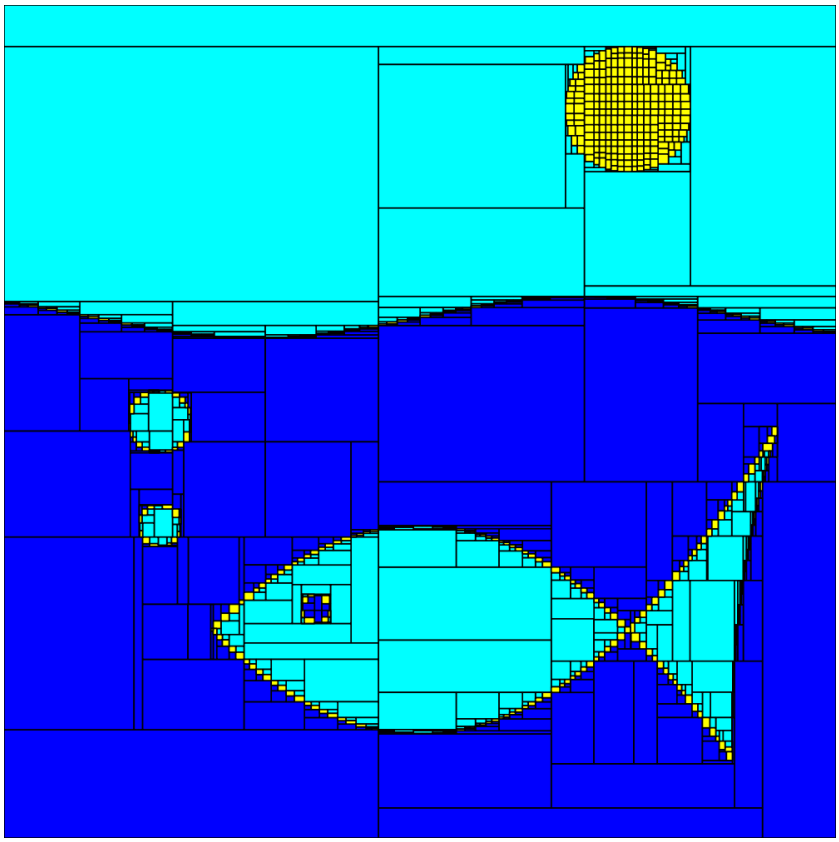
10th Summer Workshop on Interval Methods, and 3rd International Symposium on Set Membership Applications, Reliability and Theory. (SWIM - SMART 2017)

Aerospace Research Institute
United Kingdom Interval Methods Working Group



BOOK OF ABSTRACTS

The University of Manchester

Manchester, UK
14-16, June 2017

10th Summer Workshop on Interval Methods, and 3rd International Symposium on Set Membership Applications, Reliability and Theory.

(SWIM - SMART 2017) – PROGRAMME

(Please be aware that presentation time is 20 minutes and 10 minutes for Q&A.)

DAY 1

Session 1: Mathematics 1

Time	Title	Authors
9:00 am - 9:30 am	Opening Session	
9:30 am - 10:00 am	Rigorous Function Calculi for Hybrid Systems and Beyond	Pieter Collins, Davide Bresolin, Luca Geretti and Tiziano Villa
10:00 am - 10:30 am	Generation of Test matrices with Specified Eigenvalues	Katsuhisa Ozaki and Takeshi Ogita
10:30 am - 11:00 am	Coffee break	

Session 2: Mathematics 2

Time	Title	Authors
11:00 am - 11:30 am	Accurate Numerical Solutions of Linear Systems	Ryota Ochiai, Takeshi Terao and Katsuhisa Ozaki
11:30 am - 12:00 pm	Fast verification methods for proving non-singularity of matrices	Takeshi Terao and Katsuhisa Ozaki

Session 3: Implementation

Time	Title	Authors
12:00 pm - 12:30 pm	Interval-based QuickXplain Algorithm	Adrien Bisutti, Julien Alexandre dit Sandretto, Alexandre Chapoutot, and R é mi Delmas
12:30 pm - 1:00 pm	Extending interval and zonotopic set computation to polytopic set computation for nonlinear discrete-time systems	Jian Wan
1:00 pm - 3:00 pm	Lunch	

Session 4: Control Systems 1

Time	Title	Authors
3:00 pm - 3:30 pm	Interval-Based Techniques for Variable-Structure and Backstepping Control of Nonlinear Multi-Input Multi-Output Systems	Andreas Rauh, Julia Kersten, and Harald Aschemann
3:30 pm - 4:00 pm	Observer-based state feedback for a class of interval Model: Application to multi-Dof micro-positioning system	Mounir Hammouche, Philippe Lutz and Micky Rakotondrabe
4:00 pm - 4:30 pm	Interval Methods for Robust Gain Scheduling Controllers	Julia Kersten, Andreas Rauh and Harald Aschemann
4:30 pm - 5:00 pm	Coffee break	

Session 5: Robotics and Autonomous Systems 1

Time	Title	Authors
5:00 pm - 5:30 pm	BoxRRT* - A Reliable Motion Planner	Adina M. Panchea, Alexandre Chapoutot and David Filliat
5:30 pm - 6:00 pm	Vision based Pose domain characterization of an Unmanned Aerial Vehicle using Interval Analysis	Ide-Flore Kenmogne, Vincent Drevelle and Eric Marchand

END OF DAY 1

DAY 2

Session 6: Neural Networks

Time	Title	Authors
9:00 am - 9:30 am	Interval Methods for Resolving Neural Computation Issues	S. P. Adam, D. A. Karras, M. D. Magoulas and M. N. Vrahatis

Session 7: Control Systems 2

Time	Title	Authors
9:30 am - 10:00 am	Tight interval state estimator based on output set-inversions	Nacim Meslem and Nacim Ramdani
10:00 am - 10:30 am	A bounded-error quaternion-based attitude estimation approach	Nacim Ramdani and Sylvain Miossec
10:30 am - 11:00 am	Coffee break	

Session 8: Robotics and Autonomous Systems 2

Time	Title	Authors
11:00 am - 11:30 am	An Interval Approach to Multiple UAV Collision Avoidance	James A. Douthwaite, Allan De Freitas and Lyudmila S. Mihaylova
11:30 am - 12:00 pm	Robust Motion Planning Based on Sliding Horizon and Validated Simulation	Elliot Brendel, Julien Alexandre dit Sandretto, and Alexandre Chapoutot

Session 9: Stability and Viability Theory

Time	Title	Authors
12:00 pm - 12:30 pm	Contractor Based Viability Algorithms	St é phane Le M é nec
12:30 pm - 1:00 pm	Eulerian state estimation	Thomas Le M é zo, Luc Jaulin and Benoıt Zerr
1:00 pm - 3:00 pm	Lunch	

Session 10: Control Systems 3

Time	Title	Authors
3:00 pm - 3:30 pm	Nonlinear Optimal Control via Occupation Measures and Interval Analysis	Nicolas Delanoue, S é bastien Lagrange and Mehdi Lhommeau
3:30 pm - 4:00 pm	The Box Regularized Particle Filter: A probabilistic set-membership observer	Nicolas Merlinge, Karim Dahia, H é lene Piet-Lahanier, James Brusey, Nadjim Horri
4:00 pm - 4:30 pm	Extended Quantified Set Inversion Algorithm with Applications to Control	Pau Herrero and Miguel A. Sainz
4:30 pm - 5:00 pm	Coffee break	

Session 11: Robotics and Autonomous Systems 3

Time	Title	Authors
5:00 pm - 5:30 pm	Localization for Group of Robots using Matrix Contractors	Nisha Rani Mahato, Luc Jaulin and Snehashish Chakaverty
5:30 pm - 6:00 pm	Bandwidth efficient concurrent ranging and communication for localisation in underwater acoustic networks	Jan Sliwka, Andrea Munafo, Roberto Petroccia

Session 12: Applications

Time	Title	Authors
6:00 pm - 6:30 pm	Comparison between Particle Filter and Interval Analysis for Wind Farm Targets Detection by Multistatic Radar System	Waleed al Mashhadani

7:30 pm	Official dinner

END OF DAY 2

DAY 3

Time	Title	Authors
9:00 am - 10:00 am	Interval vs Set-membership Approaches: Application to State/Parameter Estimation and Fault Detection	Professor Vicenç Puig, Polytechnic University of Catalonia, Barcelona, Spain

Session 13: Image Processing

Time	Title	Authors
10:00 am - 10:30 am	Interval-state cellular automata and their applications to image segmentation	Irina Voiculescu, Imre Boros, Nicolae Popovici, Laura Diosan and Anca Andreica
10:30 am - 11:00 am	Primitive shapes recognition using interval methods	Salvador Pacheco
11:00 am - 11:30 am	Coffee break	

Session 14: Fault Diagnosis and Fault Tolerant Control

Time	Title	Authors
11:30 am - 12:00 pm	Set-membership functional diagnosability through linear functional independence	Carine Jauberthie, Nathalie Verdiere, Louise Trav é -Massuyes
12:00 pm - 12:30 pm	Examples on Verified Diagnosis of Safety Critical Dynamic Systems Based on Kaucher Interval Arithmetik	Stefan Schwab, Oliver Stark and Soeren Hohmann
12:30 pm - 1:00 pm	Fault Tolerant Control using Viability Theory	Vicenç Puig, Majid Ghaniee Zarch and Javad Poshtan

Session 15: Robotics and Autonomous Systems 4

Time	Title	Authors
1:00 pm - 1:30 pm	Improving Guaranteed Coverage Assessment of a Robotic Survey in the Translation Invariant Case	Vincent Drevelle
1:30 pm - 2:00 pm	Guaranteed SLAM - An Interval Approach	Mohamed Mustafa, Eduard Codres, Nicolas Delanoue, and Alexandru Stancu

Session 16: Stability and Viability Theory 2

Time	Title	Authors
2:00 pm - 2:30 pm	A new method for computing an inner and outer approximation for guaranteed integration with uncertain initial conditions	Alexandru Stancu, Eduard Codres, Mario Martinez Guerrero and Vicenç Puig,
2:30 pm - 3:00 pm	Computing an Inner Approximation of the Viability Kernel using capture tubes	Eduard Codres, Joaquim Blesa, Mario Martinez, and Alexandu Stancu
3:00 pm - 5:00 pm	Lunch	

END OF DAY 3

Preface

About SWIM (Summer Workshop on Interval Methods)

The goal of SWIM is to bring together researchers and practitioners working on interval methods and their applications, in the broader sense, providing a forum to review and discuss the state-of-the-art in this area and fostering cross-fertilization between different approaches. The workshop was initiated by the French MEA working group on Set Computation and Interval Techniques of the French research group on Automatic Control GDR MACS. The MEA group aims at promoting interval analysis techniques and applications.

About SMART (International Symposium on Set Membership - Applications, Reliability and Theory)

Set-membership techniques and related interval methods are computational methods that can perform, in a natural way, nonlinear computations with sets of real numbers. They are at the core of guaranteed system solving methods that can prove the existence of a solution and, if the latter is not unique, compute the set of all solutions while taking into account all sources of uncertainty. These methods have direct applicability to a broad range of scientific areas from engineering, to financial and medical domains. The goal of SMART is to bring together researchers working on set-membership techniques and related interval analysis methods, and interested in both fundamental and applied research. The SMART symposium was initiated by the Aerospace Research Institute from The University of Manchester, UK together with French MEA working group on Set Computation and Interval Techniques of the French research group on Automatic Control GDR MACS.

