uncertainty is the use of interval observers, which provide lower and upper bounds, that always enclose the true trajectories. Among other techniques, interval observers are recurrently designed based on the monotone system theory [1]. A consequence of that is, that the design conditions are augmented by the condition of cooperativity, which can complicate finding appropriate observer gains. There are numerous techniques to address this problem. One possible method for relaxing the design conditions is to use a TNL observer structure [2]. Besides the observer gain $\mathbf{L}$, this observer structure provides two additional degrees of freedom with the matrices $\mathbf{T}$ and $\mathbf{N}$. However, it is still challenging to find suitable observer gains for real systems, especially when the state variables to be estimated are scaled differently or the dynamics are described by stiff system models. In this presentation, we will discuss several techniques that can be used to systematically parameterize the observer gains of a TNL observer for uncertain systems with differently scaled state variables.

TNL Observer

Consider an uncertain time-invariant system according to

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_\mathrm{d} \cdot \mathbf{x}_k + \mathbf{B}_\mathrm{d} \cdot \mathbf{u}_k + \mathbf{E}_\mathrm{d} \cdot \boldsymbol{w}_k \\ \mathbf{y}_k &= \mathbf{C} \cdot \mathbf{x}_k + \boldsymbol{v}_k \ , \end{aligned} \tag{1}$$

with $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{y}_k \in \mathbb{R}^m$, $\mathbf{u}_k \in \mathbb{R}^p$, $\boldsymbol{w}_k \in \mathbb{R}^q$, $\boldsymbol{v}_k \in \mathbb{R}^m$. The matrices $\mathbf{A}_\mathrm{d}$, $\mathbf{B}_\mathrm{d}$, $\mathbf{C}$, and $\mathbf{E}_\mathrm{d}$ are constant and of appropriate dimensions. We assume that the process and measurement uncertainties are unknown but bounded, so that $\underline{\boldsymbol{w}}_k \le \boldsymbol{w}_k \le \overline{\boldsymbol{w}}_k$ and $\underline{\boldsymbol{v}}_k \le \boldsymbol{v}_k \le \overline{\boldsymbol{v}}_k$. As proposed by Z. Wang et al. in [2], a TNL observer can be designed for system (1) corresponding to

$$\begin{aligned} \zeta_{k+1} &= \mathbf{T}\mathbf{A}_\mathrm{d}\hat{\mathbf{x}}_k + \mathbf{T}\mathbf{B}_\mathrm{d}\mathbf{u}_k + \mathbf{L}(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_k) + \boldsymbol{\Delta}_k \ , \\ \hat{\mathbf{x}}_k &= \zeta_k + \mathbf{N}\mathbf{y}_k \ , \end{aligned} \tag{2}$$

with the observer gains $\mathbf{T}$, $\mathbf{N}$, $\mathbf{L}$ and $\underline{\zeta}_{k+1} \le \zeta_{k+1} \le \overline{\zeta}_{k+1}$, $\underline{\hat{\mathbf{x}}}_k \le \hat{\mathbf{x}}_k \le \overline{\hat{\mathbf{x}}}_k$, $\overline{\zeta}_{k+1} = \overline{\zeta}_{k+1}(\overline{\hat{\mathbf{x}}}_k(\overline{\zeta}_k), \overline{\boldsymbol{\Delta}}_k)$, $\underline{\zeta}_{k+1} = \underline{\zeta}_{k+1}(\underline{\hat{\mathbf{x}}}_k(\underline{\zeta}_k), \underline{\boldsymbol{\Delta}}_k)$, $\boldsymbol{\Delta}_k = \boldsymbol{\Delta}_k(\mathbf{T}\mathbf{E}_\mathrm{d}\, \boldsymbol{w}_k, \mathbf{L}\, \boldsymbol{v}_k, \mathbf{N}\, \boldsymbol{v}_{k+1})$ and $\underline{\boldsymbol{\Delta}}_k \le \boldsymbol{\Delta}_k \le \overline{\boldsymbol{\Delta}}_k$. With the error defined as $\overline{\mathbf{e}}_k = \overline{\hat{\mathbf{x}}}_k - \mathbf{x}_k$ and $\underline{\mathbf{e}}_k = \underline{\hat{\mathbf{x}}}_k - \mathbf{x}_k$, $\tilde{\mathbf{B}}_\mathrm{d} = \begin{bmatrix}\mathbf{I}_n & \mathbf{L} & \mathbf{N}\end{bmatrix}$, $\mathbf{d}_k = \begin{bmatrix}(\boldsymbol{\Delta}_k - \mathbf{T}\mathbf{E}_\mathrm{d}\boldsymbol{w}_k) & \boldsymbol{v}_k & \boldsymbol{v}_{k+1}\end{bmatrix}^T$, $\overline{\mathbf{d}}_k = \mathbf{d}_k(\overline{\boldsymbol{\Delta}}_k)$, $\underline{\mathbf{d}}_k = \mathbf{d}_k(\underline{\boldsymbol{\Delta}}_k)$, and a given scalar $\gamma > 0$, the error dynamics of this state observer can be written as [2]

$$\mathbf{e}_{k+1} = (\mathbf{T}\mathbf{A}_\mathrm{d} - \mathbf{L}\mathbf{C})\mathbf{e}_k + \tilde{\mathbf{B}}_\mathrm{d}\mathbf{d}_k \ , \tag{3}$$

with $\overline{\mathbf{e}}_{k+1} = \overline{\mathbf{e}}_{k+1}(\overline{\mathbf{e}}_k, \overline{\mathbf{d}}_k)$, $\underline{\mathbf{e}}_{k+1} = \underline{\mathbf{e}}_{k+1}(\underline{\mathbf{e}}_k, \underline{\mathbf{d}}_k)$ and the virtual output equation

$$\mathbf{z}_{\mathrm{e},k} = \mathbf{C}_\mathrm{e}\mathbf{e}_k + \mathbf{D}_\mathrm{e}\mathbf{d}_k \ . \tag{4}$$

Given that the initial condition is chosen so that $\underline{\hat{\mathbf{x}}}_0 \le \mathbf{x}_0 \le \overline{\hat{\mathbf{x}}}_0$, the true value is reliably bounded with lower and upper bounds $\underline{\hat{\mathbf{x}}}_k \le \mathbf{x}_k \le \overline{\hat{\mathbf{x}}}_k$ if the matrix $\mathbf{M} = (\mathbf{T}\mathbf{A}_\mathrm{d} - \mathbf{L}\mathbf{C})$ is Schur stable and elementwise non-negative. Additionally, the error dynamics satisfy the H_∞ criterion $\parallel \overline{\mathbf{e}}_k \parallel < \gamma \parallel \overline{\mathbf{d}}_k \parallel$ and $\parallel \underline{\mathbf{e}}_k \parallel < \gamma \parallel \underline{\mathbf{d}}_k \parallel$ if there exists an arbitrary positive definite diagonal matrix $\mathbf{P}$, so that the following matrix inequality is fulfilled [2]

$$\begin{bmatrix} \mathbf{M}^T\mathbf{P}\mathbf{M} + \mathbf{C}_\mathrm{e}^T\mathbf{C}_\mathrm{e} - \mathbf{P} & \mathbf{M}^T\mathbf{P}\mathbf{B}_\mathrm{d} + \mathbf{C}_\mathrm{e}^T\mathbf{D}_\mathrm{e} \\ \mathbf{B}_\mathrm{d}^T\mathbf{P}\mathbf{M} + \mathbf{D}_\mathrm{e}^T\mathbf{C}_\mathrm{e} & \mathbf{B}_\mathrm{d}^T\mathbf{P}\mathbf{B}_\mathrm{d} + \mathbf{D}_\mathrm{e}^T\mathbf{D}_\mathrm{e} - \gamma^2\mathbf{I} \end{bmatrix} \prec 0 \ . \tag{5}$$

Systematic Observer Parameterization

Besides the observer gains $\mathbf{T}$, $\mathbf{N}$, and $\mathbf{L}$, there are two additional design degrees of freedom that can be used during the observer parameterization, namely the matrices $\mathbf{C}_\mathrm{e}$ and $\mathbf{D}_\mathrm{e}$ that determine how the virtual output is affected by the individual error terms and the process and measurement noise. In the literature, it is common to choose the matrix $\mathbf{C}_\mathrm{e}$ to be the identity matrix or a uniformly scaled matrix [3, 4, 5]. To directly impose different convergence rates of individual state variables, we choose a diagonal matrix structure which can have different diagonal elements [6]. Additionally, we propose different techniques to enforce interpretable structures in the matrices $\mathbf{T}$, $\mathbf{N}$, and $\mathbf{L}$, that can lead to improved estimation results with regard to interval width. Although this approach does not provide an optimal solution in the sense of the H_∞ criterion, it might be beneficial when dealing with uncertain systems with parametric uncertainty [7]. To specify particular matrix structures, we use additional design conditions and cost functions in the LMI solution. Those can be for example based on eigenvalues of the observer system matrix or on a specified ratio between the matrices $\mathbf{L}$ and $\mathbf{N}$. Additionally, a Luenberger observer like structure can be imposed for a simplified observer parameterization.

References

[1] H. L. Smith. *Monotone dynamical systems: An introduction to the theory of competitive and cooperative systems*, (vol) 41 of *Mathematical surveys and monographs.* American Mathematical Society, Providence, Rhode Island, USA, 1995.

[2] Z. Wang, C.-C. Lim, and Y. Shen. Interval observer design for uncertain discrete-time linear systems. *Systems & Control Letters*, 116:41–46, 2018.

[3] T. Chevet, T. N. Dinh, J. Marzat, and T. Raïssi. Robust sensor fault detection for linear parameter-varying systems using interval observer. In *Proc. of the 31st European Safety and Reliability Conference (ESREL)*, pp. 1486–1493. Angers, France, 2021. Research Publishing Services.

[4] G. Marouani, T. N. Dinh, Z. Wang, T. Raïssi, and H. Messaoud. Interval observers design for discrete-time linear switched systems. In *Complex Systems: Spanning Control and Computational Cybernetics: Foundations*, (vol) 414 of *Studies in Systems, Decision and Control*, (ed.) P. Shi, J. Stefanovski, and J. Kacprzyk, pp. 245–268. Springer Cham, 2022.

[5] Z. He. Interval observer–based compensators design for linear systems with uncertain time-varying parameters. *Transactions of the Institute of Measurement and Control*, 45(16):3247–3253, 2023.

[6] M. Hypiusová and D. Rosinová. Robust pole placement: Dr-regions for discrete-time systems. In *Proc. of the 22nd International Conference on Process Control (PC)*, pp. 108–113. Štrbské Pleso, Slovakia, 2019. IEEE.

[7] M. Lahme, A. Rauh, and G. Defresne. Interval observer design for an uncertain time-varying quasi-linear system model of lithium-ion batteries. In *Proc. of the 2024 European Control Conference (ECC)*. Stockholm, Sweden, 2024. accepted for publication.

Estimating the success probability of a set event in a probabilistic world

Damien Esnault [1], Luc Jaulin [2], Simon Rohou [2] and Fabrice Le Bars [2]

[1] DGA Techniques Navales
Rue du Cap N, 29200 Brest, France
damien.esnault@ensta-bretagne.org

[2] Lab-STICC, ENSTA Bretagne
2 rue François Verny, 29200 Brest, France
lucjaulin@gmail.com
simon.rohou@ensta-bretagne.fr
fabrice.le_bars@ensta-bretagne.fr

Keywords: Probabilities, Interval Methods, Underwater Robotics

Introduction

Because of the constraints inherent to their environment, underwater robots carry out most of their missions autonomously. For critical applications, such as mine warfare, it is essential to be able to guarantee the success of a mission or, at least, to estimate its probability of success.

Missions are often achieved by validating a set event, such as the coverage of a specific area by a sensor, or the respect of a navigation corridor. However, sensor modeling and navigation algorithms are generally based on probabilistic assumptions, such as Gaussian noise and Kalman filtering. So, to estimate the success probability of such mission, it is necessary to find a method that conciliates the ensemblist approach with the probabilistic one, without confusing the two.