Organising Committee

Dr Alexandru Stancu, The University of Manchester, UK

Prof Luc Jaulin, ENSTA Bretagne, France

Prof Nacim Ramdani, Université d'Orléans, France

Prof Constantinos Soutis, The University of Manchester, UK

Dr Zhengtao Ding, The University of Manchester, UK

Dr Adam Joesbury, The University of Manchester, UK

Dr Mohamed Mustafa, The University of Manchester, UK

Dr Salvador Pacheco-Gutierrez, The University of Manchester, UK

Mrs Alma Rebeca Velasco Olmos, The University of Manchester, UK

Mr Eduard Codres, The University of Manchester, UK

Mr Mario G. Martinez Guerrero, The University of Manchester, UK

Mrs Claire Taylor, The University of Manchester, UK

Contents

Pieter Collins, Davide Bresolin, Luca Geretti and Tiziano Villa:
Rigorous Function Calculi for Hybrid Systems and Beyond 16

Katsuhisa Ozaki and Takeshi Ogita:
Generation of Test matrices with Specified Eigenvalues 20

Ryota Ochiai, Takeshi Terao and Katsuhisa Ozaki:
Accurate Numerical Solutions of Linear Systems 22

Takeshi Terao and Katsuhisa Ozaki:
Fast verification methods for proving non-singularity of matrices . 24

Adrien Bisutti, Julien Alexandre dit Sandretto, Alexandre Chapoutot, and Rmi Delmas:
Interval-based QuickXplain Algorithm 26

Jian Wan:
Extending interval and zonotopic set computation to polytopic set computation for nonlinear discrete-time systems 30

Andreas Rauh, Julia Kersten, and Harald Aschemann:
Interval-Based Techniques for Variable-Structure and Backstepping Control of Nonlinear Multi-Input Multi-Output Systems 32

Mounir Hammouche, Philippe Lutz and Micky Rakotondrabe:
Observer-based state feedback for a class of interval Model: Application to multi-Dof micro-positioning system 35

Julia Kersten, Andreas Rauh and Harald Aschemann:
Interval Methods for Robust Gain Scheduling Controllers 39

Adina M. Panchea, Alexandre Chapoutot and David Filliat:
BoxRRT - A Reliable Motion Planner* 42

Ide-Flore Kenmogne, Vincent Drevelle, and Eric Marchand:
Vision based Pose domain characterization of an Unmanned Aerial Vehicle using Interval Analysis 46

S. P. Adam, D. A. Karras, M. D. Magoulas and M. N. Vrahatis:
Interval Methods for Resolving Neural Computation Issues 51

Nacim Meslem and Nacim Ramdani:
Tight interval state estimator based on output set-inversions 55

Nacim Ramdani and Sylvain Miossec:
A bounded-error quaternion-based attitude estimation approach . . 59

James A. Douthwaite, Allan De Freitas and Lyudmila S. Mihaylova:
An Interval Approach to Multiple UAV Collision Avoidance 63

Elliot Brendel, Julien Alexandre dit Sandretto, and Alexandre Chapoutot:
Robust Motion Planning Based on Sliding Horizon and Validated Simulation 65

Stphane Le Mnec:
Contractor Based Viability Algorithms 69

Thomas Le Mézo, Luc Jaulin and Benoît Zerr:
Eulerian state estimation 73

Nicolas Merlinge, Karim Dahia, Hlne Piet-Lahanier, James Brusey, Nadjim Horri:
The Box Regularized Particle Filter: A probabilistic set-membership observer 86

Pau Herrero and Miguel A. Sainz:
Extended Quantified Set Inversion Algorithm with Applications to Control 90

Nisha Rani Mahato, Luc Jaulin and Snehashish Chakaverty:
Localization for Group of Robots using Matrix Contractors 91

Jan Śliwka, Andrea Munaf, Roberto Petroccia:
Bandwidth efficient concurrent ranging and communication for localisation in underwater acoustic networks 98

Waleed al Mashhadani:
Comparison between Particle Filter and Interval Analysis for Wind Farm Targets Detection by Multistatic Radar System 102

Irina Voiculescu, Imre Boros, Nicolae Popovici, Laura Dioşan and Anca Andreica:
Interval—state cellular automata and their applications to image segmentation 108

Salvador Pacheco-Gutierrez:
Primitive Shape Recognition using Interval Methods 112

Carine Jauberthie, Nathalie Verdière, Louise Travé-Massuyès:
Set-membership functional diagnosability through linear functional independence 119

Stefan Schwab, Oliver Stark and Soeren Hohmann:
Examples on Verified Diagnosis of Safety Critical Dynamic Systems Based on Kaucher Interval Arithmetik 123

Vicenç Puig, Majid Ghaniee Zarch and Javad Poshtan:
Fault Tolerant Control using Viability Theory 127

Vincent Drevelle:
Improving Guaranteed Coverage Assessment of a Robotic Survey in the Translation Invariant Case 134

Mohamed Mustafa , Eduard Codres, Nicolas Delanoue and Alexandru Stancu:
Guaranteed SLAM — An Interval Approach 138

Alexandru Stancu, Eduard Codres, Mario Martinez Guerrero and Vicenç Puig:
A new method for computing an inner and outer approximation for guaranteed integration with uncertain initial conditions. 141

Eduard Codres, Joaquim Blesa, Mario Martinez Guerrero, Alexandru Stancu:
Computing an Inner Approximation of the Viability Kernel Using Capture Tubes 156

Rigorous Function Calculi for Hybrid Systems and Beyond

Pieter Collins[1], Davide Bresolin [2], Luca Geretti[3] and Tiziano Villa[3]

[1] Maastricht University, Department of Data Science and Knowledge Engineering, Bouillonstraat 8-10, Postbus 616, 6200MD, The Netherlands
pieter.collins@maastrichtuniversity.nl
[2] Università di Bologna, Dipartimento di Informatica, Mura Anteo Zamboni 7, 40126 Bologna, Italy
davide.bresolin@unibo.it
[3] Università degli Studi di Verona, Dipartimento di Informatica Strada le Grazie 15, 37134 Verona, Italy
{luca.geretti,tiziano.villa}@univr.it

Keywords: Rigorous function calculi, hybrid systems, Ariadne

Introduction

Almost all problems in applied mathematics deal with spaces of real-valued functions on Euclidean domains in their formulation and solution, representing dependencies on parameters, time- and space- dependent solutions, or distributions of random variables. For applications requiring rigorous analysis and verification of dynamic systems, it is important to be able manipulate functions in a natural, rigorous and efficient way. I will present the rigorous calculus of continuous and differentiable functions as implemented in the package Ariadne for reachability analysis of hybrid systems [1], and suggest ideas for extensions to other classes of function.

Computable operations

From a theoretical perspective, it is important to know what representations and operations are possible. Defining a real number x as

a sequence of nested intervals $[\underline{x}_n, \overline{x}_n]$ with intersection $\{x\}$, a continuous function $f : \mathbb{R} \to \mathbb{R}$ can be defined by its *interval extensions* in such a way that evaluation $(f, x) \mapsto f(x)$ is a computable operation. Composition $f, g \mapsto f \circ g$ is computable, from which we can deduce computability of pointwise arithmetical operations. Integration $(f, a, b) \mapsto \int_a^b f(x)dx$ is computable, but differentiation is uncomputable, so continuously differentiable functions must be specified directly via their derivatives (which can be computed automatically given a symbolic formula). Algebraic equations with isolated nondegenerate roots are solvable, as are differential equations with unique solutions. The maximum over a closed bounded interval $(f, a, b) \mapsto \max_{x\in[a,b]} f(x)$ is computable, as as the supremum norm $||f||_{[a,b]} = \max_{x\in[a,b]} |f(x)|$. Measurable functions can be defined as the effective completion of continuous functions (or piecewise-constant functions) under the Fan metric $d(f, g) = \sup\{\epsilon \in \mathbb{R}^+ \mid \lambda(\{x \mid d(f(x), g(x)) > \epsilon\}) < \epsilon\}$, but evaluation of measurable functions is uncomputable.

Function models

From a practical perspective, it is important to have efficient implementations of these function types. Continuous functions on unbounded domains can be represented by interval extensions; we say a representative is *effective* if it allows arbitrarily accurate evaluation, and *validated* if evaluation is only possible up to a given precision. However, it is more convenient to represent continuous functions by approximations on some bounded domain $\prod_{i=1}^n [a_i, b_i]$ with a given (uniform) error bound. A natural representation is given by scaled Taylor polynomials $f(x) = p(s^{-1}(x)) \pm e$ where $s : [-1, +1]^n \to \prod_{i=1}^n [a_i, b_i]$ is a scaling function, $p(x) = \sum_\alpha c_\alpha x^\alpha$ is a polynomial in the standard (Taylor) basis with coefficients c_α, and e is an error bound [2]. The main computable operations given above have all been implemented with respect to this representation in ARIADNE; for example, differential equations can be solved using the Picard operator. Alternative and equivalent representations include Chebyshev and Bernstein basis polynomials.

Hybrid systems

Hybrid systems are dynamic systems in which the state evolves continuously via a differential equation $\dot{x} = f(x)$ until a *guard* condition $g(x) \geq 0$ is satisfied, at which time the state jumps by the *reset* to $x' = r(x)$. The solution of the differential equation at time t starting at x can be represented by the *flow* $\phi(x,t)$, and assuming crossings of the guard set boundary are transverse, the *crossing time* $\tau(x)$ starting from a point x is given by $g(\phi(x,\tau(x)))$, so the state immediately following a jump is $\psi(x) = r(\phi(x,\tau(x)))$.The evolution can therefore be computed by the operations of (i) solving a differential equation, (ii) solving a parametrised algebraic equation, and (iii) function composition. This approach is implemented in Ariadne, and provides an accurate and efficient way of computing the infinite-time reachable sets. The set reached starting in a set X_0 up to the first event is given by $\{\phi(x,t) \mid x \in X_0 \wedge t \in [0,\infty) \mid t \leq \tau(x)\}$, and the set $\psi(X_0)$ reached immediately after the first event is used as a starting-point for further evolution.

Extended function types

Moving beyond finite-dimensional deterministic systems requires more advanced function types and representations. Switching systems and optimal control problems require piecewise-continuous functions. Galerkin methods for partial differential equations suggest the development of a Fourier function calculus, such as that implemented in [3] for the Kot-Schaffer growth-dispersal model, while finite-element methods require rigorous spline functions. Stochastic processes require measurable functions to represent probability distributions, which could be represented as completions of piecewise-constant functions, or via Gaussian basis functions. For all these applications, the key will be to develop both efficient concrete representations and operations, and clean abstractions to facilitate ease-of-use and interoperability of different implementations.

References

[**1**] Pieter Collins, Davide Bresolin, Luca Geretti, and Tiziano Villa. computing the evolution of hybrid systems using rigorous function calculus. In *Proceedings of the 4th IFAC Conference on Analysis and Design of Hybrid Systems*, 2012.

[**2**] M. Berz and K. Makino. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4(4):361–369, 1998.

[**3**] S. Day, O. Junge, and K. Mischaikow. A rigorous numerical method for the global analysis of infinite-dimensional discrete dynamical systems. *SIAM Journal on Applied Dynamical Systems*, 3(2):117–160, 2004.

Generation of Test matrices with Specified Eigenvalues

Katsuhisa Ozaki[1] and Takeshi Ogita[2]

[1] Shibaura Institute of Technology, Department of Mathematical Sciences,
307 Fukasaku, Minuma-ku, Saitama-shi, Saitama 337-8570, Japan
ozaki@sic.shibaura-it.ac.jp
[2] Tokyo Woman's Christian University, Division of Mathematical Sciences,
2-6-1 Zempukuji, Suginami-ku, Tokyo 167-8585, Japan
ogita@lab.twcu.ac.jp

Keywords: Test matrices, Numerical-linear algebra, Floating-point arithmetic, Eigenvalue problems

Introduction

This discussion, which is concerned with test matrices for eigenvalue problems, will be useful for verifying behavior of numerical algorithms and the accuracy of numerical results. Test matrices are well summarised in [2, Section 28]. There are matrices whose eigenvalues are known exactly, e.g. circulant, Clement and the tridiagonal Toeplitz matrices. We aim to develop a method which produces a matrix with specified eigenvalues.

Problems and Overview of Proposed Method

From a given diagonal matrix $D \in \mathbb{R}^{n\times n}$, we use the following diagonalisation forms:

$$A := XDX^{-1} \quad \text{or} \quad B := QDQ^T \tag{1}$$

where $X \in \mathbb{R}^{n\times n}$ is a non-singular matrix and $Q \in \mathbb{R}^{n\times n}$ is an orthogonal matrix. Then, the exact eigenvalues of A and B are known. However, the following problems arise if floating-point numbers and floating-point arithmetic are used.

Problem 1 Even if $X \in \mathbb{F}^{n\times n}$, X^{-1} may not be represented by floating-point numbers.

Problem 2 Q can be obtained from a non-singular matrix by QR decomposition. However, the exact orthogonal matrix cannot be obtained if rounding errors occur in QR decomposition computations.

Problem 3 Even if X, X^{-1} and Q can be represented by floating-point numbers, XDX^{-1} and QDQ^T may not be representable by floating-point numbers.

Let $\mathbb{F}$ be a set of floating-point numbers as defined by IEEE 754 [1]. We use a scaled Hadamard matrix for Q and an extension of unimodular matrices for X to solve Problems 1 and 2. Next, we compute a diagonal matrix $D' \in \mathbb{F}^{n\times n}$ from D such that there is no rounding error in the evaluation of $XD'X^{-1}$ and $QD'Q^T$. Then, Problem 3 can be solved. We can obtain the matrix D' ($\approx D$) using an error-free transformation of floating-point numbers [3, Algorithm 3.2]. The cost of the proposed method is much less than that of matrix multiplication. The details of the proposed method will be presented at the workshop.

References

[**1**] *IEEE Standard for Floating-Point Arithmetic*, Std 754-2008, 2008.

[**2**] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 2nd ed., 2002.

[**3**] S.M. Rump, T. Ogita, and S. Oishi, Accurate floating-point summation part I: Faithful rounding, *SIAM J. Sci. Comput.*, 31(1):189–224, 2008.

Accurate Numerical Solutions of Linear Systems

Ryota Ochiai[1], Takeshi Terao[1] and Katsuhisa Ozaki[2]

[1] Graduate School of Engineering and Science, Shibaura Institute of Technology
mf16013@shibaura-it.ac.jp

[2] Department of Mathematical Sciences, Shibaura Institute of Technology

Keywords: linear systems, verified numerical computations

Introduction and Notation

Our talk is concerning accurate numerical solutions of linear systems. The aim is to produce numerical results with guaranteed accuracy, e.g. faithful rounding or rounding to the nearest floating-point number. In addition, our method produces the best possible inf-sup interval as the enclosure of the exact solution.

Let $\mathbb{F}$ be the set of floating-point numbers as defined in the IEEE 754 standard [1]. $\mathbb{U}$ denotes the set comprising the subnormal floating-point numbers and zero. Let $\mathbf{u}$ be the relative rounding error unit. The concept of "faithful rounding" was proposed by Rump, Ogita and Oishi [2]. $\hat{a} \in \mathbb{F}$ is a faithful rounding of $a \in \mathbb{R}$ if and only if $\texttt{pred}(\hat{a}) < a < \texttt{succ}(\hat{a})$. This can be extended to vectors in a straightforward manner. The function $\mathrm{ufp}(a)$, for $a \in \mathbb{F}$, returns the unit in the first place of the binary representation of a. We consider linear systems of the form $Ax = b$, where $A \in \mathbb{F}^{n \times n}$ is the coefficient matrix and $b \in \mathbb{F}^n$ is the right-hand-side vector.

Proposed method

We propose a method that produces results of guaranteed accuracy. Each element of a numerical solution is represented by an unevaluated

sum of floating-point numbers such that

$$\hat{x} = \sum_{i=1}^{k} \hat{x}^{(i)},\ \hat{x}^{(i)} \in \mathbb{F}^n,\ k \geq 2. \tag{1}$$

We obtain $\hat{x}$ using iterative refinements. We set $\delta \in \mathbb{F}^n$ as

$$|\sum_{i=1}^{k} \hat{x}^{(i)} - x| \leq \delta. \tag{2}$$

Here we present the following lemma based on [2].

Lemma 1. *For δ in (2) and $\hat{x}$ in (1), if*

$$|\sum_{i=2}^{k} \hat{x}^{(i)}| + |\delta| < \mathbf{u}|\hat{x}^{(1)}| \quad \textit{and} \quad \hat{x}^{(1)} \in \mathbb{F}^n \backslash \mathbb{U}^n$$

are satisfied, then $\hat{x}^{(1)}$ is a faithful rounding of x. If

$$|\sum_{i=2}^{k} \hat{x}^{(i)}| + |\delta| < \frac{1}{2}\mathbf{u} \cdot \mathrm{ufp}(\hat{x}^{(1)}) \quad \textit{and} \quad \hat{x}^{(1)} \in \mathbb{F}^n \backslash \mathbb{U}^n$$

are satisfied, then $\hat{x}^{(1)}$ is the nearest floating-point number to x. Assume that $\hat{x}^{(1)}$ is a faithful rounding of x and that $|\sum_{i=3}^{k} \hat{x}^{(i)}| + |\delta| < |\hat{x}^{(2)}|$, then

$$\hat{x}^{(2)} < 0 \Rightarrow \mathrm{pred}(\hat{x}^{(1)}) < x < \hat{x}^{(1)}, \quad \hat{x}^{(2)} > 0 \Rightarrow \hat{x}^{(1)} < x < \mathrm{succ}(\hat{x}^{(1)}).$$

Adaptive implementation and numerical results will be shown at the presentation.

References

[**1**] *IEEE Standard for Floating-Point Arithmetic*, Std 754-2008, 2008.

[**2**] S. M. Rump, T. Ogita, and S. Oishi, Accurate Floating-Point Summation Part I: Faithful Rounding, *SIAM J. Sci. Comput.* Volume 31, Issue 1, pp. 189-224 (2008).

Fast verification methods for proving non-singularity of matrices

Takeshi Terao[1] and Katsuhisa Ozaki[2]

[1] Shibaura Institute of Technology, Graduate School of Engineering and Science,
307 Fukasaku, Minuma-ku, Saitama-shi, Saitama 337-8570, Japan
nb17105@shibaura-it.ac.jp
[2] Shibaura Institute of Technology, Department of Mathematical Sciences
307 Fukasaku, Minuma-ku, Saitama-shi, Saitama 337-8570, Japan
ozaki@sic.shibaura-it.ac.jp

Keywords: computer-assisted proof, verified numerical computations, interval arithmetic

Introduction and Preceding Study

This study aims to verify non-singularity of matrices using numerical computations. For $A \in \mathbb{R}^{n\times n}$, if there exists $R \in \mathbb{R}^{n\times n}$ such that $\|RA - I\| < 1$, A is non-singular, where I is the identity matrix. Based on this theorem, a fast verification method has been proposed by Oishi and Rump [1].

Here, we briefly review the Oishi-Rump method. Let $\hat{L}$ and $\hat{U}$ be the computed LU factors of PA, where P is a permutation matrix, i.e., $PA \approx \hat{L}\hat{U}$. Matrices X_L and X_U are the approximate inverse matrices of $\hat{L}$ and $\hat{U}$, respectively. Let $e = (1, \ldots, 1)^T \in \mathbb{R}^n$. Oishi and Rump set $R := X_U X_L P$ and derived the following inequality:

$$\|RA - I\|_\infty \leq \|2nu|X_U||X_L||L||U|e + nu|X_U||U|e\|_\infty, \qquad (1)$$

where u is the unit roundoff, e.g., $u = 2^{-53}$ for the binary64 format in the IEEE 754 standard. Assume that no underflow occurs in numerical computations. The upper bound of Eq. (1) can be calculated without matrix multiplication. The total cost is $\frac{4}{3}n^3 + \mathcal{O}(n^2)$ flops.

The Proposed Method

Let $R = (\hat{L}\hat{U})^{-1}P$ and $\Delta A = PA - \hat{L}\hat{U}$. Then, we obtain

$$\|RA - I\|_\infty = \|RP^T(\hat{L}\hat{U} + \Delta A) - I\|_\infty \leq \|R\|_\infty \|\Delta A\|_\infty. \quad (2)$$

Let $\Delta L = I - \hat{L}X_L$ and $\Delta U = I - X_U\hat{U}$. If $\|\Delta L\| < 1$ and $\|\Delta U\| < 1$, then $\|R\|$ is bounded by

$$\|R\| = \|(\hat{L}\hat{U})^{-1}\| \leq \frac{\|X_U X_L\|}{(1 - \|\Delta U\|)(1 - \|\Delta L\|)}. \quad (3)$$

By substituting Eq. (3) into Eq. (2), we obtain

$$\|RA - I\|_\infty \leq \frac{\|X_U X_L\|_\infty \|\Delta A\|_\infty}{(1 - \|\Delta U\|_\infty)(1 - \|\Delta L\|_\infty)}. \quad (4)$$

From [2], the upper bounds of $\|\Delta A\|, \|\Delta L\|$ and $\|\Delta U\|$ are given by

$$\|\Delta A\| \leq nu|||\hat{L}||\hat{U}|||, \ \|\Delta L\| \leq nu|||\hat{L}||X_L|||, \ \|\Delta U\| \leq nu|||X_U||\hat{U}|||.$$

Then, we can compute the upper bound of (4) as follows

$$\|RA - I\|_\infty \leq \frac{nu|||X_U||X_L|e\|_\infty|||\hat{L}||\hat{U}|e\|_\infty}{(1 - nu|||X_U||\hat{U}|e\|_\infty)(1 - nu|||\hat{L}||X_L|e\|_\infty)}. \quad (5)$$

The computational cost of Eq. (5) is comparable to that of the Oishi-Rump method. In addition, we propose methods with $\frac{11}{6}n^3$ flops, $\frac{7}{3}n^3$ flops and $\frac{17}{6}n^3$ flops based on (4) using interval enclosure approaches.

References

[1] S.Oishi and S.M. Rump, *Fast verification of solutions of matrix equations*, Numer. Math. , 90(4), 755–773, 2002.

[2] S.M. Rump and C.-P. Jeannerod, *Improved backward error bounds for LU and Cholesky factorizations*, SIAM. J. Matrix Anal. & Appl. (SIMAX), 35(2), 684–698, 2014.

Interval-based QuickXplain Algorithm

Adrien Bisutti[1], Julien Alexandre dit Sandretto[1], Alexandre Chapoutot[1], and Rémi Delmas[2]

[1]ENSTA ParisTech, Université Paris-Saclay,
828, bd des maréchaux, 91762 Palaiseau Cedex, France
{bisutti,alexandre,chapoutot}@ensta.fr

[2]ONERA, DTIM
2, av. Edouard Belin BP74025, 31055 Toulouse Cedex 4, France
remi.delmas@onera.fr

Keywords: SMT, Conflict Clause, Interval Analysis

Introduction

Cyber-physical systems are made of discrete-time components, *i.e.*, piece of software, and continuous-time components, *i.e.*, a plant which continuously and strongly interact each other. Such kind of systems is usually found in critical application, *e.g.*, aircraft autopilot or cruise control mechanism in a car. In consequence, it is important to ensure safety of such systems in order to avoid the lost of human life.

Formal safety verification techniques aim at automatically and mathematically prove that a mathematical model of a cyber-physical system is safe. One difficulty of this approach is to deal with models involving a combination of state transition systems, representing the software part, and ordinary differential equations, representing the plant part. Model-checking techniques based on *SAT Modulo Theory* (SMT) techniques are efficient and robust enough to deal with such heterogeneous mathematical models. More precisely, *SAT modulo ODE* techniques [3, 4] are very promising to prove safety properties of cyber-physical systems. A SMT solver aims at proving that a first-order logical formula ϕ involving terms coming from different theories, *e.g.*, linear integer arithmetic (LIA) or non-linear real arithmetic (NRA), is *satisfiable*, *i.e.*, there is a value of the variables which

make the formula ϕ true or *unsatisfiable*, *i.e.*, for all values of the variables, ϕ is false. The main algorithm used in SMT solver is known as Conflit-Driven Clause Learning (CDCL).

This article is interested in SMT with NRA theory. This theory is associated to a solver based on *Interval Constraint Propagation* (ICP) techniques [1] which is easily implementable with IBEX library. The contribution of the paper is the definition of an algorithm dedicated to the conflict analysis step. This algorithm is an adaptation of the QuickXplain algorithm [2], mainly dedicated to discrete domain *Constraint Satisfaction Problems* (CSP), to continuous or numerical CSP.

Main contribution

Basically when only one theory T is involved, a SMT solver is made of a SAT solver and a T-solver, such as IBEX for NRA theory. The combination of the two solvers works as follows, starting from a logical formula ϕ involving T terms in *normal conjunctive form* (CNF), $\phi_{\text{CNF}} \equiv \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m} \ell_{ij}$,

1. For each literal ℓ_{ij}, *e.g.*, $\cos(x) + y \leqslant 1$, a Boolean variable $b(\ell_{ij})$ is assigned.

2. SAT solver searches for an assignment α of $b(\ell_{ij})$ such that ϕ_{CNF} is true.

3. T-solver is started if an assignment α exists and it determines if the conjunction of constraints induced by α holds true in the theory T. If the constraints are true then SMT solvers returns SAT, otherwise a *conflict clause* κ is generated and added to ϕ_{CNF} to avoid unfeasible paths during the search of a new α in Step 2.