Formalism

Given a random trajectory $\mathbf{x}(\cdot) : \mathbb{R} \mapsto \mathbb{R}^n$, with a known probability density function $\pi_{\mathbf{x}}$, and a random vector $\mathbf{p} \in \mathbb{R}^m$, with an unknown probability density function $\pi_{\mathbf{p}}$ but a known support $[\mathbf{p}] \in \mathbb{IR}^m$.

We consider an event E, defined with a function $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, such as:

$$E : \forall \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}, h(\mathbf{x}(t), \mathbf{p}) \leq 0. \tag{1}$$

By defining the boolean variable y, such as $y = 1$ if E is verified and $y = 0$ otherwise, we have:

$$P_E = P(y = 1) \in \int_{\mathbf{x}(\cdot) \in \mathbb{X}} [\eta](\mathbf{x}(\cdot)) \cdot \pi_{\mathbf{x}}(\mathbf{x}(\cdot)) \cdot d\mathbf{x}(\cdot) \tag{2}$$

where:

$$[\eta](\mathbf{x}(\cdot)) = \begin{cases} [1] & \text{if } \forall \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}, h(\mathbf{x}(t), \mathbf{p}) \leq 0 \\ [0] & \text{if } \forall \mathbf{p} \in [\mathbf{p}], \forall t \in \mathbb{R}, h(\mathbf{x}(t), \mathbf{p}) > 0 \\ [0, 1] & \text{otherwise} \end{cases} \tag{3}$$

As a result, we will obtain an interval $[P_E]$, containing the probability P_E of the event E.

To estimate this interval of probability, we propose to use a Monte-Carlo method, generating $N \in \mathbb{N}^*$ samples of the trajectory $\mathbf{x}(\cdot)$ and verifying for each one the value of $[\eta_i]$ with $i \in [1, N]$ and where:

$$[\eta_i] = [\eta](\mathbf{x}_i(\cdot)) \text{ where } \mathbf{x}_i(\cdot) \text{ is the } i\text{-th draw of } \mathbf{x}(\cdot)$$

It is thus possible to obtain an estimate of $[P_E]$, by calculating the average of the $[\eta_i]$:

$$[P_E] \approx \frac{1}{N} \sum_{i=1}^{N} [\eta_i] \tag{4}$$

The law of large numbers leads us to believe that this estimate will be all the more accurate as $N \to \infty$.

Simulation

In this scenario, we consider a mobile robot described by a state equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ and initialized to a given state $\mathbf{x_0}$. It aims to follow a reference trajectory in the plane (represented by a black line in the Figure 1).

To do so, the robot evolves in dead-reackoning, with only a compass to estimate its heading (with a white Gaussian noise). Using a controller and a Kalman filter, it can generate a command $\mathbf{u_d}$ from the estimate of its state $\hat{\mathbf{x}}$. The robot's trajectory $\mathbf{x}(\cdot)$ is therefore random, and $N \in \mathbb{N}^*$ draws from this trajectory can be obtained by simulating the system N times.

We now introduce into the scenario three beacons whose positions are uncertain. The position $\mathbf{p}_j \in \mathbb{R}^2$ of the j-th beacon is in the set $[\mathbf{p}_j] \in \mathbb{IR}^2$. To detect the beacon, the robot must be close enough to it, i.e. at a distance less than or equal to the robot's detection radius r_b. We can thus define a function $h : \mathbb{R}^2 \times \mathbb{R}^2 \mapsto \mathbb{R}$ to quantify this statement:

$$h(\mathbf{x}, \mathbf{p}) = \|\mathbf{x} - \mathbf{p}\| - r_b \tag{5}$$

The event E_j describes the case where the robot has seen the j-th beacon, and can be formalised by the following condition:

$$E_j : \forall \mathbf{p}_j \in [\mathbf{p}_j], \exists t \in \mathbb{R}, h(\mathbf{x}(t), \mathbf{p}_j) \leq 0 \tag{6}$$

For a given, and previously calculated, trajectory $\mathbf{x}_i(\cdot)$, with $i \in [1, N]$, it is therefore possible to estimate the probability of the event E_j by introducing the function $[\eta_j]$:

$$[\eta_j](\mathbf{x}_i(\cdot)) = \begin{cases} [1] & \text{if } \forall \mathbf{p}_j \in [\mathbf{p}_j], \exists t \in \mathbb{R}, h(\mathbf{x}_i(t), \mathbf{p}_j) \leq 0 \\ [0] & \text{if } \forall \mathbf{p}_j \in [\mathbf{p}_j], \forall t \in \mathbb{R}, h(\mathbf{x}_i(t), \mathbf{p}_j) > 0 \\ [0,1] & \text{otherwise} \end{cases} \tag{7}$$

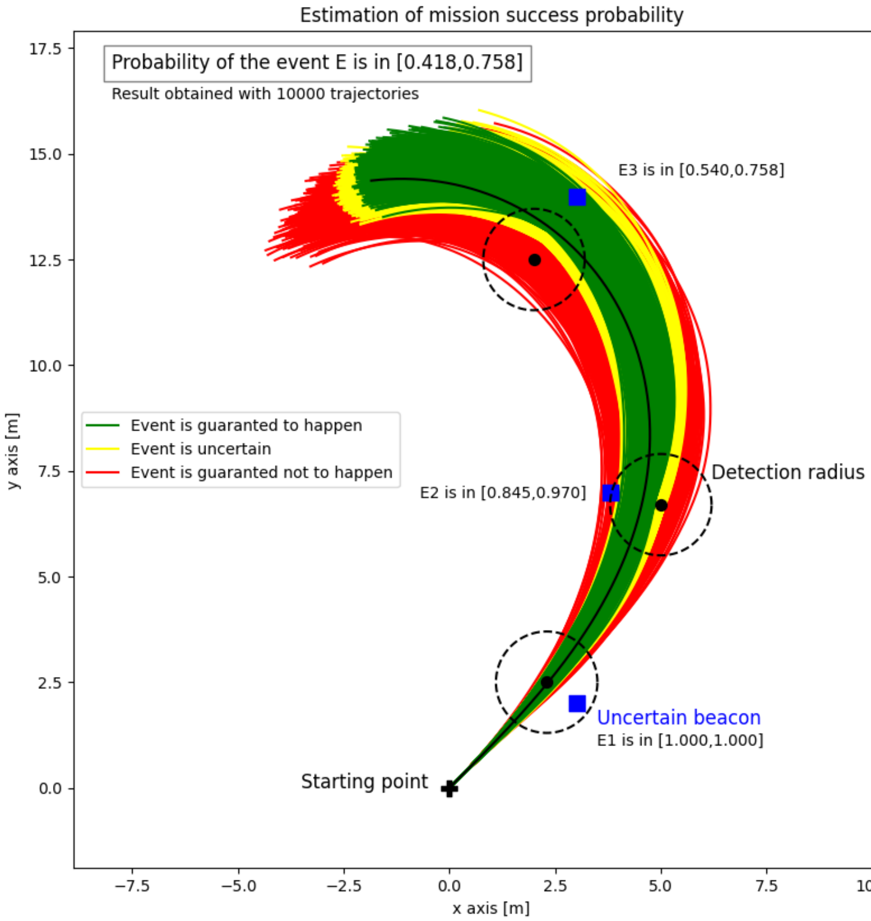


Figure 1: Simulation of the scenario

By calculating the value of $[\eta_j]$ for each trajectory $\mathbf{x}_i(\cdot)$ and then averaging, it is now possible to obtain an estimate of $[P_{E_j}]$.

In this scenario, it is even possible to go one step further and estimate the probability that the robot will detect all 3 beacons during a single trajectory $\mathbf{x}_i(\cdot)$. This mission is described by the event E, with $E = E_1 \wedge E_2 \wedge E_3$, and the function $[\eta_j]$ is replaced by the function $[\eta]$ where:

$$[\eta](\mathbf{x}_i(\cdot)) = [\eta_1](\mathbf{x}_i(\cdot)) \wedge [\eta_2](\mathbf{x}_i(\cdot)) \wedge [\eta_3](\mathbf{x}_i(\cdot)) \tag{8}$$

where $\wedge$ should be interpreted using the three valued logic.

On Propagating Uncertainties for Estimation and Association using Validated Interval Simulation

Charlotte Govignon[1], Elliot Brendel[2] and Julien Alexandre dit Sandretto[3]

[1] Ecole des Ponts ParisTech
Cité Descartes, 6 et 8 avenue Blaise-Pascal, 77420 Champs-sur-Marne, France
`charlotte.govignon@eleves.enpc.fr`
[2] Thales Land and Air Systems
Paris, France
`elliot.brendel@thalesgroup.com`
[3] ENSTA Paris, Institut Polytechnique de Paris
828 Boulevard des Maréchaux, Palaiseau 91120, France
`julien.alexandre-dit-sandretto@ensta-paris.fr`

Keywords: Estimation, Association, Satellite tracking, Potential clouds, Validated simulation

Introduction

Estimation and association are usually dealt with probalistic methods. However, these methods often require strong hypothesis on the distribution of uncertainties which are usually needed to be Gaussian. When such hypothesis can no longer be verified, validated numerical integration methods can provide useful informations on the considered uncertainties. Validated numerical integration methods, also named reachability analysis or guaranteed simulation, are interval counterparts of numerical integration methods [1] [2]. Further, by using intervals instead of probabilities, the notion of distribution density is lost. With confidence contractors, it becomes possible to create potential clouds, which are the validated counterparts of confidence intervals [3].

Basic properties

Definition 1 (Confidence contractor)**.**

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

with $[y]$ *defined such that* $Pr(x \in [y]) = \int_{[y]} f_X(x)\,dx = cc$ *(*$[y]$ *is the confidence interval),* $Pr(x \in [y])$ *stands for "probability that* x *is in* $[y]$*", with* x *following the distribution* f_X*, and* cc *being the confidence coefficient (*$0 \leq cc \leq 1$*).*

Main results

We apply our technique to satellite detection, association and tracking with promising results in simulation. A model of satellites in an Equinoctial frame is simulated from an initial observation. After a potentially long period, new measures are acquired and association to the simulated satellite is verified to decide if it is the same or a new object. If associated, the measures are used to improve the quality of the satellite estimation. Our approach provides, then, a rich solution (with guarantees and probabilities simultaneously) to the important problem of association and estimation in space object tracking.

To test our method, three scenarios were studied. The first scenario consists in the propagation of the 5 % and 95 % potential clouds of a satellite for 6 hours. Figure 1 shows the results on the equinoctial parameters p and L. For an easier comparison between the two potential clouds, the intervals are translated using the center of the 95 % interval. In the second scenario, the uncertainties of a satellite given its initial state were propagated for 12 hours, then a series of measures on its position were added to the simulation to improve the estimation. Figure 2 provides the evolution of the uncertainties of the six state parameters, with and without the added measures after 12 hours. In the last scenario, the 5 % and 95 % potential clouds are propagated using validated simulation for the two satellites. At last, the trajectories of the two satellites can be compared on Figure 3.

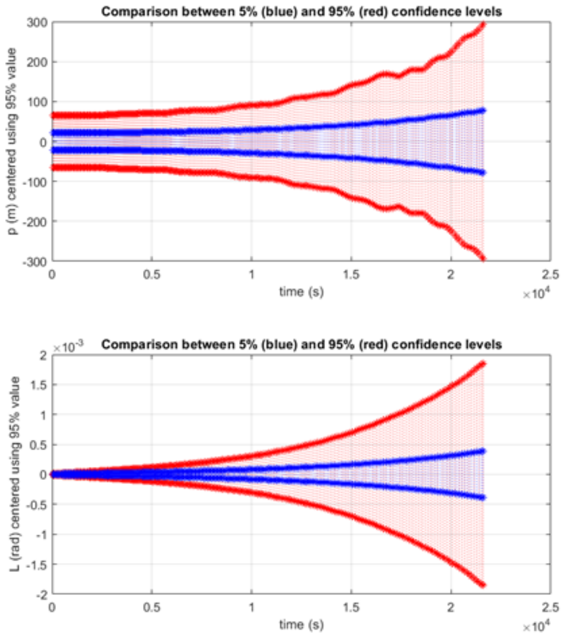


Figure 1: Propagation of the 5 % (blue) and 95 % (red) potential clouds on the p and L parameters with a simulated time of 6 hours.

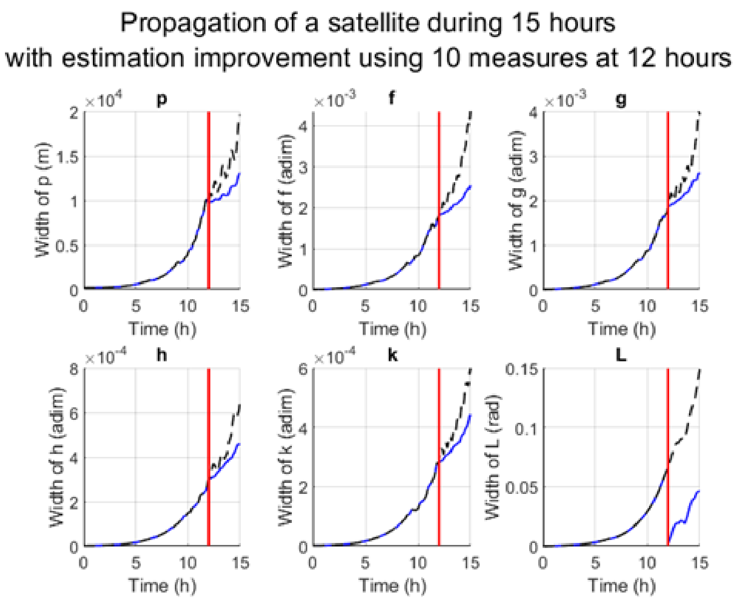
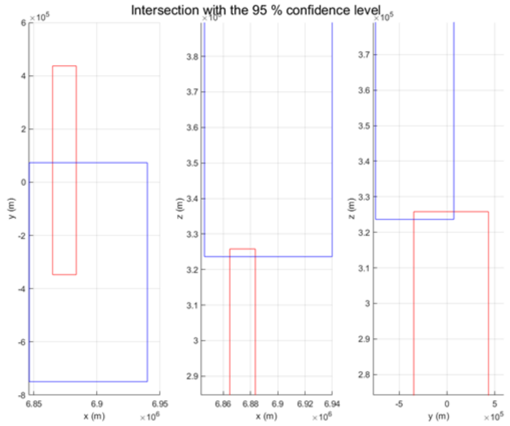
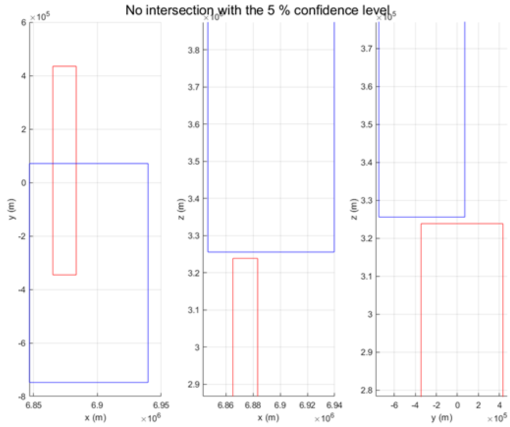


Figure 2: Propagation of the uncertainties of the state of a satellite for 15 hours, with addition of 10 measures (blue) and without the addition of measures (black) after 12 hours (red).



(a) Intersection of position boxes (x, y and z in the ECI frame) for the propagation of the 95 % potential cloud of two satellites.



(b) No intersection of position boxes (x, y and z in the ECI frame) for the propagation of the 5 % potential cloud of two satellites at the same time.

Figure 3: Comparison of position boxes of a same satellite for the propagation of the 5 % and 95 % potential clouds.

References

[1] N.S. Nedialkov, K.R. Jackson and G.F. Corliss. Validated solutions of initial value problems for ordinary differential equations. *Appl. Math. and Comp*, 105:21–68, 1999.

[2] J. Alexandre dit Sandretto and A. Chapoutot. Validated Explicit and Implicit Runge-Kutta Methods. *Reliable Computing*, 22, 2016.

[3] J. Alexandre dit Sandretto. Confidence-based Contractor, Propagation and Potential Clouds for Differential Equations. *Acta Cybernetica*, 25:49-68, 2021.

Introducing Box Chains to simplify Reachability Analysis

Maël GODARD[1], Luc JAULIN [1] and Damien MASSE[2]

[1] ENSTA Bretagne, Lab-STICC - ROBEX Team
1 rue Françis Verny, 29200 Brest, France
mael.godard@ensta-bretagne.org
lucjaulin@gmail.com
[2] Université de Bretagne Occidentale, Lab-STICC - ROBEX Team
20 avenue Victor le Gorgeu, 29200 Brest, France
damien.masse@univ-brest.fr

Keywords: Reachability, Interval Analysis, Box Chain

Introduction

Reachability Analysis is an important tool in robotics when it comes to ensure that a robot performs its mission safely. Performing a Reachability Analysis comes down to calculating the image of a disk D by a function f. In this work we will assume that for all x in D we have $det(J_f(x)) > 0$ where det is the determinant function and J_f the Jacobian matrix of f. This means that if we choose an orientation for the contour ∂D of D, it is conserved by the function f.

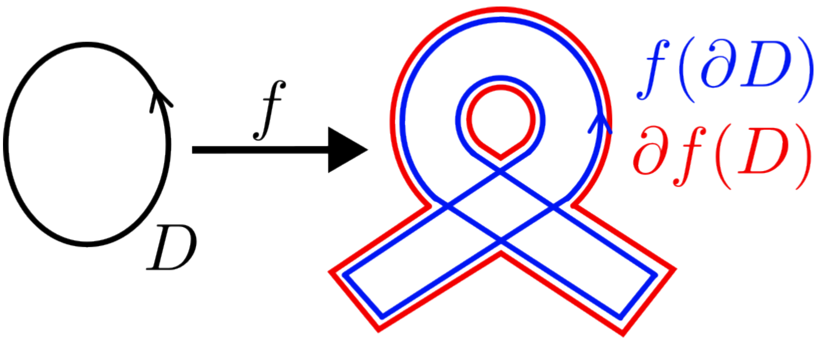


Figure 1: Reachability Analysis

Different methods [1] [2] allow us to estimate the reachable set of a robot at a given time. We will focus on the method presented in [3] which provides the contour of the Reachable set and its normal. We will see how we can efficiently apply it with Interval Analysis tools.

More specificly we will introduce the notion of "Box Chain" and see how it can be used to detect fake boundaries in two dimensions and greater. If we adopt the notations in fig 1 , eliminating fake boundaries comes down to finding $\partial f(D)$ from $f(\partial D)$.

Main results

We propose a definition of the Box Chain and use it to partition a contour in two dimensions. This partioning allows us to detect intersections in the contour, meaning that there are fake boundaries in it.

The main contribution of this work is to show how we can use these Box Chains to detect fake boundaries. This method uses the fact that we know the normal to the contour to first color the inside of the Reachable set, and then suppress the fake boundaries.

We will finally see how this method can be extended in higher dimensions.

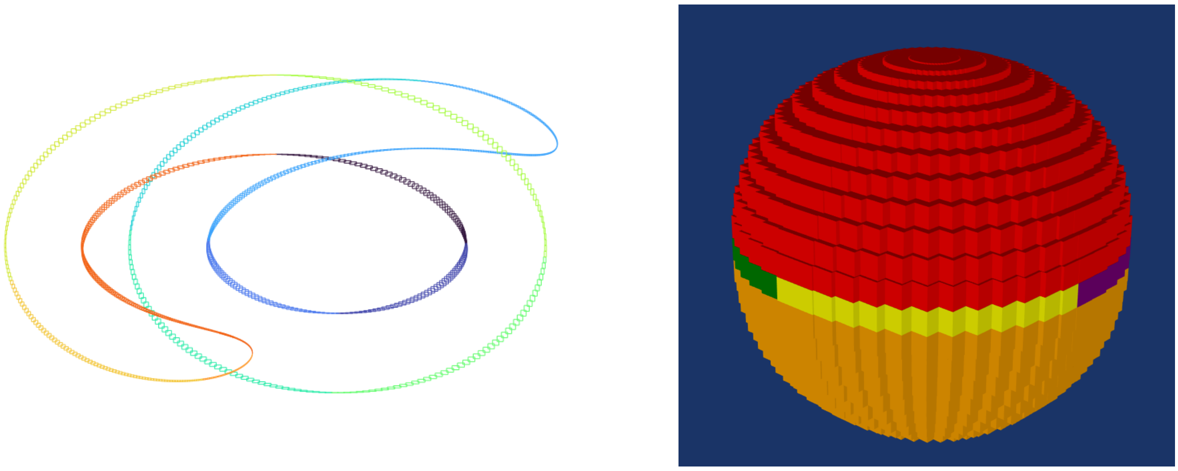


Figure 2: Box Chain decomposition of a 2D and a 3D contour

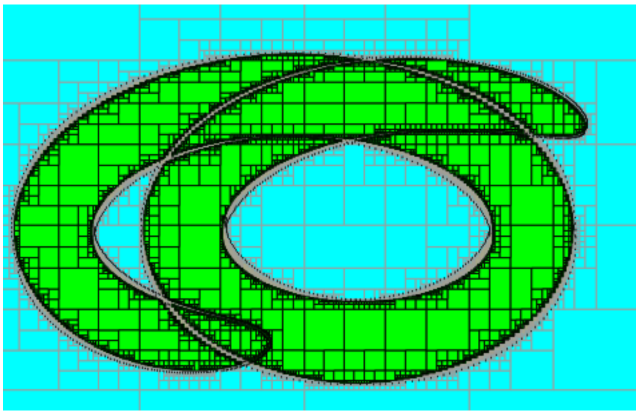


Figure 3: Inside of the Reachable set colored in green

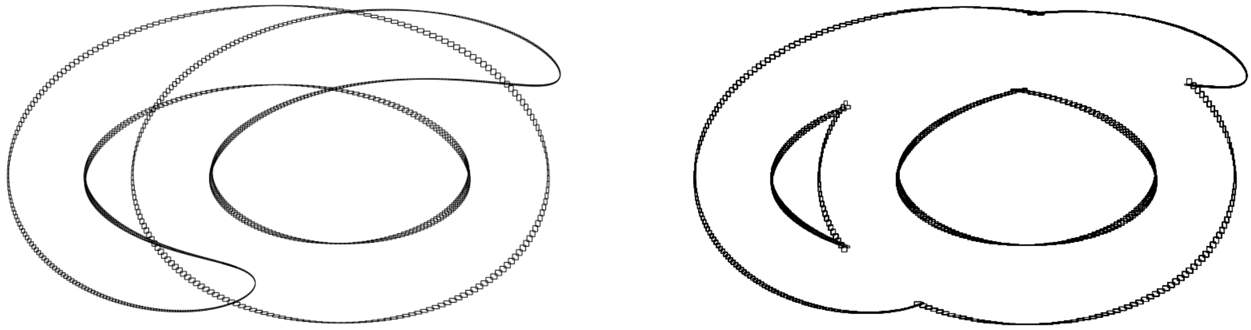


Figure 4: Eliminating fake boundaries

Acknowledgement

Thanks to the Brittany region and the AID.

References

[1] J. Damers, L. Jaulin and S. Rohou. *Lie symmetries applied to interval integration.* Automatica 2022.

[2] L. C. G. J. M. Habets, P. J. Collins, and J. H. van Schuppen. *Reachability and Control Synthesis for Piecewise-Affine Hybrid Systems on Simplices.* IEEE Transactions on Automatic Control, 2006.

[3] T. Lew, R. Bonalli and M. Pavone. *Exact Characterization of the Convex Hulls of Reachable Sets.* 62nd IEEE Conference on Decision and Control (CDC 2023), Dec 2023.