4. If no assignment α can be found then UNSAT is returned.

In iSat [3], which tightly integrates SAT and ICP solvers, a conflict learning algorithm, to generate κ, is used based on the decision tree at the SAT-solver level. Otherwise, it seems that no conflict analysis

is defined for T-solver keeping the relative independence between the two solvers. To cope with this lack, a new conflict analysis method is proposed for NRA theory and based on the QuickXplain algorithm [2]. An implementation of this algorithm has been performed in IBEX.

Algorithm 1 Continuous QuickXplain algorithm

1: **function** QUICKXPLAIN(C, U, d)
2: **if** $\text{card}(C) \neq 0 \wedge \text{contract}(C, d) = \emptyset$ **then**
3: **return** $\emptyset$
4: **end if**
5: **if** $U = \emptyset$ **then**
6: **return** $\emptyset$
7: **end if**
8: $\alpha_0, \ldots, \alpha_{n-1}$ be an enumeration of U
9: $k \leftarrow 0; \quad C_s \leftarrow C; \quad d_s \leftarrow d$
10: **while** $d_s \neq \emptyset \wedge k < \text{card}(U)$ **do**
11: $C_s \leftarrow C_s \cup \{\alpha_k\}; \quad d_s \leftarrow \text{contract}(C_s, d); \quad k \leftarrow k + 1$
12: **end while**
13: **if** $d_s \neq \emptyset$ **then**
14: **return** $\emptyset$
15: **end if**
16: $k \leftarrow k - 1; \quad X \leftarrow \{\alpha_k\}; \quad i \leftarrow \lfloor k/2 \rfloor$
17: $U_1 \leftarrow \{\alpha_0, \ldots, \alpha_{i-1}\}$
18: $U_2 \leftarrow \{\alpha_i, \ldots, \alpha_{k-1}\}$
19: **if** $U_2 \neq \emptyset$ **then**
20: $C_2 \leftarrow C \cup U_1 \cup X$
21: $X_2 \leftarrow \text{QUICKXPLAIN}(C_2, U_2, d)$
22: $X \leftarrow X \cup X_2$
23: **end if**
24: **if** $U_1 \neq \emptyset$ **then**
25: $C_1 \leftarrow C \cup X$
26: $X_1 \leftarrow \text{QUICKXPLAIN}(C_1, U_1, d)$
27: $X \leftarrow X \cup X_1$
28: **end if**
29: **return** X
30: **end function**

Results

The new conflict analysis method is given in Algorithm 1 where C stands for the smallest set of conflicted constraints (initially $\emptyset$), U stands for the initial set of constraints, d stands for the domain of variables. The contract operations in Line 2 and 14 is implemented using the HC4 algorithm. A positive side effect of this adaptation to continuous constraints of the original work [2] is that some propagation operations have been removed.

Acknowledgement

This work benefited of the "Chair Complex Systems Engineering - École polytechnique, THALES, DGA, FX, DASSAULT AVIATION, DCNS Research, ENSTA ParisTech, Télécom ParisTech, Fondation ParisTech and FDO ENSTA" and also partially funded by DGA MRIS.

References

[1] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer, 2001.

[2] U. Junker and F. Valbonne. QuickXPlain: Conflict Detection for Arbitrary Constraint Propagation Algorithms. In *Proc. of the IJCAI Workshop on Modelling and Solving problems with constraints*, 2001.

[3] A. Eggers, M. Fränzle, and C. Herde. SAT Modulo ODE: A direct sat approach to hybrid systems. In *Proc. of International Symposium on Automated Technology for Verification and Analysis*, Springer, 2008.

[4] S. Gao, S. Kong, and E. M. Clarke. dReal: An SMT Solver for Nonlinear Theories over the Reals, In *Proc. of International Conference on Automated Deduction*, Springer, 2013.

Extending interval and zonotopic set computation to polytopic set computation for nonlinear discrete-time systems

Jian Wan [1]

[1] School of Engineering, The University of Plymouth, United Kingdom
jian.wan@plymouth.ac.uk

Keywords: interval analysis, zonotope, polytope, set computation

Introduction

Taking a set as the input for a function, set computation returns another set as the output of the function. Enclosing real numbers in intervals and real vectors in boxes, interval analysis has become a powerful set computation tool for nonlinear systems [1]. Interval set computation for nonlinear systems can take in an interval or a box as its input and return an interval or a box as its output. For those inputs that are not in the shape of an interval or a box, interval set computation can also be applied through approximating those inputs by a union of intervals or boxes obtained from bisection and selection.

Similar to interval set computation, the dynamic evolution of a nonlinear system with a zonotopic set as the initial state can also be computed directly via zonotopic set computation and the wrapping effect can be reduced greatly with comparison to interval set computation [2]. Except for the reduced wrapping effect, zonotopes as a special kind of polytopes are also more flexible in shape than intervals. Therefore, zonotopic set computation has been increasingly used to solve control and estimation issues for nonlinear systems.

Compared to intervals and zonotopes, polytopes are more flexible in shape. The dynamic evolution of a nonlinear discrete-time system with a polytopic set as the initial state cannot be computed directly due to its mathematical format involving inequality constraints. However,

exact polytopic set computation for nonlinear discrete-time systems can be realized through computing the dynamic evolution of those individual zonotopes whose intersection forms the polytope.

Polytopic set computation

The key to implement exact polytopic set computation for nonlinear discrete-time systems is to represent the polytope $\mathcal{P}$ exactly by the intersection of zonotopes $\mathcal{P} = \mathcal{Z}_1 \cap \cdots \cap \mathcal{Z}_n$. Once the initial polytopic set is represented exactly by the intersection of zonotopes, then $\mathbf{f}(\mathcal{P}) = \mathbf{f}(\mathcal{Z}_1 \cap \cdots \cap \mathcal{Z}_n) \subseteq \mathbf{f}(\mathcal{Z}_1) \cap \cdots \cap \mathbf{f}(\mathcal{Z}_n)$ according to set theory.

Assume that the polytope $\mathcal{P} \subset \Re^2$ has n_c inequality constraints, then the convex polygon $\mathcal{P}$ can be represented exactly by the intersection of $\frac{n_c}{2}$ zonotopes if n_c is even or exactly by the intersection of $\frac{n_c+1}{2}$ zonotopes if n_c is odd. The construction of a parallelogram to contain the 2-D polytope can be transformed to be a linear programming (LP) problem that minimizes the sum of the base length and the side length for the parallelogram to be minimal in volume.

Conclusion

This paper aims to extend interval and zonotopic set computation to polytopic set computation for nonlinear discrete-time systems so that the dynamic evolution of a nonlinear discrete-time system with a polytopic set as the initial state can also be computed exactly.

References

[**1**] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*, Springer, London, 2001.

[**2**] W. Kühn, Rigorously computed orbits of dynamical systems without the wrapping effect, *Computing*, 61(1): 47-67, 1998.

Interval-Based Techniques for Variable-Structure and Backstepping Control of Nonlinear Multi-Input Multi-Output Systems

Andreas Rauh, Julia Kersten, and Harald Aschemann

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

Keywords: Interval analysis, Lyapunov functions, Nonlinear state feedback control, Sliding mode control, Backstepping

Introduction

If nonlinear dynamic systems with uncertain but bounded parameters are concerned, it is desired to design feedback control strategies that are capable of robustly stabilizing the system dynamics with a minimum amount of conservatism. Conservatism is usually caused by overapproximating the influence of uncertain parameters during offline design stages in terms of worst-case (norm) bounds.

Classical Variable-Structure Control Techniques

Using such bounds, classical sliding mode techniques [7,8] can be designed which lead to a guaranteed asymptotically stable behavior of the closed-loop control system. However, the offline overapproximation of the influence of uncertain quantities typically leads to unnecessarily large control amplitudes. Such large control amplitudes are strongly related to the effect of chattering, namely, switchings of the input signal with high frequency around some effective average value. In such a way, chattering may cause undesired actuator wear and unnecessarily large energy consumption.

Interval-Based Control Procedures

To avoid chattering, interval-based variable-structure control procedures were developed in previous work for a guaranteed stabilizing online adaptation of the control strategy [3,5,6]. These approaches have already been applied successfully to various real-life systems and have been extended by approaches that allow for handling both one-sided and two-sided hard state constraints [1,2]. Besides the interval-based extension of variable-structure control laws, an interval-based generalization of backstepping controllers [4] was published in [1]. However, these interval-based variable-structure and backstepping controllers were so far only developed for single-input single-output systems (SISO systems). This contribution aims at a generalization of both types of control strategies to the more general case of multi-input multi-output control tasks which cannot be decoupled perfectly — due to uncertain parameters — into independent SISO systems by a nonlinear coordinate transformation.

References

[**1**] A. Rauh, J. Kersten, and H. Aschemann, Interval-Based Implementation of Robust Variable-Structure and Backstepping Controllers of Single-Input Single-Output Systems, In *Proc. of 20th World Congress of the International Federation of Automatic Control*, Toulouse, France, 2017.

[**2**] A. Rauh, L. Senkel, and H. Aschemann, Interval-Based Sliding Mode Control Design for Solid Oxide Fuel Cells with State and Actuator Constraints, *IEEE Transactions on Industrial Electronics*, 62(8), 5208–5217, 2015.

[**3**] A. Rauh, L. Senkel, and H. Aschemann, Reliable Sliding Mode Approaches for the Temperature Control of Solid Oxide Fuel Cells with Input and Input Rate Constraints, In *Proc. of 1st IFAC Conference on Modelling, Identification and Control of Nonlinear Systems, MICNON 2015*, St. Petersburg, Russia, 2015.

[**4**] S. Sastry, *Nonlinear Systems: Analysis, Stability, and Control*, Interdisciplinary Applied Mathematics, Springer-Verlag, New York, 1999.

[**5**] L. Senkel, A. Rauh, and H. Aschemann, Sliding Mode Techniques for Robust Trajectory Tracking as well as State and Parameter Estimation, *Mathematics in Computer Science*, 8(3–4), 543–561, 2014.

[**6**] L. Senkel, A. Rauh, and H. Aschemann. Experimental and Numerical Validation of a Reliable Sliding Mode Control Strategy Considering Uncertainty with Interval Arithmetic, In A. Rauh and L. Senkel (eds.), *Variable-Structure Approaches for Analysis, Simulation, Robust Control and Estimation of Uncertain Dynamic Systems*, Mathematical Engineering. Springer-Verlag, pp. 87-122, 2016.

[**7**] Y. Shtessel, Ch. Edwards, L. Fridman, and A. Levant, *Sliding Mode Control and Observation*, Springer-Verlag, New York, 2014.

[**8**] V.I. Utkin, *Sliding Modes in Control and Optimization*, Springer-Verlag, Berlin, Heidelberg, 1992.

Observer-based state feedback for a class of interval Model: Application to multi-Dof micro-positioning system

Mounir Hammouche, Philippe Lutz and Micky Rakotondrabe

FEMTO-ST Institute, UMR CNRS 6174-UBFC / ENSMM / UTBM Automatic Control and Micro-Mechatronic Systems depart, AS2M 24 rue Alain Savary, France,
{mounir.hammouche, plutz ,mrakoton}@femto-st.fr

Keywords: Robust Observer-Based State Feedback, Interval models, SIVIA (Set Inversion via Interval Analysis), Multi-DOF micro-positioning system.

Introduction

During the last decades, micro-positioning systems have gained much attention in microrobotic applications such as micro/nano-assembly, micromanipulation, nanotechnology,... [1,2]. However, the control of micro-positioning systems always presents a very difficult task due to the high sensitivity to the environment at this scale, the characteristics of the used smart actuators (hysteresis, creep, ...), and the cross-couplings effects present between the different axes for multivariable case. In fact, different controllers synthesis approaches have been developed to control such system including real-time adaptive, robust control ($H_\infty, \mu - synthesis, ..$), nonlinear approaches and robust interval-based techniques [1-4]. In this presentation we suggest to model the system uncertainties by a linear and time-invariant interval state-space model which is well adapted to multivariable control systems. Furthermore, we propose a robust Observer-Based State Feedback design using interval techniques to control the micro-positioning systems [4-6]. The proposed control strategy is tested in simulation and validated experimentally using a multi-Dof (degrees of freedom) micro-positioning system.