Outer approximation of the occupancy set left by a mobile robot

Luc Jaulin

Lab-STICC, ENSTA-Bretagne
2, rue F. Verny, Brest
lucjaulin@@gmail.com

Keywords: Occupancy set, Mobile robotics, Interval Methods

Introduction

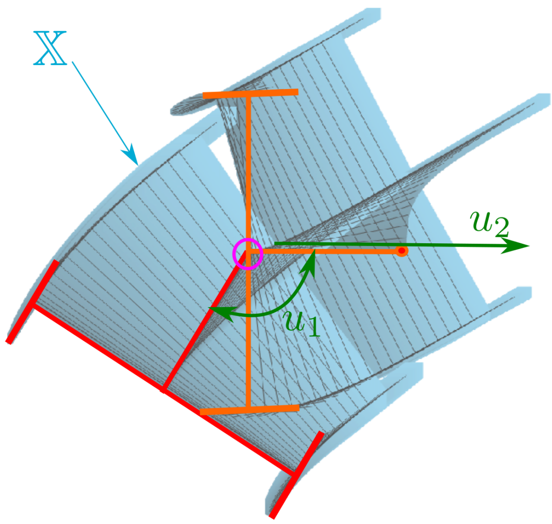
We consider a multi-body mobile robot described by a state equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, where the input $\mathbf{u} \in [\mathbf{u}]$ is uncertain. The robot has a given shape. For a given trajectory $\mathbf{x}(t)$, we define the *occupancy set* $\mathbb{X}$ as the set of all points $\mathbf{a}$ of a world that has been occupied by the robot at least once during the mission. It is defined by

$$\mathbb{X} = \left\{ \mathbf{a} \in \mathbb{R}^2 \,|\, \exists t \in [0, \ldots, t_{\max}], \exists i \,\text{s.t.}\; h_i(\mathbf{x}(t), \mathbf{a}) \leq 0 \right\}$$

where h_i is the shape function of the ith body.

Main results

We propose a new interval-based method to enclose $\mathbb{X}$. This is illustrated by the figure for a car-trailer. Here, the robot has two bodies (red and orange) with two inputs u_1 (rotation rate) and u_2 (the acceleration).



It is described by the following state equation

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{pmatrix} = \begin{pmatrix} x_5 \cos x_3 \\ x_5 \sin x_3 \\ u_1 + x_5 \sin(x_3 - x_4) \\ x_5 \sin(x_3 - x_4) \\ u_2 \end{pmatrix}$$

where x_1, x_2 are the coordinates of the center, x_3 is the heading of the first body, x_4 is the internal angle and x_5 is the speed.

The main contribution of this work is to show how to find a diffeomorphism on the state space to rewrite the system into a causal chain. The interval integration of the causal chain can then easily be done even in case of uncertainty in $\mathbf{u}$. In a second step, we show how to characterize $\mathbb{X}$ using an interval-based projection algorithm.

Optimal Path Planner over Receding Horizon using Interval Analysis

Lucas Si Larbi[1,2], Julien Alexandre dit Sandretto[1] and Eric Lucet[2]

[1] U2IS, ENSTA Paris, Institut Polytechnique de Paris
91120 Palaiseau, France
`julien.alexandre-dit-sandretto@ensta-paris.fr`
[2] Université Paris Saclay, CEA, List
F-91120 Palaiseau, France
{`lucas.silarbi,eric.lucet`}`@cea.fr`

Keywords: Mobile Robot, Local Path Planner, Receding Horizon, Non-Linear Problem, Global Optimization, Interval Method.

Introduction

The transition from global to local planning often represents a significant gap in outdoor navigation. The local path planner must allow for any unforeseen environmental constraints, such as new obstacles detected in the vicinity of the robot. The solution suggested here, to fill this gap, consists in generating a path optimized over a Receding Horizon (RH), based on waypoints generated by the global planner, using an Interval Branch and Bound (B&B) algorithm. This path is then followed by using a Model-based Predictive Controller (MPC) [1].

Related Work and Motivation

Michael Defoort [2] and then José Mendes Filho [3] contributed to the generation of an optimized trajectory over a RH. They chose to use B-splines as the solution support due to their properties: a local modification; a definition of the entire curve only with several control points (CPs); a setting of the degree of continuity of the curve, and therefore the smoothness of the path generated. Defoort formalized the problem as a nonlinear constrained problem. Non-interval solvers like Sequential Least Squares Quadratic Programming (SLSQP) [4] were used to find a solution. Two main issues were raised: the inability to guarantee a solution if one exists; and the inability to control computation time and therefore to ensure real-time (RT) performance. Interval methods applied to constrained global optimization make it possible to "find the global optimum and provide bound on its value and location" (Hansen [5]). Moreover, interval B&B algorithms can prematurely stop the optimization and provide a sub-optimal solution which respects constraints and therefore ensure RT performance.

Contribution

A formalization of the problem making it compatible with interval methods and a custom interval B&B algorithm are presented. The main idea is to optimize CPs, represented by interval vectors (2D boxes, one dimension per coordinate of CPs), of a B-spline in order to avoid obstacles (constraints) while minimizing a cost function (optimization). The optimal path over a RH is obtained with a succession of bisections and contractions. This step is then repeated after horizon displacement by retaining some of the optimized CPs of the previous B-spline to ensure continuous curvature at the connection and to respect the principle of generation over a RH.

First Result and Proof of Concept

Figure 1 shows the first result using a basic formalization of the problem and a default configuration of IbexOpt [6]. Circular obstacles are used to simplify the description of the environment. In further work they will be replaced by a paving of a binary cost map (obstacles/free space). A weighted cost map will be used to ensure the optimization. Moreover, uncertainties corresponding to control error margin or perception quality are taken into account by adding an error term $[-\tau, \tau]$

on constraints to guarantee that the path obtained is safe. For e.g. with $\tau = 0.2m$, an obstacle-free zone of 20 cm on each side of the path is guaranteed (see Figure 1.b).

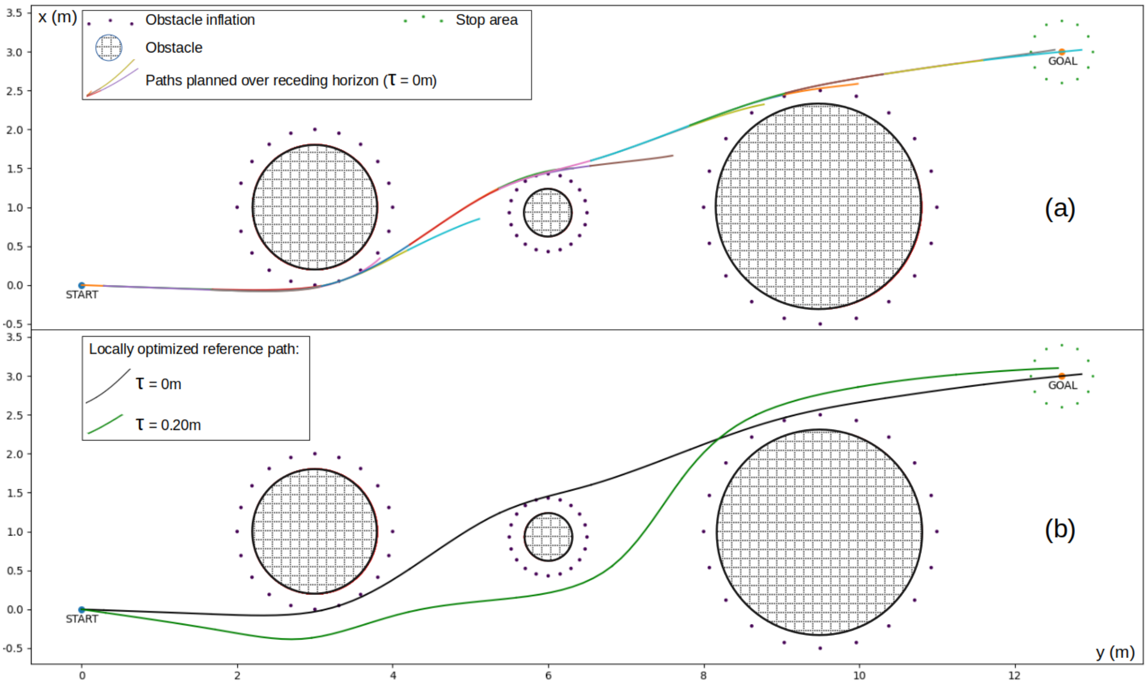


Figure 1: (a) Detailed view of all paths generated over successive RH. Each color corresponds to a B-spline arc. (b) View of two locally optimized reference paths, with two different τ, corresponding to the kept part of each of the paths generated over successive RH.

Acknowledgement

This work was carried out in the scope of MOBILEX project. This project received funding from the French National Research Agency (ANR) and the French Defense Innovation Agency (AID), in partnership with the French National Center for Space Studies (CNES) and the French Agency for Transport Innovation (AIT).

References

[1] Eric Lucet, Alain Micaelli, and François-Xavier Russotto. Accurate autonomous navigation strategy dedicated to the storage of buses in a bus center. *Robotics and Autonomous Systems*, 136:103706, 2021.

[2] Michael Defoort, Jorge Palos, Annemarie Kokosy, Thierry Floquet, and Wilfrid Perruquetti. Performance–based reactive navigation for nonholonomic mobile robots. *Robotica*, 27, 03 2009.

[3] José Mendes Filho and Eric Lucet. Multi-robot motion planning: A modified receding horizon approach for reaching goal states. volume 56, 09 2015.

[4] D. Kraft. *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988.

[5] E. Hansen and G.W. Walster. *Global Optimization Using Interval Analysis: Revised And Expanded*. 2004.

[6] Ibex Team. "ibexopt documentation". `https://ibex-team.github.io/ibex-lib/optim.html`.

Interval Methods applied to Signal Temporal Logic – Overview and Extension on Tubes

Joris Tillet, Julien Alexandre dit Sandretto and Elena Vanneaux

ENSTA Paris, Institut Polytechnique de Paris
828, boulevard des maréchaux, 91762 Palaiseau Cedex, France
{joris.tillet,julien.alexandre-dit-sandretto,elena.vanneaux}
@ensta-paris.fr

Keywords: STL, Interval Methods

Introduction

Designers and users of cyber-physical systems often require guarantees on the system's behavior, hence the need of a runtime verification process. This formal verification can be done using temporal logic, a formalism able to specify the system's requirements with temporal constraints. In 1977, Linear Temporal Logic (LTL) [9] is introduced, allowing to describe discrete time properties on a discrete signal. This is mainly used for formal verification of software or digital hardware, but reaches its limits when considering real-time constraints. When considering CPS, which are hybrid systems, man needs an extension of LTL able to handle real-time properties on real-value signal. This is the purpose of Signal Temporal Logic (STL) [8]. We consider an approach using interval analysis to ensure computation guarantees and take into account uncertainties in a robust manner.

STL is defined recursively by:

$$\phi := \mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \, \mathcal{U}_{[a,b]}\phi_2 \mid \top \tag{1}$$

with ϕ an STL formula, and $\mathcal{U}$ the *until* operator. The interval $[a, b]$ is defined with $a, b \in \mathbb{R}^+$. We have

$$(x, t) \vDash \phi_1 \, \mathcal{U}_{[a,b]} \phi_2 \iff \exists t' \in [a, b] \; (x, t + t') \vDash \phi_2 \text{ and } \forall t'' \in [t, t'] \; (x, t'') \vDash \phi_1. \quad (2)$$

μ is an atomic predicate: $\mu_x \equiv f(x_1(t), \ldots, x_n(t)) > 0$.

Several operators can be derived from STL, such as the well known finally operator $\mathcal{F}_{[a,b]}\phi \equiv \top \, \mathcal{U}_{[a,b]}\phi$, noted $\lozenge$, which is sometimes called *eventually*. The satisfaction of ϕ by a signal $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is written $(\mathbf{x}, t) \vDash \mu$ and is true if and only if we have $f(x_1(t), \ldots, x_n(t)) > 0$.