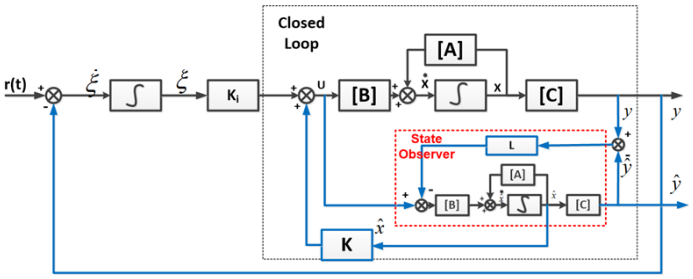


Figure 1: Observer-based state feedback schema.

Problem formulation

In our work the problem of observer-based state feedback controller with integral compensator is addressed for an interval state-space model with realization $([A], [B], [C])$, as shown in fig.1. The objective of the proposed control strategy is to find the set of robust gains matrices $[[K]\ [K_i]]$ and $[L]$, for the controller and the observer respectively, such that the closed-loop state matrices of the controller and of the observer possess their eigenvalues within two desired subregions: one for the controller $\Omega_{Desired\,region\,controller}$ and the other for the observer $\Omega_{Desired\,region\,Observer}$, as depicted in fig.2. These two subregions are defined such that the closed-loop of the observer is (i.e. the state estimation), at least, four times faster then the closed-loop of the controller, and such that they provide a guaranteed stability margin and some predefined performances.

Main results

In order to obtain the set of robust gains for the controller and the observer using pole assignment techniques and interval analysis, foremost, we propose to adopt the separation theorem to be able to find the gains of the controller and the observer in separated way and also to reduce the computation complexity [7]. Furthermore we convert the problem of pole assignment to set inversion problem and solve it using the Set Inversion Via Interval Analysis (SIVIA) algorithm [5] with the help of interval eigenvalues computation [8-10]. This recursive

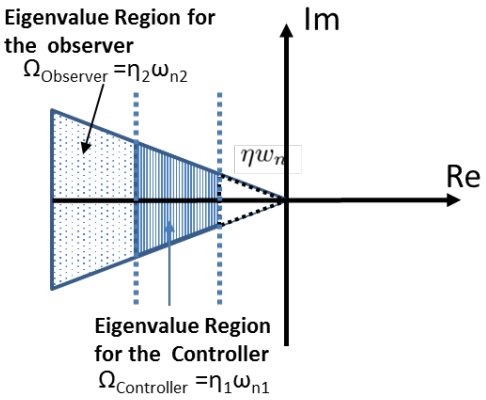


Figure 2: Desired regions for the controller and the observer.

SIVIA-based algorithm approximates with subpaving the set solutions $[[K]\ [K_i]]$ and $[L]$ that satisfy the following inclusions:

$$eig\left[A_{Controller-cl}([A],[B],[C],[[K]\ [K_i]])\right] \subseteq \Omega_{Desired\,region\,controller} \tag{1}$$

$$eig\left[A_{Observer-cl}([A],[B],[C],[L])\right] \subseteq \Omega_{Desired\,region\,observer} \tag{2}$$

where $A_{controller-cl}$ is the augmented closed-loop matrix for the controller and $A_{Observer-cl}$ is the closed-loop matrix for the observer. These two closed-loop matrices are obtained from the separation theorem.

Finally, the effectiveness of the proposed algorithm is tested in simulation by mean of Monte-Carlo simulation and is illustrated by a real experimentation to control a new multi-Dof micro-positioning system.

References

[1] M. Rakotondrabe, Smart materials-based actuators at the micro/nano-scale, , 2013.

[2] S. Devasia, E. Eleftheriou, and S. O. R. Moheimani, A survey of control issues in nanopositioning, *TCST*, 2007.

[3] S. Khadraoui, M. Rakotondrabe, P. Lutz, Interval force/position

modeling and control of a microgripper composed of two collaborative piezoelectric actuators and its automation, *IJCAS*, 2014.

[**4**] Y. Smagina and I. Brewer, Using interval arithmetic for robust state feedback design, *Systems and Control Letters*, 2002.

[**5**] L. Jaulin, Applied interval analysis: with examples in parameter and state estimation, *Springer Science and Business Media*, 2001.

[**6**] D. Efimov, T. Raissi., Design of interval observers for uncertain dynamical systems, *ARC* 2016.

[**7**] K. Zhou, J. C. Doyle, K. Glover, Robust and optimal control, *Book vol.40* 1996.

[**8**] J. Rohn, *A handbook of results on interval linear problems*, 2005.

[**9**] M. Hladik, Bounds on eigenvalues of real and complex interval matrices ,*AMC*, 2013.

[**10**] M. T. Hussein, Assessing 3-d uncertain system stability by using matlab convex hull functions, *IJACSA* 2011.

Interval Methods for Robust Gain Scheduling Controllers

Julia Kersten, Andreas Rauh and Harald Aschemann [1]

[1] University of Rostock, Chair of Mechatronics,
Justus-von-Liebig Weg 6, 18059 Rostock, Germany
{julia.kersten, andreas.rauh, harald.aschemann}@uni-rostock.de

Keywords: off-line gain scheduling, robust and optimal control, LMIs

Introduction

The presented approach allows for combining the fields of a verified simulation of ordinary differential equations [1] and the robustly stabilizing control design using linear matrix inequality (LMI) techniques [2]. It aims, firstly, at a guaranteed stabilization of the system dynamics and, secondly, at a minimum amount of conservatism in the controller gains by their adaptation in terms of guaranteed enclosures for those state variables that influence entries in the system and input matrices of quasi-linear state-space representations.

Gain Scheduling with Robust LMI-Based Control

The main idea is to compute controller gains off-line for a predefined prediction horizon. As mentioned earlier, LMIs are used to ensure robust stability [3]. Although LMI techniques were originally developed for purely linear system models, they can be extended towards nonlinear systems if the state equations can be reformulated into a quasi-linear representation. The controller gain matrix ensures asymptotically stable convergence to the desired operating point for all following time steps if the system model is time-invariant. Due to the fact that asymptotic stability of the closed-loop system dynamics is desired by the choice of a sequence of gain matrices, the final goal is to obtain

state enclosures for which the prediction result of the interval vector is a true subset of the actual interval vector in all components. If the predicted interval is not a subset of the actual interval, it cannot be proven that the interval width reduces in each of the vector components. In this case, the controller gain needs to be adapted in such a way that desired stability, robustness, and optimality criteria are guaranteed to be satisfied for the complete prediction horizon. Finally, for sufficiently large times, this leads to a contraction of the interval widths between two subsequent temporal discretisation steps. To prevent overestimation and an excessive blow-up of the interval width due to the wrapping effect [4], techniques for the reduction of overestimation are applied in the interval-based state prediction. In this, intervals are subdivided and re-merged to limit the conservatism due to too big bounding boxes. This is done once by a reduction of the number of subintervals by a convex interval hull with bounded overestimation and by a re-approximation using disjoint subintervals [5],[6]. The method is implemented in Matlab and verified by simulating the application scenario of an inversed pendulum together with possibilities to construct computationally inexpensive exponential state enclosures and suitable bonding systems.

Results

Starting from initial state intervals, guaranteed bounds of all reachable states were computed along with corresponding robustly stabilizing controller gains. The calculation of the controller gains is done by LMI techniques using a quasi-linear state-space representation of the non-linear continuous-time system model of the pendulum. Thus, robust stability and the optimization of performance criteria such as the H_2 and H_∞ norms can be considered directly in addition to the limitation of desired eigenvalue domains. While the gain scheduling itself guarantees asymptotic stability, the techniques for the reduction of overestimation in the interval-based state prediction are essential to avoid a blow-up of the widths of the computed state enclosures due to the dependency and wrapping effect. In such a way, the techniques for

reduction of overestimation help to limit the computational effort and simultaneously reduce conservatism in the controller gains [7].

References

[**1**] Y. Lin and M. A. Stadtherr, Validated solution of initial value problems for ODEs with interval parameters, *NSF Workshop Proceeding on Reliable Engineering Computing*, Savannah GA, February 22-24 2006.

[**2**] C. Scherer and S. Weiland, Linear Matrix Inequalities in Control, *Control System Advanced Methods*, 2nd ed., ser. The Electrical Engineering Handbook Series, W. S. Levine, Ed. Boca Raton: CRC Press, 2011, pp. 24–1—24–30.

[**3**] J. Ackermann *Robust Control - The Parameter Space Approach* Springer-Verlag, 2002.

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

[**5**] A. Rauh, M. Kletting, H. Aschemann, and E. P. Hofer, Reduction of Overestimation in Interval Arithmetic Simulation of Biological Wastewater Treatment Processes, *Journal of Computational and Applied Mathematics*, vol. 199, no. 2, pp. 207–212, 2007.

[**6**] I. Krasnochtanova, A. Rauh, M. Kletting, H. Aschemann, E. P. Hofer, and K.-M. Schoop, Interval Methods as a Simulation Tool for the Dynamics of Biological Wastewater Treatment Processes with Parameter Uncertainties, *Applied Mathematical Modeling*, vol. 34, no. 3, pp. 744–762, 2010.

[**7**] J. Kersten, A. Rauh, and H. Aschemann, Interval Methods for the Implementation and Verification of Robust Gain Scheduling Controllers, *22nd International Conference on Methods and Models in Automation and Robotics*, under review, 2017.

BoxRRT* - A Reliable Motion Planner

Adina M. Panchea[1], Alexandre Chapoutot[2] and David Filliat[2]

[1] LIX, Ecole Polytechnique, CNRS, Univ. Paris-Saclay, 91128, Palaiseau, France
panchea@lix.polytechnique.fr
[2] U2IS, ENSTA ParisTech, Univ. Paris-Saclay, 91120, Palaiseau, France
{chapoutot, filliat}@ensta.fr

Keywords: Planning and control, Robotics, DynIBEX.

Introduction

This work aims at providing a new reliable (asymptotic) optimal motion planner, denoted BoxRRT* which can guarantee a safe path to an unknown initial mobile robot localisation.
Our proposed motion planner is build upon: **i)** methods which use stochastic sampling to discretise the configuration space, *e.g.*, Rapidly-exploring Random Tree, which can guarantee (asymptotic) optimality of the solution, *e.g.*, optimal RRT (RRT*, see for example [2]) and **ii)** modern and new tools [1] for the guaranteed numerical integration.

Problem formulation

Consider the differential system which describes the evolution of a mobile robot system:

$$\dot{\mathbf{s}}(t) = \mathbf{f}(\mathbf{s}(t), \mathbf{u}(t)) \tag{1}$$

with $\mathbf{s} \in \mathbb{S} \subset \mathbb{R}^n$ the measurable state of the system and $\mathbf{u}(t) \in \mathbb{U}$ the admissible control input.
The purpose of the *robust motion planner* is to provide a sequence of control inputs $\mathbf{u} \in \mathbb{U}_{[\mathbf{u}]}^{\Delta t}$ bounded over intervals of time of the form $[K\Delta t, (K+1)\Delta t[$, with $\Delta t > 0$ and $K \in \mathbb{N}$, which will drive the system to reach $\mathbb{S}_{\text{goal}}$ from initial states $\mathbf{s} \in \mathbb{S}_{\text{init}}$ while avoiding the

non-admissible states $\mathbb{S}_{\text{obs}}$.
Starting from the formulation given in [3] of such a robust motion planner, there exists a sequence of control input $\mathbf{u} \in \mathbb{U}_{[\mathbf{u}]}^{\Delta t}$ to drive the system from an uncertain initial state to a set of goal states $\mathbb{S}_{\text{goal}}$ is as follows

$$\begin{aligned} &\exists K > 0 \text{ and } \mathbf{u} \in \mathbb{U} \text{ such that} \\ &\forall \mathbf{s}_0 \in \mathbb{S}_{\text{init}},\ \forall\, \mathbf{s}(K\Delta t; \mathbf{s}_0) \in \mathbb{S}_{\text{goal}} \text{ and} \\ &\forall t \in [0, K\Delta t],\ \mathbf{s}(t; \mathbf{s}_0) \in \mathbb{S}_{\text{free}}, \end{aligned} \tag{2}$$

with $\mathbf{s}(t; \mathbf{s}_0)$ the exact solution of (1) from the initial condition $\mathbf{s}_0$.