Monitoring using intervals

Related works

The qualitative satisfaction of an STL formula is a boolean value. This binary result can not contain an information of robustness of satisfaction. Quantitative satisfaction [5] introduces this notion of robustness to provide a measure of how much a trace satisfy or violate a given property. This approach can be used to handle some uncertainties, as in [4] where spatial and temporal uncertainties are considered, sometimes by the help of intervals. For instance, in [11], an offline monitoring algorithm with intervals for finite time STL formulas is proposed. In [3], an online monitoring is proposed using an interval of all the possible quantitative satisfaction of a partial signal with unbounded future.

In the literature, most of the researchers proposing STL monitors deal with single traces. However, in our approach, we want to go further by providing a robustness to numerical and model approximations. We then consider set of traces that can be represented by tubes (interval of trajectories).

Tubes

Interval STL is introduced in [1] to take uncertainties in predicates. Signals of intervals are considered, and quantitative satisfaction is returned as an interval. Then, depending on the inclusion or not of the zero value in this interval, the resulting satisfaction is a three-valued logical value: true, false, or undef. In [10], the authors cut the tubes in the time dimension and interpret STL formulas as successive discrete states in a new formalism called Reachset Temporal Logic. When there exists a trace in the set of trace that does not satisfy the constraint, then the proposed monitor returns false. The approach proposed in [6] is closer to [8]: it searches times when properties become true or false, and then propagates the information in a bottom up manner to deduce the result for the whole STL formula. In the latter, there is no completeness: the algorithm can return “unknown” when there is an ambiguity on the satisfaction or the violation of every trace.

Boolean interval extension

In our work, we propose to extend these previous results to boolean interval, such that the ambiguous “unknown” result is represented by the interval $[0, 1]$, and the ambiguity may be eliminated later if enough information are provided. Indeed, in the following example, both approaches from [10] and [6] lead to an unknown or false result, despite the signal does satisfy the formula.

Consider the tube $[x](t)$ represented in the Figure 1. We have a set point $\rho \in \mathbb{R}$ we want our system to reach in the time interval $[a, b] \subset \mathbb{R}^+$ (and not before). The corresponding STL formula Φ is:

$$\Phi \equiv \neg\phi \, \mathcal{U}_{[a,b]}\phi$$

with $\phi \equiv [x] \geq \rho$.

In this example, during the period $[a, b]$, the satisfaction of the predicate ϕ is ambiguous. However, the STL formula Φ is well satisfied by any trace taken in the tube $[x](t)$. In order to conclude the satisfaction rather than an unknown or false result, we propose to use

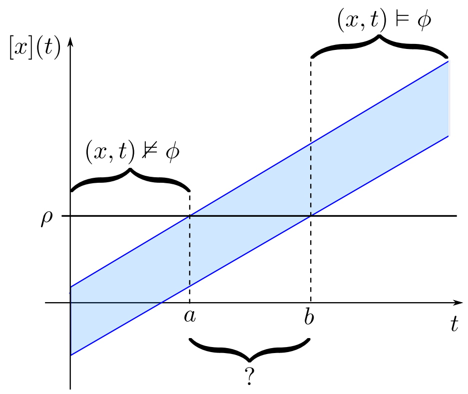


Figure 1: The tube is represented in blue. Before a and after b, there is no doubt on satisfaction or violation of the predicate $\phi \equiv x \geq \rho$. However, during the period $[a, b]$, some traces may satisfy ϕ while other not.

reachability analysis to obtain the result. The difficulty lies in the fact that the time t' from equation (2) may be different for every possible trace included in the tube, so there is an infinite number of trace and time to check.

The proposed solution is to use a branching algorithm on the time interval $[a, b]$, such that we can conclude the satisfaction of the formula when the set $[\mathbf{x}](0)$ is completely covered by the union of the inner backward reachsets starting from the set X_ϕ of states satisfying ϕ and from times $[t'] \subset [a, b]$ (see e.g. [7]).

Acknowledgement

The authors acknowledge support from the CIEDS[1] with the STARTS project.

References

[1] Luke Baird, Akash Harapanahalli, and Samuel Coogan. Interval signal temporal logic from natural inclusion functions. Conference Name: IEEE Control

[1] CIEDS: French Interdisciplinary Center for Defense and Security.

Systems Letters.

[2] Ezio Bartocci, et al.Maler, Dejan Ničković, and Sriram Sankaranarayanan. Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications. *Lectures on Runtime Verification*, volume 10457, pages 135–175. Springer.

[3] Jyotirmoy V. Deshmukh, et al. Robust online monitoring of signal temporal logic.

[4] Alexandre Donzé and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. *Formal Modeling and Analysis of Timed Systems*, volume 6246, pages 92–106. Springer.

[5] Georgios E. Fainekos and George J. Pappas. Robustness of temporal logic specifications for continuous-time signals.

[6] Daisuke Ishii, Naoki Yonezaki, and Alexandre Goldsztejn. Monitoring temporal properties using interval analysis.

[7] Alexander B. Kurzhanski and Pravin Varaiya. *Dynamics and Control of Trajectory Tubes: Theory and Computation*, volume 85 of *Systems & Control: Foundations & Applications*. Springer International Publishing.

[8] Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pages 152–166. Springer.

[9] Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 46–57.

[10] Hendrik Roehm, et al. STL model checking of continuous and hybrid systems. *Automated Technology for Verification and Analysis*, Lecture Notes in Computer Science, pages 412–427. Springer.

[11] Bingzhuo Zhong, Claudius Jordan, and Julien Provost. Extending signal temporal logic with quantitative semantics by intervals for robust monitoring of cyber-physical systems.

Guaranteed integration on Lie groups

Damien Massé[1]

[1] University of Brest, Lab-STICC, Robex team
20 avenue Le Gorgeu, 29200 Brest, France
damien.masse@univ-brest.fr

Keywords: Lie Groups, Guaranteed Integration, Interval Methods

Introduction

A Lie group $(G, \cdot)$ is a manifold G with an differentiable internal operator $\cdot$ defining a group structure. Lie groups are used in robotics [3] as convenient ways to represent simutaneously states (as a continuous manifold) and transformations on states (with the operator).

To each Lie group G is assocatied a Lie algebra $\mathfrak{g}$ which represents the vector space tangent to G at the neutral element Id and, by extension, at any element g of G.

Hence a differential equation on a Lie group G is expressed as:

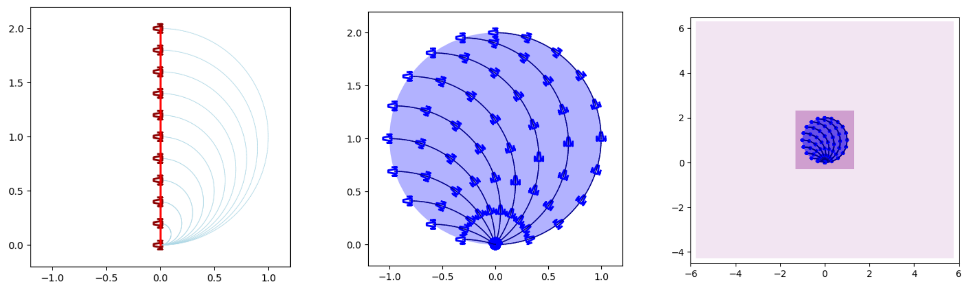
$$\dot{R} = f(R, t) \quad R(t_0) = R_0 \tag{1}$$

where $f : G \times \mathbb{R} \to \mathfrak{g}$. As an element of $\mathfrak{g}$, $f(R, t)$ represents the evolution of R in the local tangent space of G near R.

Our goal is to compute an guaranteed approximation of R, when G is a matrix Lie group, given uncertainties on the values of f. We specifically consider here the case where f depends only on t.

Contribution

G is a matrix Lie group when its elements can be represented as invertible matrices. In this case, the solution of the differential equation



(a) (b) (c)

Figure 1: Results of differential equation in the Lie Group SE(2) (representing the pose of a 2D car). The car has a constant rotation speed to the left and a variable speed forward. (a) Trajectories and final states for constant $\dot{R}$ (i.e. $e^{[\mathbf{v}]}$). (b) Trajectories and final states for variable $\dot{R}$ (i.e. $\mathrm{Exp}([\mathbf{v}])$). (c) Approximations of $\mathrm{Exp}([\mathbf{v}])$ with boxes, using the Taylor sum (light purple) and scaling-and-squaring (dark purple).

$\dot{R} = f(t)$ is expressed as a right product integral for matrix functions [2]:

$$R(t) = R(0) \cdot (\mathrm{Id} + f(\tau)d\tau) \prod_{0}^{t} \tag{2}$$

When $f = \mathbf{v}$ is constant, the exact result is:

$$R(t) = R(0) \cdot e^{t\mathbf{v}} \tag{3}$$

When f varies inside a box $[\mathbf{v}]$, we consider a new operator Exp defined as:

$$\mathrm{Exp}([\mathbf{v}]) = \{ \ (\mathrm{Id} + f(\tau)d\tau) \prod_{0}^{1} \ | \ f : [0,1] \to \mathbf{v}\} \tag{4}$$

Figure 1 (a) and (b) illustrates the difference between $e^{[\mathbf{v}]}$ and $\mathrm{Exp}([\mathbf{v}])$ with an example from the Lie Group SE(2).

We use the following properties to approximate $\mathrm{Exp}([\mathbf{v}])$:

1. Exp([**v**]) can be approximated with an unoptimised Taylor sum:

$$\text{Exp}([\mathbf{v}]) \subseteq \text{Id} + \frac{[\mathbf{v}]}{1!} + \frac{[\mathbf{v}][\mathbf{v}]}{2!} + \frac{[\mathbf{v}][\mathbf{v}][\mathbf{v}]}{6!} + \dots \quad (5)$$

2. Scaling-and-squaring [1] can be used to refine the approximation of Exp([**v**]):

$$\text{Exp}(2[\mathbf{v}]) = \text{Exp}([\mathbf{v}]) \cdot \text{Exp}([\mathbf{v}]) \quad (6)$$

These results enable to compute a guaranteed and relatively accurate approximation of Exp([**v**]) (Fig.1 (c) gives an example).

Acknowledgement

The author wishes to thank Luc Jaulin for his comments and suggestions.

References

[1] Alexandre Goldsztejn and Arnold Neumaier. On the exponentiation of interval matrices. *Reliab. Comput.*, 20:53–72, 2014.

[2] Antonín Slavík. *Product integration, its history and applications.* 01 2007.

[3] Joan Solà, Jérémie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics. *CoRR*, abs/1812.01537, 2018.

Exact Real Arithmetic and the Efficiency of Taylor Models

Norbert Th. Müller[1]

Universität Trier, Germany
mueller@uni-trier.de

Keywords: Taylor models, logistic map, exact real arithmetic

Abstract

An essential aspect of software for exact real arithmetic (ERA) is the way how it deals with the composition of functions. The resulting error propagation depends on the way intermediate results are evaluated, which could be, for example, with naive interval arithmetic or with more elaborate applications of intervals like in Taylor models [MB01].

In this short note we want to discuss aspects of this error propagation in a quite special application of ERA: the computation of relatively long trajectories in iterated function systems. As a prominent example for such systems we will concentrate on the logistic map

$$L(x) = \mu \cdot x \cdot (1 - x)$$

with different control parameters $\mu \in [2, 4]$ [CE80] which has been discussed in many papers in the field of ERA, like [Bla05, Spa10, Spa14].

Although Taylor models have an improved error propagation compared to naive interval arithmetic, we present cases where implementations based on the approximation methods must have the same asymptotic computational complexity, so the gain from using Taylor models

[1] Joint work with F. Brauße <franz.brausse@manchester.ac.uk> and M. Korovina <rita.korovina@gmail.com>.
This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 731143. The research leading to these results has received funding from the DFG grant WERA MU 1801/5-1.

is reduced to just a factor. This is essentially due to the following property of the computational complexity $M(n)$ of multiplication of n-bit integers:

$$M(c \cdot n) \in O(M(n))$$

This property implies that even significant differences in the error propagation are asymptotically almost irrelevant. In consequence, for both approaches repelling fixpoints of the logistic map lead to a complexity of $O(n \cdot M(k+n))$ for the computation of n iterates with a final precision of k bits.