Main results

Let G be the exploration tree, $[\mathbf{s}_{\text{init}}] = \text{Hull}(\mathbb{S}_{\text{init}})$, $[\mathbf{s}_{\text{obs}}] = \text{Hull}(\mathbb{S}_{\text{obs}})$ and $[\mathbf{s}_{\text{goal}}] = \text{Int}(\mathbb{S}_{\text{goal}})$, with $\text{Hull}(\mathbb{S}_{\text{init}})$ the smallest box which contains $\mathbb{S}_{\text{init}}$ (*e.g.*, interval hull) and $\text{Int}(\mathbb{S})$ a box included in $\mathbb{S}$ (*e.g.*, inner approximation). The minimal cost from $[\mathbf{s}_1]$ to $[\mathbf{s}_2]$ according to the Hausdorff distance of two intervals (d), is denoted by $\text{cost}([\mathbf{s}_1], [\mathbf{s}_2])$. Let $\text{cost}([\mathbf{s}_1])$ be the total cost to arrive at $\mathbf{s}_1$, that is $\text{cost}([\mathbf{s}_1]) = \text{cost}([\mathbf{s}_{\text{init}}], [\mathbf{s}_1])$.

BoxRRT* motion planner brief description:

1. First, G is initialized with the given initial configuration $[\mathbf{s}_{\text{init}}]$.
2. Then, a state $[\mathbf{s}_{\text{rand}}] \in \mathbb{S}_{\text{free}}$ is generated randomly.
3. The tree G is searched for the nearest vertex to $[\mathbf{s}_{\text{rand}}]$ according to a user-defined metric d and the $[\mathbf{s}_{\text{nearest}}]$ vertex is provided.
4. A control input $\mathbf{u}$ is selected according to a desired behaviour. Then, (1) is integrated over a fixed time interval Δt with the initial condition $[\mathbf{s}_{\text{nearest}}]$, to find a new state $[\mathbf{s}_{\text{new}}]$. If the new state and the path between it and $[\mathbf{s}_{\text{nearest}}]$ lie in $\mathbb{S}_{\text{free}}$ (*e.g.*, is a collision free path), then $[\mathbf{s}_{\text{new}}]$ is added.

5. Next, the planner tries to find a better parent and children for $[\mathbf{s}_{\text{new}}]$, which needs to provide collision-free-path and a lower cost to and from $[\mathbf{s}_{\text{new}}]$, respectively. For the better parent one searches a set of k-nearest other potential parents to arrive at $[\mathbf{s}_{\text{new}}]$, while for the better children one searches a set of k-nearest other potential children from $[\mathbf{s}_{\text{new}}]$ to other vertices.

6. If a better parent and/or children are found with collision free path and lower cost, than the $[\mathbf{s}_{\text{new}}]$ parent and children information are updated.

These steps are repeated until the algorithm reaches K iterations. Thus, the BoxRRT* algorithm can improve the optimality of the solution, in terms of distance, over time even after the first solution is found.

Application:

The BoxRRT* is performed on the simple car model which involves nonholonomic constraints. The resulted for $K = 20000$ are reported and can be seen in Fig. 1.

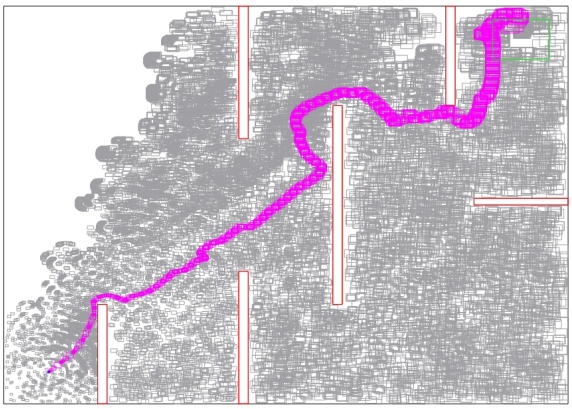


Figure 1: A BoxRRT* solution with $[\mathbf{s}_{\text{init}}]$ the blue box and $[\mathbf{s}_{\text{init}}]$ the green one while the obstacles are in red: CPU = 745 [s], the number of vertices and the planned path length are 96 and 105 [cm], respectively, while the total number of vertices is 13655.

Acknowledgement

This research was financial supported by the DGA MRIS.

References

[**1**] J. Alexandre dit Sandretto and A. Chapoutot. DynIBEX: a differential constraint library for studying dynamical systems (poster), HSCC, ACM, 2016.

[**2**] S. Karaman and E. Frazzoli. Optimal kinodynamic motion planning using incremental sampling-based methods. CDC 2010, pages 7681-7687.

[**3**] R. Pepy, M. Kieffer, and E. Walter. Reliable robust path planning with application to mobile robots. Int. J. Appl. Math. Comput. Sci., 19(3):413-424, 2009.

Vision based Pose domain characterization of an Unmanned Aerial Vehicle using Interval Analysis

Ide-Flore Kenmogne[1], Vincent Drevelle[1,2] and Eric Marchand[1,2]

[1] IRISA-INRIA Rennes Bretagne Atlantique,Rennes, France
ide-flore.kenmogne@inria.fr .
[2] Université de Rennes 1, Rennes, France
{vincent.drevelle,eric.marchand}@irisa.fr .

Keywords: Pose estimation, Set Inversion, Fault detection

Introduction

To navigate and accomplish their tasks, Unmanned Aerial Vehicles (UAVs) need to locate themselves with respect to the environment and have confidence information about their position. An onboard camera can be used in order to enhance the robot localization using an image-based primary positioning system which enables to overcome GPS and compass unreliability in difficult environments. Solutions to pose estimation from a set of known landmarks ([1]) exist in Computer Vision but they classically provide a punctual estimate of the location. Considering image measurements and landmark positions uncertainties, we aim at characterizing a domain that contains the 3-D pose of a UAV equipped with a camera and proprioceptive sensors. We use an interval-based set-membership approach [2]; which is a powerful tool for rigorous uncertainty propagation([3],[4]).

Problem statement

Estimating the pose of a camera consists in determining the transformation between the world frame $\mathcal{F}_w$ and the camera frame $\mathcal{F}_c$ (Fig. 1).

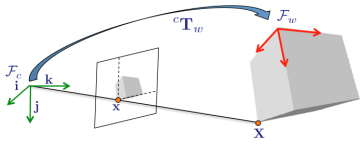


Figure 1: Perspective projection of an object in the image plane

Let ${}^c\mathbf{T}_w$ be this transformation, defined by: ${}^c\mathbf{T}_w = \left({}^c\mathbf{R}_w \mid {}^c\mathbf{t}_w \right)$, where ${}^c\mathbf{R}_w$ and ${}^c\mathbf{t}_w$ is a function of the attitude and position of the camera in the world frame. To solve this problem, the perspective projection equation of a 3-D point (in the world frame) on the image frame (a 2-D point) is used.

$$\mathbf{x} = \mathbf{K}\,\mathbf{\Pi}\,{}^c\mathbf{T}_w\,{}^w\mathbf{X} \tag{1}$$

with ${}^w\mathbf{X} = (X, Y, Z, 1)^\top$ the homogeneous 3-D point coordinates; $\mathbf{x} = (u, v, 1)^T$ the pixel coordinates (projection of ${}^w\mathbf{X}$ in the image); $\mathbf{K}$ the camera intrinsic parameters matrix and $\mathbf{\Pi}$ the perspective projection matrix.

Supposing we have N points ${}^w\mathbf{X}_i, i = 1..N$ in $\mathcal{F}_w$ and their projections $\mathbf{x}_i$; pose estimation amount in solving the system of equations (1) for ${}^c\mathbf{T}_w$. This is an inverse problem that is known as the Perspective from N Points problem or PnP. It is classically solved by minimizing the norm of the reprojection error using a non-linear minimization such as a Gauss-Newton of a Levenberg-Marquardt technique.

Interval based Pose Estimation

Placing ourselves in the context of bounded error measurements, each image point $\mathbf{x}_i$ and world point ${}^w\mathbf{X}_i$ can be represented as an interval vector. Instead of computing the pose by solving equation (1), our approach rely on seeking the domain of all the feasible poses $\mathbf{q}$ such that (1) is verified for $i = 1..N$ according to 2D-3D point correspondences. It consists in computing the solution set :

$$Q = \{\mathbf{q} \mid \exists {}^wX \in [{}^wX], K\Pi {}^cT_w(\mathbf{q}) {}^wX \in [x])\} \tag{2}$$

, which is the set of all feasible pose compatible with $[\mathbf{x}], [{}^{w}\mathbf{X}]$. The initial domain of the altitude, pitch and roll components of $\mathbf{q}$ is set from onboard sensors measurements. We compute an outer subpaving of Q using Interval Analysis.

Main results

Figure 2: *Left:* UAV in the room. *Right:* Onboard camera view

Experimental trials have been conducted with a quadcopter UAV MK-Quadro from MikroKopter (left image of Fig. 2). This quadcopter is equipped with an onboard camera for image acquisition. Six cubes are used as landmarks of known coordinates and tracked in the image.

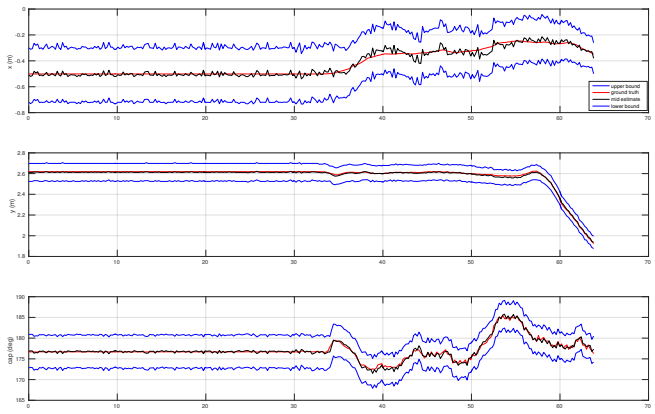


Figure 3: Bounds of the computed pose domain in the whole trial with $\pm 0.5px$ error and $\pm 1cm$ landmark coordinates error. *Black*: mid taken as punctual estimate, *Red*: ground truth and *Blue*: Lower and Upper bounds of the domain

Measurements Uncertainties influence on pose domain size

We can observe the evolution of the pose domain w.r.t. an increasing error bounds in the image measurements in Fig. 4.

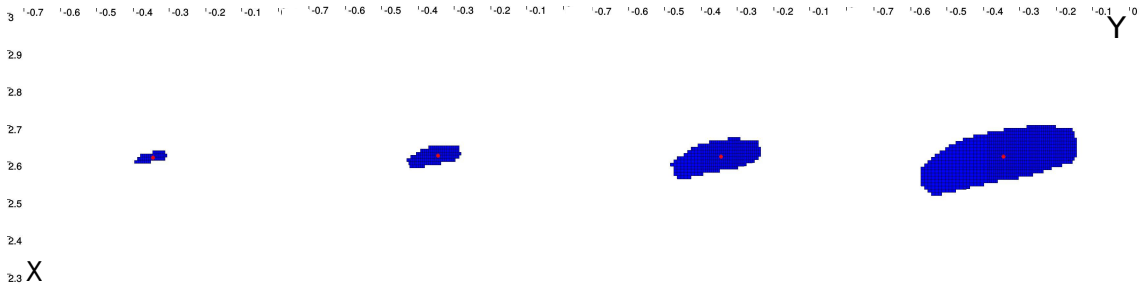


Figure 4: X,Y plane of domain subpaving w.r.t. increasing image tracking error bounds: $\pm 0.25, \pm 0.5, \pm 1, \pm 2$ px

Fault detection and restart

It may happen that no solution can be found with the current set of measurements. In this case, the method outputs an empty set. Possible causes are modeling errors, underestimation of measurement error bounds (it is not generally possible to fix a tight error bound that will cover even rare events), or the presence of spurious measurements. The latter case happens when the landmark tracking algorithm fails. When an empty solution set occurs, a "fault detected" flag is raised, and the landmark tracking is reset.

Conclusion

We proposed an interval based set-membership approach to compute a domain that contains the pose of an UAV, from uncertain bounded-error measurements of known landmarks in the image. While interval methods provide guaranteed results as long as the measurement errors bounds are not violated, setting guaranteed measurement error bound in practice is generally impossible or very pessimistic. Fault detection

system is thus implemented, in order to cope with inconsistencies due to tracking errors.