On the other hand, linear Taylor models exhibit an asymptotically better behavior for stable fixpoints: Here the complexity is reduced to $O(n \cdot M(k))$, while naive intervals still remain at $O(n \cdot M(k+n))$.

References

[Bla05] Jens Blanck. Efficient exact computation of iterated maps. *J. Log. Algebr. Program.*, 64(1):41–59, 2005.

[CE80] Pierre Collet and Jean-Pierre Eckmann. *Iterated maps on the interval as dynamical systems.* Progress in physics 1. Birkhäuser, Basel, 1980.

[MB01] K. Makino and M. Berz. Higher order verified inclusions of multidimensional systems by Taylor models. *Nonlinear Analysis: Theory, Methods & Applications*, 47(5), 2001. Proceedings of the Third World Congress of Nonlinear Analysts.

[Spa10] Christoph Spandl. Computational Complexity of Iterated Maps on the Interval (Extended Abstract). In *Proceedings Seventh International Conference on Computability and Complexity in Analysis, CCA 2010, Zhenjiang, China.*, 2010.

[Spa14] Christoph Spandl. True orbit calculations of dynamical systems forvalidating molecular dynamics simulations. slides CCA 2014, 2014.

Reinforced Set Projection Algorithm

Nuwan Herath Mudiyanselage[1], Luc Jaulin[1]
and Simon Rohou[1]

[1] ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France
2 rue François Verny, 29806 Brest Cedex 09, France
lucjaulin@gmail.com,{nuwan.herath,simon.rohou}@ensta-bretagne.fr

Keywords: Set description, Set projection, Contractors, Separators

Introduction

With interval arithmetic, solution sets to given problems can be described by operators called contractors and separators. First, we recall what they are.

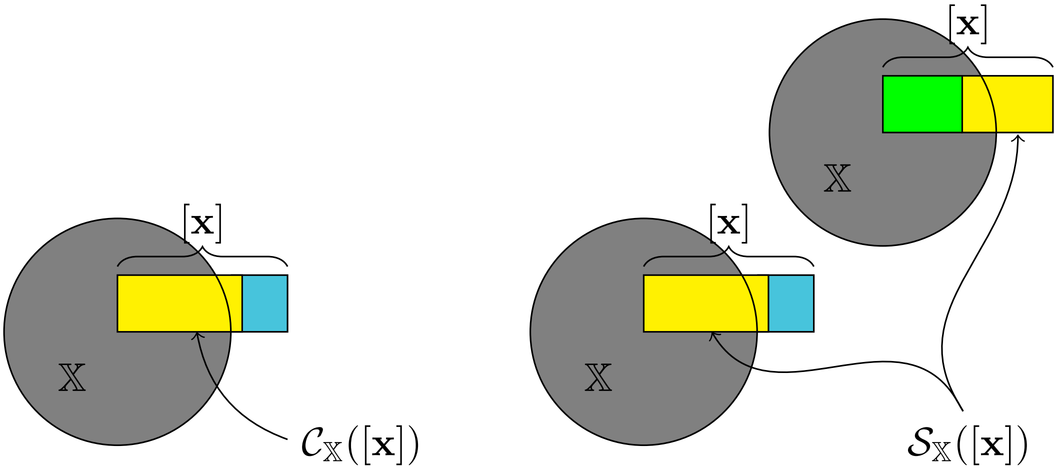
Set descriptors: from contractors to separators

Let us say that our solution set is $\mathbb{X}$. There exist interval operators to describe it [1]. $\mathcal{C}_{\mathbb{X}}$ denotes the contractor that describes the set $\mathbb{X}$. $\mathbb{IR}$ denotes the intervals of $\mathbb{R}$. One can apply the contractor on the box $[\mathbf{x}] \in \mathbb{IR}^n$ and get $\mathcal{C}_{\mathbb{X}}([\mathbf{x}])$, as illustrated by Figure 1a. $\mathcal{C}_{\mathbb{X}}$ verifies two properties:

$$\mathcal{C}_{\mathbb{X}}([\mathbf{x}]) \subset [\mathbf{x}] \text{ (contractance) and } \mathcal{C}_{\mathbb{X}}([\mathbf{x}]) \cap \mathbb{X} = [\mathbf{x}] \cap \mathbb{X} \text{ (correctness).}$$

One can guarantee that $[\mathbf{x}] \setminus \mathcal{C}_{\mathbb{X}}([\mathbf{x}]) \not\subset \mathbb{X}$. It is then possible to construct a paving made of blue boxes that do not contain points from $\mathbb{X}$ and yellow boxes that may contain points from $\mathbb{X}$ (see Figure 2a). The latter form an exterior approximation denoted by $\mathbb{X}^+$.

Separators simultaneously provide inner and outer approximations of the set $\mathbb{X}$, as illustrated by Figure 1b. Thus, one can also identify green boxes that are contained in $\mathbb{X}$ (see Figure 2b). They form an



(a) Contractor (b) Separator

Figure 1: Contractor $\mathcal{C}_{\mathbb{X}}$ and separator $\mathcal{S}_{\mathbb{X}}$ applied on the box $[\mathbf{x}]$.

interior approximation denoted by $\mathbb{X}^-$. Green and yellow boxes form $\mathbb{X}^+$. We then have an enclosure: $\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+$.

Set projection separators

In some applications, one may only be interested in the projection of the solution set [2]. We have a separator for that operation: `SepProj` in the Codac library [3].

Reinforced projection separators

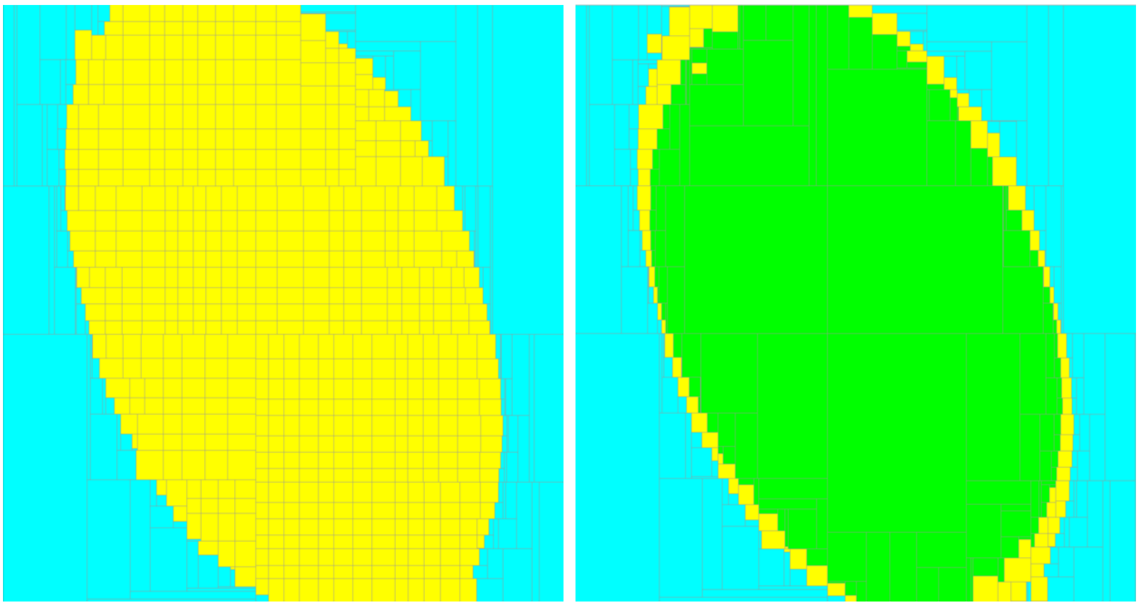
Let us look at the projection along the z-axis of the set $\mathbb{X}$ defined by

$$2x^2 + 2.2xy + xz + y^2 + z^2 \leq 10. \tag{1}$$

By defining $f(x, y, z) = 2x^2 + 2.2xy + xz + y^2 + z^2 - 10$, it can be written as

$$f(x, y, z) \leq 0. \tag{2}$$

The current implementation of the separator of the projection requires fine-tuning. Without the proper parameters, it produces bad quality



(a) Contractor $\to \mathbb{X}^+$ (b) Separator $\to \mathbb{X}^-, \mathbb{X}^+$

Figure 2: Pavings of the set $\mathbb{X} = \{x, y \in \mathbb{R}^2 \,|\, 2x^2 + xy + y^2 \leq 1\}$ for the two classes of descriptors.

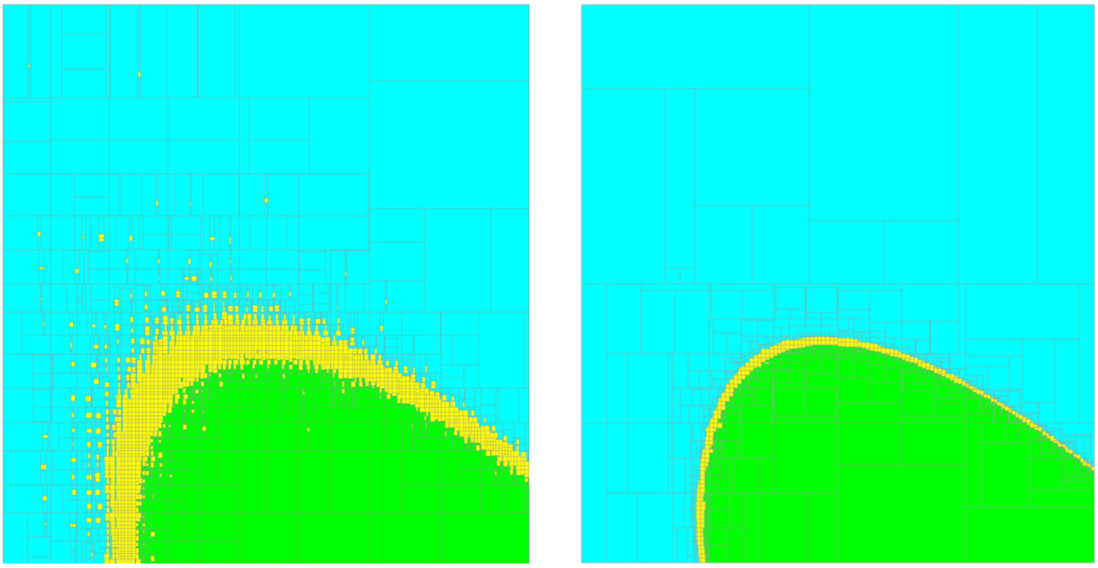
boundaries for our particular problem (see Figure 3a). The approximation is not minimal due to pessimistic results coming from interval dependency. Indeed, Equation 1 has multiple occurrences of the same variable. We present a new approach for differentiable sets which focuses on the boundary (see Figure 3b).

We reinforce the separator on $\partial \operatorname{Proj} \mathbb{X}$, the boundary of the projection. For our example, we use the knowledge of the locii of the vertical tangents to $\mathbb{X}$ which are defined by

$$\begin{cases} f(x, y, z) = 0, \\ \dfrac{\partial f}{\partial z}(x, y, z) = 0. \end{cases} \tag{3}$$

In our case, that is

$$\begin{cases} 2x^2 + 2.2xy + xz + y^2 + z^2 = 10, \\ x + 2z = 0. \end{cases} \tag{4}$$



(a) `SepProj` (b) Reinforced projection

Figure 3: Pavings of the projection along the z-axis of the set $\mathbb{X} = \{x, y, z \in \mathbb{R}^3 \mid 2x^2 + 2.2xy + xz + y^2 + z^2 \leq 10\}$.

`SepProj` was constructed from $\mathcal{S}_{\mathbb{X}}$ based on Equation 1. For the reinforced projection algorithm, we add $\mathcal{C}_{\partial\,\mathrm{Proj}\,\mathbb{X}}$ based on the knowledge of Equation 4.

References

[1] Gilles Chabert and Luc Jaulin. Contractor programming. *Artificial Intelligence*, Volume 173, Issue 11, 2009.

[2] Benoît Desrochers. Simultaneous Localization and Mapping in Unstructured Environments. Thesis, 2018.

[3] Simon Rohou, Benoît Desrochers and Fabrice Le Bars. The Codac Library. *Acta Cybernetica*, Mar. 2024.

Solving non-linear constraints in CDCL-style

Franz Brauße[1], Konstantin Korovin [1], Margarita Korovina[2] and Norbert Th. Muller[3]

[1] The University of Manchester, UK
{franz.brausse,konstantin.korovin}@manchester.ac.uk
[2] A.P. Ershov Institute of Informatics Systems, Novosibirsk, Russia
rita.korovina@gmail.com
[3] Universität Trier, Germany
mueller@uni-trier.de

Keywords: Logic in Computer Science, Continuous Constraints, iRRAM-computations

Abstract

We give a detailed overview of the **ksmt** calculus developed in a conflict driven clause learning framework for checking satisfiability of non-linear constraints over the reals. Non-linear constraint solving naturally arises in the development of formal methods for verification of safety critical systems, program analysis and information management. Implementations of formal methods are widely used to approve in advance that designed systems satisfy all specification requirements, such as reliability, safety and reachability. Historically, there have been two main approaches to deal with non-linear constrains: the symbolic one originated by Tarski's decision procedure for the real closed fields and the numerical one based on interval constraint propagations. It is well known that both approaches have their strength and weakness concerning completeness, efficiency and expressiveness. Nowdays, merging strengths of symbolical and numerical approaches is one of the challenging research areas in theoretical and applied computer science.

The **ksmt** calculus successfully integrates strengths of symbolical and numerical methods. The key steps of the decision procedure based on this calculus contain assignment refinements, inferences of linear resolvents driven by linear conflicts, backjumping and constructions of local linearisations of non-linear components initiated by non-linear conflicts. In [1] we showed that the procedure is sound and makes progress by reducing the search space. This approach is applicable to a large number of constraints involving computable non-linear functions, piecewise polynomial splines, transcendental functions and solutions of polynomial differential equations. In [2, 3] we proved, among other results, that **ksmt** is a δ-complete decision procedure for bounded problems. In this setting we discuss recent and future research work.

References

[1] F. Brauße, K. Korovin, M. Korovina, and N. Th. Muller. A CDCL-style calculus for solving non-linear constraints. *Frontiers of Combining Systems - 12th International Symposium, FroCoS 2019, London, UK, September 4-6, 2019, Proceedings*, ed. Andreas Herzig and Andrei Popescu, LNCS (LNAI), Springer, 11715: 131–148, 2019.

[2] F. Brauße, K. Korovin, M. Korovina, and N. Th. Muller. The ksmt Calculus Is a δ-complete Decision Procedure for Non-linear Constraints. *The 28th International Conference on Automated Deduction, CADE 2021*, ed. Andrè Platzer, Geoff Sutcliffe, LNCS, Springer, 12699: 113-130, 2021.

[3] F. Brauße, K. Korovin, M. Korovina, and N. Th. Muller. The ksmt Calculus Is a δ-complete Decision Procedure for Non-linear Constraints. *Theor. Comput. Sci.*, 975: 114–125, 2023.

An aymptotic minimal contractor for non-linear equations using the Codac library

Simon Rohou

ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France
simon.rohou@ensta-bretagne.fr

Keywords: non-linear systems, contractors, centered form

Introduction

We consider the problem of approximating the solutions of the system $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, where $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^p$ is a non-linear function. In particular, we will consider systems where $p < n$ for which the solution set $\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{f}(\mathbf{x}) = \mathbf{0}\}$ has infinitely many solutions.

Interval methods can be used to over-approximate such sets in a reliable way. They are often based on axis-aligned boxes $[\mathbf{x}] \in \mathbb{IR}^n$. When sets are described as non-linear systems, such as $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, natural inclusion functions can be used to easily and reliably evaluate boxes $[\mathbf{f}]([\mathbf{x}])$. They can be employed in branch-and-prune algorithms in order to pave the solution set $\mathbb{X}$ more accurately. However, these methods involve bisections which comes with an exponential complexity with respect to n. *Contractor* operators, often designed with polynomial complexity, have been shown to overcome this issue by using bisections as a last resort in exploration algorithms [2].

Contractors

A contractor on a set $\mathbb{X}$, denoted by $\mathcal{C}_{\mathbb{X}}$, is an operator that aims at narrowing a box $[\mathbf{x}] \in \mathbb{IR}^n$ in order to reliably remove vectors of $[\mathbf{x}]$

that are not part of the set $\mathbb{X}$. Algorithms exist to automatically build contractors for a given non-linear equation; the state-of-the-art on this topic is the `HC4Revise` algorithm [1].

As for natural inclusion functions, the efficiency of an `HC4Revise` contractor will be impacted by multi-occurrences in the analytic expression of $\mathbf{f}$. Solutions exist, such as symbolic rewriting or affine arithmetic [6], but they are not always minimal, difficult to use, or based on complicated algorithms. On the other hand, the use of centered-form computations provides asymptotically minimal results, as shown in [3].

Centered form contractor

The centered form approach improves the contractions by involving the Jacobian of $\mathbf{f}$ as expressed in Equation (1), where $\overline{\mathbf{x}}$ is the center of the box $[\mathbf{x}]$:

$$\mathbf{f}_c([\mathbf{x}]) = \mathbf{f}(\overline{\mathbf{x}}) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\big([\mathbf{x}]\big)\big([\mathbf{x}] - \overline{\mathbf{x}}\big). \tag{1}$$

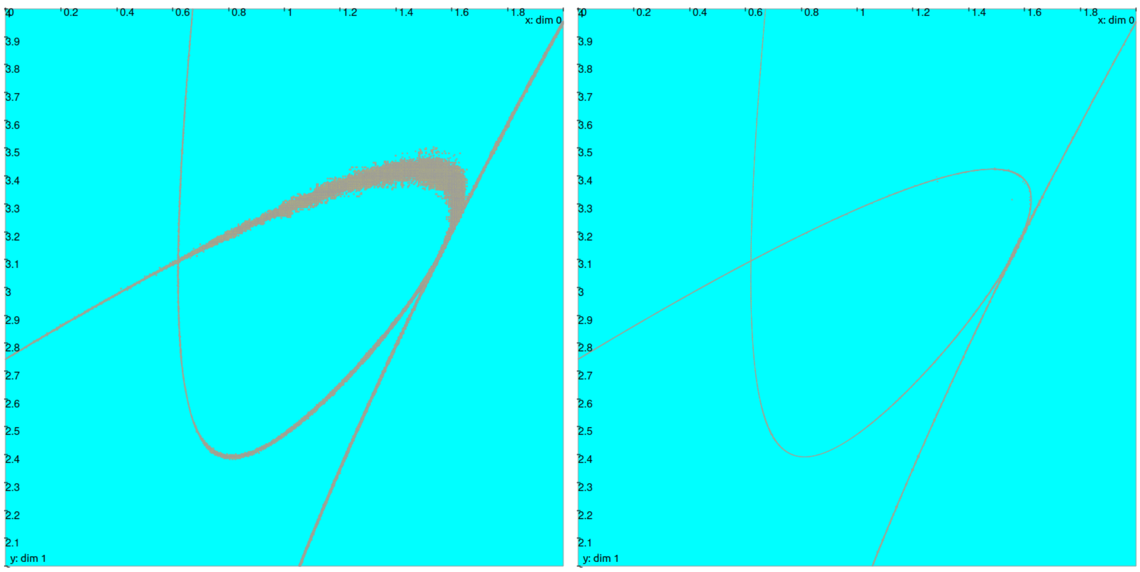
We propose an automatic way to obtain centered form contractors for non-linear systems. First, the interval Jacobian $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\big([\mathbf{x}]\big)$ is computed using Automatic Differentiation. Then, the matricial expression of Equation (1) is treated using an efficient linear contractor with preconditioning.

Implementation and results

The efficiency of the proposed contractor, called `CtcInverse`, will be illustrated on several examples. One of them is taken from the literature [4] and given by the following function $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^2$:

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} -x_3^2 + 2x_3 \sin(x_3 x_1) + \cos(x_3 x_2) \\ 2x_3 \cos(x_3 x_1) - \sin(x_3 x_2) \end{pmatrix}. \tag{2}$$

The solution set $\mathbb{X}$ of Equation (2) for $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ is illustrated in Figure 1 for $[\mathbf{x}_0] = [0,2] \times [2,4] \times [0,10]$ and $\epsilon = 4 \times 10^{-3}$. A comparison between the two contractors `HC4Revise` and `CtcInverse` is given: these two algorithms are provided in Codac and are based on the same elementary reverse operations provided by the GAOL library. The computation time difference is mainly due to the number of boxes: `CtcInverse` allows asymptotically minimal contractions for small boxes thanks to the centered form, and so a thinner and thus faster approximation of $\mathbb{X}$.



(a) $\mathbb{X}$ computed with `HC4Revise`. Computation time: 4.51s. 27430 boxes.

(b) $\mathbb{X}$ computed with `CtcInverse`. Computation time: 0.69s. 3713 boxes.

Figure 1: Example of set inversion of Equation (2) using the state-of-the-art `HC4Revise` and the proposed `CtcInverse` contractors. The approximated three-dimensional solution sets are projected onto (x_1, x_2).

`CtcInverse` is now available in the Codac library [5] (v2.0). In particular, the code of Figure 1b is given in Figure 2 as an example of use of Codac. The user does not have to provide the Jacobian of Equation (1), it is deduced by Automatic Differentiation. Codac is available in C++, Python and Matlab languages and provided under GNU LGPL. More information on: `http://codac.io`

```
from codac import * # using Codac 2.0 at least

x = VectorVar(3)
f = AnalyticFunction([x], vec(
  -sqr(x[2])+2*x[2]*sin(x[2]*x[0])+cos(x[2]*x[1]),
  2*x[2]*cos(x[2]*x[0])-sin(x[2]*x[1])
))

ctc = CtcInverse(f, [[0],[0]])
pave([[0,2],[2,4],[0,10]], ctc, 0.004)
```

Figure 2: Inversion of Eq. (2) using the Codac library (here in Python).

References

[1] F. Benhamou, F. Goualard, L. Granvilliers, and J. F. Puget. Revising hull and box consistency. In *Proceedings of the International Conference on Logic Programming*, pages 230–244, 1999.

[2] Gilles Chabert and Luc Jaulin. Contractor programming. *Artificial Intelligence*, 173(11):1079–1100, July 2009.

[3] Luc Jaulin. Asymptotically minimal interval contractors based on the centered form; application to the stability analysis of linear time-delayed differential equations. *Acta Cybernetica*, 2024.

[4] Rachid Malti, Milan R. Rapaić, and Vukan Turkulov. A unified framework for exponential stability analysis of irrational transfer functions in the parametric space. *Annual Reviews in Control*, 57:100935, 2024.

[5] Simon Rohou, Benoît Desrochers, and Fabrice Le Bars. The Codac Library. *Acta Cybernetica*, Mar. 2024.

[6] Siegfried M. Rump and Masahide Kashiwagi. Implementation and improvements of affine arithmetic. *Nonlinear Theory and Its Applications, IEICE*, 6(3):341–359, 2015.