References

[**1**] E. Marchand and H. Uchiyama and F. Spindler, Pose estimation for augmented reality: a hands-on survey, *IEEE Transactions on Visualization and Computer Graphics* 22,2633-2651,2016.

[**2**] Jaulin, Luc and Kieffer, Michel and Didrit, Olivier and Walter, Eric, *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*, Springer Verlag,2001.

[**3**] B. Telle and M. J. Aldon and N. Ramdani, Guaranteed 3D visual sensing based on interval analysis, *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on* 2,1566-1571,2003.

[**4**] V. Drevelle and P. Bonnifait, Localization Confidence Domains via Set Inversion on Short-Term Trajectory, *IEEE Transactions on Robotics* 29,1244-1256,2013.

Interval Methods for Resolving Neural Computation Issues

S. P. Adam[1,2], D. A. Karras [3], M. D. Magoulas [4] and M. N. Vrahatis [2]

[1] Dept. of Computer Engineering, Technological Education Institute of Epirus, Arta, Greece,
adamsp@teiep.gr
[2] Dept. of Mathematics, University of Patras, Greece,
[3] Dept. of Automation, Technological Educational Institute of Sterea Hellas, Greece,
[4] Dept. of Computer Science and Information Systems, Birkbeck College, University of London, UK

Keywords: Neural computations, Interval methods, Reliable computing

Introduction

Effective handling of uncertainty constitutes an important issue when dealing with systems built on concepts and methods from the areas of Computational Intelligence. Uncertainty in such systems appears in various forms both when modeling a real process, as well as, during deployment and exploitation of a system, and typically, it is related to the input data and to model parameters. A multitude of research efforts are reported in the literature, concerning its quantification and effective handling. The majority of these efforts adopt concepts and methods arising from areas such as, probabilities and stochastic processes, Bayesian theory, fuzzy logic, mathematical theory of evidence, rough sets, etc. However, these approaches have no means to handle numerical errors, while at the same time it is questionable whether they can effectively tackle uncertainty without specific knowledge of the underlying process, such as the knowledge of an expert, or in the absence of known probability distributions, etc.

Interval arithmetic was introduced as a means to perform numerical computations with guaranteed accuracy and bounding the ranges of the quantities, used in the computations. Nowadays, interval analysis offers a whole toolbox for providing reliable solutions to several problems, especially, when they concern processes for which the proposed model can be cast in some closed-form or analytic expression.

The study and the use of interval analysis has attracted the interest of researchers in the area of computational intelligence, and more specifically, in the field of neural networks. Various trends are reported in the literature

regarding the use of interval analysis approaches in artificial neural networks. Most of them focus on endowing neural networks with the capability to process uncertain data expressed in the form of intervals, while the "maximalist approach", in this direction, concerns the conception and the implementation of interval neural networks i.e., neural architectures which not only are capable to process interval valued data but they, also, dispose some suitable training mechanism based on interval optimization techniques, [1]. Other notable research efforts dealing with the integration of the ability to process interval data into classical neural networks, range from, suitably modifying neural architectures, to proposing interval-like versions of the classical training algorithms such as gradient descent, [2,3].

Recent Research Results

Recently, aiming at these objectives has been reconsidered from a rather different point of view. In contrast to maximalist approaches or to some interval-mimicking techniques and algorithms, the work of Adam et al. [4,5,6] focused on using interval methods in order to resolve specific neural computation issues for which there exists some appropriate transcription in terms of intervals. In order to derive intervals for the values of the critical parameters of the networks, we had to revisit fundamental neural network concepts and well known approaches and identify the level of effective interventions. On the other hand, in order for these interventions to be successful we had to study and define the necessary theoretical tools for providing support to the effective application of the interval methods. The issues, presented hereafter, were identified for potential improvement or resolution.

Effective weight initialization of a neural network, [4]: The problem of determining good initial conditions for a local search algorithm used to train a multi-layer perceptron (MLP) was studied. For each node in the hidden layer synaptic weights are considered to be located within the bounds of some unknown intervals. These intervals together with the intervals of the values of the signals, input to a node, form an interval linear system corresponding to a linear interval tolerance problem. A number of theoretical results are proved and a new algorithm is proposed for solving this problem and, hence, for defining effective intervals for the initial weights. The proposed approach inherently includes some of the major concepts involved in neural network weight initialization, such as: the number of inputs to a node in the first hidden layer, the statistical information of the input data, effective

positioning of the hyper-planes in the pattern space and full utilization of the dynamic range of the activation function. The proposed method is tested on a number of well known benchmarks for MLPs trained with some well known back-propagation algorithms and the experimental results obtained are compared against the results of a number of well known and established weight initialization methods.

Definition of the area in the weight space of an MLP where a global minimizer of the network's output error function is guaranteed to be located, [5]: Using global optimization techniques for neural network training has been an important issue in the field of neural computation as these techniques succeed to find a global minimizer of the network's error function while avoiding the problems related to local search. However, a major problem that still remains to be solved concerns the region where these methods will effectively search for some global optimizer. Given that the weight space of an MLP is unbounded, the current practice on this matter consists in, heuristically, defining a bounded region hoping that some global optimizer is contained in there. The approach elaborated in this research relies on interval analysis and defines guaranteed bounds of the region in the search space where some global search algorithm should operate when training an MLP. These bounds, depend on the machine precision set for the solution of the problem and the term guaranteed denotes that the region determined surely encloses weight vectors that are global minimizers of the neural network's error function. Generally, the solution set of this bounding problem of an MLP is non-convex. However, the theoretical results elaborated helped deriving a box which is a convex set. This box is an outer approximation of the algebraic solutions to the interval equations resulting from the functions implemented by the network nodes.

Reliable estimation of an MLP's domain of validity, [6]: The quality of training of a neural network is represented to a great extent by its domain of validity as it helps to assess the network's ability to cope with a given problem. A number of research efforts can be found in the literature on this matter aiming to provide as accurate estimations of the domain of validity as possible. Given that the dependence of a neural network output on the pattern data is a nonlinear function, in this research, we consider that derivation of the area in the input space, effectively taken into account by the neural network function, can be addressed as a nonlinear parameter estimation problem. Hence, this problem can be tackled by SIVIA, the approach

originally introduced by Jaulin and Walter [7]. The use of interval computation, obviously, guarantees the reliability of the results in terms of accurately detecting the domain of validity. The proposed method was experimentally tested on a number of problems and the results obtained proved to be very promising for obtaining reliable conclusions on aspects of the neural network such as its ability to generalize well and its ability to be explicative.

References

[**1**] M. Beheshti, A. Berrached, A. de Korvin, C. Hu, and O. Sirisaengtaksin, On interval weighted three-layer neural networks, *Proceedings of the 31st Annual Simulation Symposium.* IEEE Computer Society Press, 188–194, 1998.

[**2**] H. Ishibuchi and H. Tanaka, Regression analysis with interval model by neural networks, *Proc. of the IEEE International Joint Conference on Neural Networks, IJCNN'91*, 1594–1599, 1991.

[**3**] R. Ak, V. Vitelli, and E. Zio, An Interval-Valued Neural Network Approach for Uncertainty Quantification in Short-Term Wind Speed Prediction, *IEEE Transactions on Neural Networks and Learning Systems* (vol. 26) 11, 2787–2800, 2015.

[**4**] S.P. Adam, D.A. Karras, G.D. Magoulas, and M.N. Vrahatis, Solving the linear interval tolerance problem for weight initialization of neural networks, *Neural Networks* (vol. 54), 17–37, 2014.

[**5**] S.P. Adam, G.D. Magoulas, D.A. Karras, and M.N. Vrahatis, Bounding the Search Space for Global Optimization of Neural Networks Learning Error: An Interval Analysis Approach, *Journal of Machine Learning Research* (vol. 17) 167, 1–40, 2016.

[**6**] S.P. Adam, D.A. Karras, G.D. Magoulas, and M.N. Vrahatis, Reliable estimation of a neural network's domain of validity through interval analysis based inversion, *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 1–8, 2015.

[**7**] L. Jaulin and E. Walter, Set Inversion Via Interval Analysis for nonlinear bounded-error estimation, *Automatica* (vol. 29) 4, 1053–1064, 1993.

Tight interval state estimator based on output set-inversions

Nacim Meslem[1] and Nacim Ramdani[2]

[1] Univ. Grenoble Alpes, CNRS, GIPSA-lab,
F-38000 Grenoble, France
nacim.meslem@gipsa-lab.fr
[2] Univ. Orléans, INSA Centre Val de Loire (CVL), PRISME
45000 Orléans, France.
nacim.ramdani@univ-orleans.fr

Keywords: Discrete-time linear systems, Observability, Set-membership state estimation, Interval analysis, Set-inversion

Introduction

Based on the classical observability property of discrete-time linear systems, a new interval state estimator is introduced in this work. Although interval computation is used to design the proposed state estimator, the convergence analysis of the width of the estimated state enclosures is fully characterized. Compared to the existing approaches in the literature [1, 2, 3, 4], the proposed interval method does not require solving optimization problems at each measurement time instant. It computes in a direct way maximal and minimal bounds for the solution sets. Therefore, this interval estimation approach allows saving computation time while guaranteeing tightness of the state enclosures.

Problem statement

Consider an uncertain discrete-time linear system described by

$$\left\{ \begin{array}{rcl} \mathbf{x}_{k+1} & = & \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{E}\mathbf{d}_k \\ \mathbf{y}_k & = & \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \mathbf{F}\mathbf{v}_k \end{array} \right. \tag{1}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the state vector, $\mathbf{u}_k \in \mathbb{R}^m$ is the input vector and $\mathbf{y}_k \in \mathbb{R}^p$ is the measured output vector. The vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ are respectively the state perturbation and the measurement noise, which act on the system. These quantities are assumed unknown-but-bounded with known bounds, i.e.

$$\forall k \geq 0,\ \mathbf{d}_k \in [\underline{\mathbf{d}},\ \overline{\mathbf{d}}] \subset \mathbb{R}^{n_d} \text{ and } \mathbf{v}_k \in [\underline{\mathbf{v}},\ \overline{\mathbf{v}}] \subset \mathbb{R}^{n_v} \tag{2}$$

where the real vectors $\underline{\mathbf{d}}$ and $\overline{\mathbf{d}}$ (resp. $\underline{\mathbf{v}}$, $\overline{\mathbf{v}}$) are the perfectly known endpoints of the box $[\mathbf{d}]$ (resp. $[\mathbf{v}]$). The matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}$ and $\mathbf{F}$ have the appropriate dimensions.

This work proposes an interval estimator of all the possible state trajectories of the uncertain system (1). More precisely, based on interval analysis, an algorithm is designed to generate an interval sequence, $[\mathbf{x}_k]$ $k \in \{1, \ldots, N\}$, which is tight, i.e. it frames tightly all the possible state trajectories $\mathbf{x}_k$ of system (1).

Main result

The proposed algorithm is a *Prediction-Correction* state estimator.

- The prediction stage is carried out by the following formula,

$$[\mathbf{x}_k]^p = \mathbf{A}^k[\mathbf{x}_s] + \big(\sum_{i=0}^{k-1} \mathbf{A}^i\big)\mathbf{E}[\mathbf{d}] + \sum_{i=0}^{k-1} \mathbf{A}^{(k-i-1)}\mathbf{B}\mathbf{u}_{s+i} \tag{3}$$

  That means from a given time instant t_s for which the state of the system is included in the box $[\mathbf{x}_s]$, all the upcoming state enclosures, $[\mathbf{x}_k]$ $k > s$, of the system (1) can be computed, in a guaranteed way, by (3).