Dimensioning a torpedo-like AUV using interval analysis

Quentin Brateau, Fabrice Le Bars, and Luc Jaulin

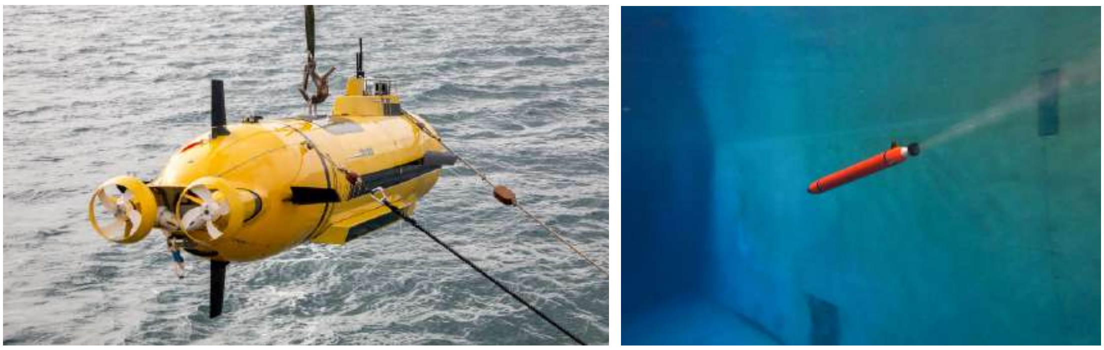
ENSTA Bretagne, UMR 6285, Lab-STICC, IAO, ROBEX
2 rue François Verny, 29806 Brest CEDEX 09, France

quentin.brateau@ensta-bretagne.org
fabrice.le_bars@ensta-bretagne.org
lucjaulin@gmail.com

Keywords: Dimensioning, Contractor, Set Inversion

Introduction

Solving a scaling problem saves time when designing a robotic platform. By analyzing the characteristics of already existing systems, we can deduce a set of parameters that must be respected between the physical quantities of our new system.



(a) Daurade [5] (b) Riptide

Figure 1: Example of torpedo-like AUVs

Torpedo-like AUV dimensioning

We consider the problem of dimensioning a torpedo-like AUV (Autonomous Underwater Vehicle). The goal is to design a new robot capable of operating in the trial pool we have at ENSTA Bretagne.

First, a set of equations involving the characteristic quantities of such a robot [1, 2, 5] and a set of parameters need to be expressed to solve this problem.

Then using examples of torpedo-like AUV characteristics, and contractors [3, 4], the set of parameters compatible with these example can be estimated.

Finally, from this estimated set of parameters, and imposed physical quantities on the new robot, the remaining unknown physical quantities to design our new AUV could be deduced using a set inversion algorithm [3, 4].

References

[1] T.I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. 2011.

[2] L. Jaulin. *Mobile Robotics*. 2019.

[3] Luc Jaulin, Michel Kieffer, Olivier Didrit, Eric Walter, Luc Jaulin, Michel Kieffer, Olivier Didrit, and Éric Walter. *Interval analysis*. Springer, 2001.

[4] Simon Rohou, Benoit Desrochers, and Fabrice Le Bars. The Codac library. *Acta Cybernetica*, Mar. 2024.

[5] Simon Rohou, Luc Jaulin, Lyudmila Mihaylova, Fabrice Bars, and Sandor Veres. Guaranteed computation of robot trajectories. *Robotics and Autonomous Systems*, 93, 03 2017.

Ultra-Wideband based Smart Wheelchair Localization using Interval Analysis

Théo Le Terrier[1], Vincent Drevelle[1], Marie Babel[1]

[1] Univ Rennes, Inria, CNRS, IRISA
{theo.le-terrier, vincent.drevelle, marie.babel}@irisa.fr

Keywords: Intervals, Robotics, Localization, Ultra-WideBand

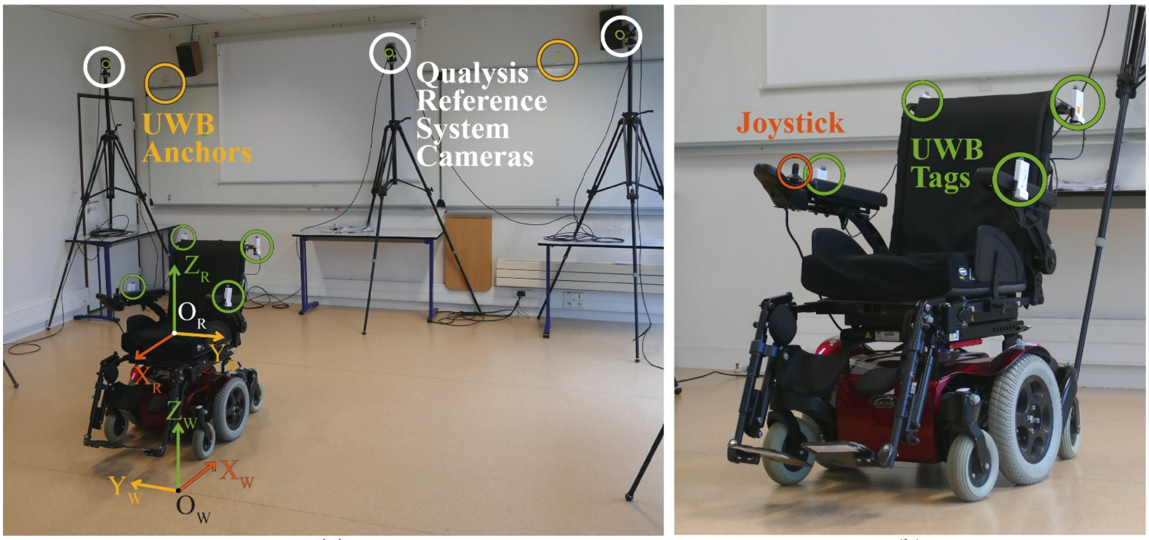
Introduction

Autonomous navigation for Smart Wheelchairs (SW) [1] enables to enhance the user's autonomy, with, e.g. autonomous docking to a charging station after going to bed. This requires reliable indoor localization. We aim at providing a confidence domain of the pose of a SW, i.e. a box which is guaranteed to contain the true wheelchair position and orientation. The proposed indoor positioning system relies on Ultra-WideBand (UWB) radio modules, with ranging capabilities.

Ultra-WideBand Sensors

A typical UWB-based indoor localization setup is a set of fixed UWB nodes in the room, or *anchors*, from which mobile UWB nodes installed on the robot, or *tags*, measure ranges. We installed such a setup with four anchors in the room and four tags on a SW (see Fig. 1).

Range measurement between an anchor and a tag is performed using two-way ranging, by measuring the time of flight of the UWB signal. The actual range between the two UWB nodes can be determined, assuming line-of-sight (LOS) propagation. In practice, UWB signals are also reflected or blocked by the environment (walls, furniture, people, wheelchair). This may lead to measuring non-line-of-sight



(a) (b)

Figure 1: Experimental Setup.

(NLOS) propagation distances, which are longer than the actual distance between the two nodes.

Figure 2 shows the range error distribution of UWB data during trials in which a user was driving the wheelchair while performing daily-life tasks (ground truth comes from a motion capture system). One can distinguish two sources of uncertainties. In LOS conditions, uncertainties are coming from UWB sensors noise, easily handled by a bounded-error model. In NLOS conditions, uncertainties are coming mostly from multipath effects, providing range measurements greater than the actual one, leading to outliers.

Wheelchair Pose Computation

We define the wheelchair pose domain computation as a CSP. Since outliers in range measurements are always positive, we can avoid using q-relaxed intersection [2] and GOMNE methods [3], as long as the wheelchair is inside the polygon defined by the anchors.

- The wheelchair pose (x, y, θ) defines the transformation between the world frame, and the body frame of the wheelchair (in which

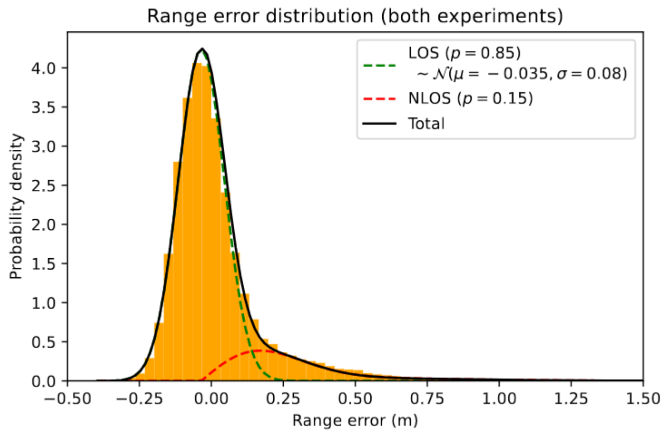


Figure 2: Histogram of the UWB range measurements error.

the tag positions are known).

- The measured UWB ranges d_{meas} provide constraints between the anchor positions $\mathbf{x}_a$ and the tag positions $\mathbf{x}_t$ in the world frame. To take NLOS propagation into account, we use only a single inequality constraint per measurement, assuming NLOS only yields longer measured ranges: $\|\mathbf{x}_t - \mathbf{x}_a\| < d_{\text{meas}} + b_{\text{UWB}}$, with b_{UWB} the UWB ranging error bound in LOS condition.

Solving the CSP with SIVIA and contractors, the set of all feasible wheelchair poses (x, y, θ) compatible with UWB measurements is computed.

Experimental Results

Experimental tests were conducted to record UWB data and validate our localization method. UWB range measurement error bounds were set to 24 cm. The midpoint of computed confidence domains is taken as an estimation of the SW pose. Experimental results are presented in Fig. 3, and demonstrate our method's ability to deal with outliers using only our bounded-error model. The mean horizontal position error over all trials is 9.90 cm.

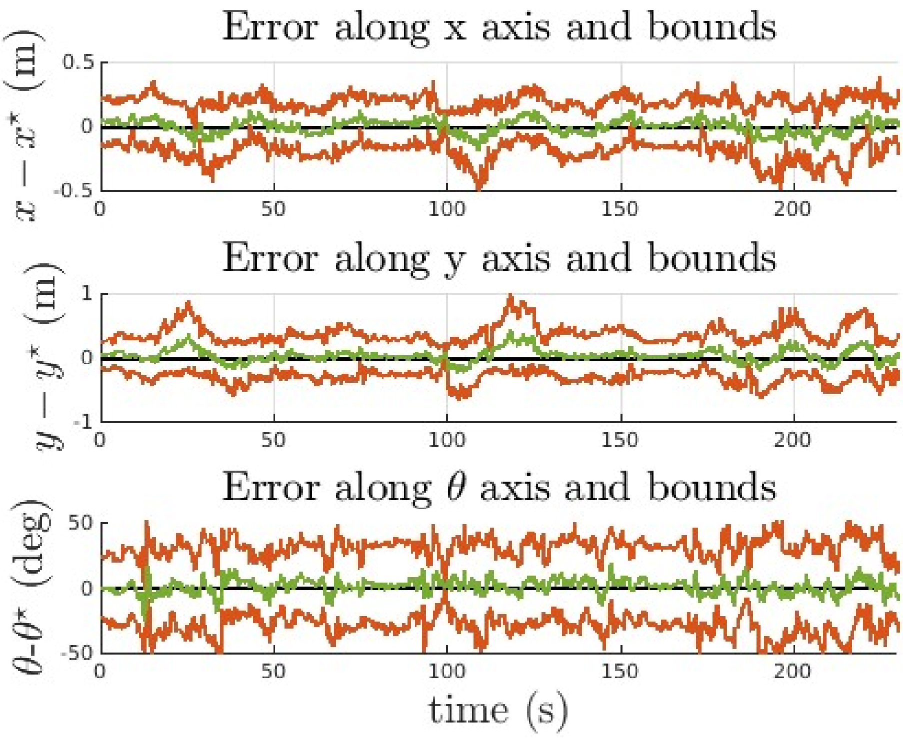


Figure 3: Set-membership position estimation error (green), and confidence lower and upper bounds (orange). The reference is at zero (black)

References

[1] F. Pasteau, M. Babel, and R. Sekkal. *Corridor following wheelchair by visual servoing* International Conference on Intelligent Robots and Systems, 2013

[2] M. Kieffer, L. Jaulin, E. Walter, and D. Meizel. *Robust autonomous robot localization using interval analysis* Reliable Computing, vol.6, n°3, 2000

[3] B. Desrochers, S. Lacroix, and L. Jaulin. *Set-membership approach to the kidnapped robot problem* International Conference on Intelligent Robots and Systems, 2015