- The correction stage is based on the observability of the pair $(\mathbf{A}, \mathbf{C})$. If the system (1) is observable, the state enclosure of the system at the time instant t_k can be computed from the future sequences of its input and output as follows

$$[\mathbf{x}_k]^c = [\mathbf{x}_k]^p \cap \mathcal{O}^{-1}\big([\mathbf{Y}_{(k:k+n-1)}] - \mathcal{O}_d\mathbf{E}[\mathbf{d}] - \mathcal{O}_u\mathbf{U}_{(k:k+n-1)}\big) \tag{4}$$

where

$$\mathcal{O} = \begin{pmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{pmatrix}, \quad \mathcal{O}_d = \begin{pmatrix} \mathbf{0} \\ \mathbf{C} \\ \vdots \\ \mathbf{C}\sum_{i=0}^{n-1} \mathbf{A}^i \end{pmatrix}, \quad \mathcal{O}_u = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{CB} & \mathbf{D} & \dots & \vdots & \vdots \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \mathbf{CA}^{n-1}\mathbf{B} & \mathbf{CA}^{n-2}\mathbf{B} & \dots & \mathbf{CB} & \mathbf{D} \end{pmatrix}$$

and

$$[\mathbf{U}_{(k:k+n-1)}] = \begin{pmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \vdots \\ \mathbf{u}_{k+n-1} \end{pmatrix}, \quad [\mathbf{Y}_{(k:k+n-1)}] = \begin{pmatrix} \mathbf{y}_k^m \\ \mathbf{y}_{k+1}^m \\ \vdots \\ \mathbf{y}_{k+n-1}^m \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} [\mathbf{e}_y] \tag{5}$$

In (5), the vector $\mathbf{y}_k^m$ stands for sensors data at the time instant t_k and $[\mathbf{e}_y]$ stands for the box of the feasible output error.

The main novelty of this work is implemented by the following algorithm.

Algorithm: Box-State-Est($[\mathbf{x}_0]$, N)

- For $k := 1$ to $k := n-1$
  1. $[\mathbf{x}_k]^p := \mathbf{A}^k[\mathbf{x}_0] + \big(\sum_{i=0}^{k-1} \mathbf{A}^i\big)\mathbf{E}[\mathbf{d}] + \sum_{i=0}^{k-1} \mathbf{A}^{(k-i-1)}\mathbf{B}\mathbf{u}_i$
  2. $[\mathbf{x}_k] := [\mathbf{x}_k]^p$
- For $k \geq n-1$ to N
  3. $j := k - (n-1)$
  4. $[\mathbf{x}_j]^{inv} := \mathcal{O}^{-1}\big([\mathbf{Y}_{(j:j+n-1)}] - \mathcal{O}_d\mathbf{E}[\mathbf{d}] - \mathcal{O}_u\mathbf{U}_{(j:j+n-1)}\big)$
  5. $[\mathbf{x}_j]^c := [\mathbf{x}_j]^{inv} \cap [\mathbf{x}_j]^p$
  6. $[\mathbf{x}_k]^c := \mathbf{A}^{n-1}[\mathbf{x}_j]^c + \big(\sum_{i=0}^{n-2} \mathbf{A}^i\big)\mathbf{E}[\mathbf{d}] + \sum_{i=0}^{n-2} \mathbf{A}^{(n-i-2)}\mathbf{B}\mathbf{u}_{j+i}$
  7. $[\mathbf{x}_k] := [\mathbf{x}_k]^c \cap [\mathbf{x}_k]^p$
  8. $k := k+1$
  9. $[\mathbf{x}_k]^p := \mathbf{A}[\mathbf{x}_{k-1}] + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{E}[\mathbf{d}]$
- Return $[\mathbf{x}_k],\ k \in \{0, 1, 2, \dots, N\}$

Proposition: *If the uncertain system* (1)-(2) *is observable, the Algorithm* ***Box-State-Est*** *generates an interval sequence* $[\mathbf{x}_k],\ k \in \{0, 1, 2, \dots, N\}$ *such that*

$$\forall k \geq 0,\ \textit{the solution to } (1)\text{-}(2),\ \mathbf{x}_k \in [\mathbf{x}_k]$$

and the width of $[\mathbf{x}_k]$ *is lower than*

$$w\big([\mathbf{x}_k]\big) \leq \big|\mathbf{A}^{n-1}\mathcal{O}^{-1}\big| w\big([\mathbf{e}_y]\big) + \big|\big(\sum_{i=0}^{n-2}\mathbf{A}^i - \mathbf{A}^{n-1}\mathcal{O}^{-1}\mathcal{O}_d\big)\mathbf{E}\big| w\big([\mathbf{d}]\big)$$

References

[**1**] L. Chisci, A. Garulli, and G. Zappa, Recursive state bounding by parallelotopes, *Automatica*, 32:1049–1055, 1996.

[**2**] Polyak, S. A. Nazin, C. Durieu, and E. Walter, Ellipsoidal parameter or state estimation under model uncertainty, *Automatica*, 40:1171–1179, 2004.

[**3**] T. Alamo, J. M. Bravo, and E. F. Camacho, Guaranteed state estimation by zonotopes, *Automatica*, 41:1035–1043, 2005.

[**4**] S. Ben Chabane, C. Stoica Maniu, T. Alamo, E. F. Camacho, and D. Dumur, Improved set-membership estimation approach based on zonotopes and el- lipsoids, *In Proc. of the IEEE European Control Conference*, pages 993–998, 2014. Strasbourg, France.

A bounded-error quaternion-based attitude estimation approach.

Nacim Ramdani and Sylvain Miossec

Univ. Orléans, INSA Centre Val de Loire, PRISME EA 4229,
F45000 Orléans, France.
nacim.ramdani@univ-orleans.fr

Keywords: Estimation, Quaternion, Bounded-Error, Monitoring.

Attitude estimation

Motion capture and analysis is a very active research field with many applications in areas such as computer animation, video games, serious games, medical therapy, tele monitoring, and human robot interaction.

The estimation of the orientation of a rigid body relative to an inertial frame may be achieved using two main technologies. Having in mind application in serious games and tele-monitoring, these sensing technologies can be grouped in two families. The first type of sensing uses RGB-D type sensors, a.o. Kinect type sensors, that need not be attached on the body. To the contrary, the second type of sensing modality is achieved by Inertial Measurement Units IMU which take the form of body-attached equipments that combine gravity and magnetic fields observation with measurement of the sensor angular velocity to reconstruct the sensor attitude.

Both sensing modalities are used in wide application areas. For instance, Bonnet et al. [2] used a Kinect sensor to develop an affordable mobile platform for pathological gait analysis, while Bethencourt and Jaulin [1] used interval analysis to develop a method for Visual Simultaneous Localisation and Mapping (VSLAM) coupling a Kinect sensor with an Inertial Measurement Unit (IMU). At the price of a dramatically reduced performance, RGB-D sensors offer low-cost motion capture solutions For example, the joint position data from the RGB-D sensor Microsoft Kinect usually exhibits significant jitters caused by

low depth accuracy, occlusions, ambiguity, and loss of tracking, and the body segment lengths vary during the motion [7]. Furthermore, IMU data are sensitive to perturbations such as cartesian acceleration, magnetic disturbance, and bias in the measurement of angular velocities.

In this work we investigate the potentials of bounded-error estimation via interval analysis for accurate attitude estimation from IMU sensors data. Our objective is to develop robust reconstruction techniques to equip the new tele-monitoring system under construction within the H2020 RISE PROPHETIC personal healthcare service for holistic remote management and treatment of Parkinson patients [6].

Modelling

The orientation of a rigid body in space is determined when the axis orientation of a coordinate frame attached to the body, the body frame, $\mathcal{B}$ is specified with respect to an absolute coordinate systems, usually called the navigation frame $\mathcal{N}$. Here, we use the unit orientation quaternions, $q \in \mathbb{Q}$, to characterize the body attitude

$$q = [s \quad \mathbf{v}^T]^T, \quad q^T q = 1, \quad \mathbf{v}^T = [v_x \quad v_y \quad v_z]^T \tag{1}$$

The rigid body angular motion obeys the vector differential equation

$$\dot{q} = \frac{1}{2} q \otimes \omega_q = \frac{1}{2} \begin{bmatrix} -\mathbf{v}^T \\ I_3 s + [\mathbf{v}^\times] \end{bmatrix} \cdot \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \frac{1}{2} [\Omega(\omega)] \cdot q, \tag{2}$$

where $\omega = [\omega_x \quad \omega_y \quad \omega_z]^T$ is the vector of angular velocities, that are measured by gyrometers, and where

$$[\mathbf{v}^\times] = \begin{bmatrix} 0 & -v_x & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \tag{3}$$

The other measured data are gathered in the total acceleration vector $\mathbf{f}$, and in the magnetic vector $\mathbf{h}$. The observation model is given by

$$\mathbf{f} = [f_x \quad f_y \quad f_z]^T = M_N^B(q) \cdot (\mathbf{g} - \mathbf{a}) + \delta_f \tag{4}$$
$$\mathbf{h} = [h_x \quad h_y \quad h_z]^T = M_N^B(q) \cdot \mathbf{m} + \delta_h \tag{5}$$

where

$$M_N^B(q) = \begin{bmatrix} 2(s^2 + v_x^2) - 1 & 2(v_x v_y + s v_z) & 2(v_x v_z - s v_y) \\ 2(v_x v_y - s v_z) & 2(s^2 + v_y^2) - 1 & 2(s v_x + v_y v_z) \\ 2(s v_y + v_x v_z) & 2(v_y v_z - s v_x) & 2(s^2 + v_z^2) - 1 \end{bmatrix}, \tag{6}$$

and where $\mathbf{a}$ is the body cartesian acceleration, $\mathbf{g} = [0 \quad 0 \quad -1]^T$ is the normalized gravity, and $\mathbf{m} = [\cos(I) \quad 0 \quad -\sin(I)]^T$ the earth magnetic field, with known tilt angle I. δ_f and δ_h are bounded noise vectors.

Main results

The state-of-the-art non-linear bounded-error estimation techniques [4] are used within a quaternion-based formulation.

Prior to the estimation, the actual data are pre-processed using a complementary filtering approach [5, 3].

Then, the predictor-corrector approach is used. In the prediction stage, the Picard-Lindelöf operator is used with Eq.(2) to compute the reachable quaternions. In the correction stage, solution techniques for non linear constraint satisfaction problems are used to prune off inconsistent quaternions.

Results will be given using actual IMU data.

Acknowledgement

This work is supported by the MSCA-RISE-2014: under grant agreement "PROPHETIC: An innovating Personal Healthcare Service for a holistic remote management and treatment of Parkinson patients." Grant agreement no: 645741. (http://www.propheticproject.eu).

References

[1] A. Bethencourt and L. Jaulin. 3D reconstruction using interval methods on the Kinect device coupled with an IMU. *International Journal of Advanced Robotic Systems*, 10(93), 2013. doi: 10.5772/54656.

[2] V. Bonnet, C. A. Coste, L. Lapierre, J. Cadic, P. Fraisse, R. Zapata, G. Venture, and C. Geny. Towards an affordable mobile analysis platform for pathological walking assessment. *Robotics and Autonomous Systems*, 66:116 – 128, 2015.

[3] H. Fourati, N. Manamanni, L. Afilal, and Y. Handrich. A nonlinear filtering approach for the attitude and dynamic body acceleration estimation based on inertial and magnetic sensors: Bio-logging application. *IEEE Sensors Journal*, 11(1):233–244, Jan 2011.

[4] M. Kieffer, L. Jaulin, and E. Walter. Guaranteed recursive nonlinear state bounding using interval analysis. *International Journal of Adaptative Control and Signal Processing*, 16:193–218, 2002.

[5] R. Mahony, T. Hamel, and J. M. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5):1203–1218, June 2008.

[6] PROPHETIC. An innovating personal healthcare service for a holistic remote management and treatment of parkinson patients. h2020 MSCA RISE 2014 grant agreement number 645741, 2015–2017. URL `www.propheticproject.eu`.

[7] Q. Wang, G. Kurillo, F. Ofli, and R. Bajcsy. Remote health coaching system and human motion data analysis for physical therapy with microsoft kinect. In *Computer Vision for Accessible and Affordable HealthCare Workshop (ICCV2016)*, 2016. URL `https://arxiv.org/abs/1512.06492`.

An Interval Approach to Multiple UAV Collision Avoidance

James A. Douthwaite, Allan De Freitas and Lyudmila S. Mihaylova

University of Sheffield, Dept. Automatic Control & Systems Engineering,
Sheffield, United Kingdom

{JADouthwaite1,a.defreitas}@sheffield.ac.uk

Keywords: Collision Avoidance, Unmanned Aerial Vehicle, Interval Analysis

Unmanned Aerial Vehicles (UAV) or Systems (UAS) are now entering the commercial and domestic sector as proposed solutions to many day-to-day tasks. Innate collision avoidance within such systems is considered a prerequisite when navigating dynamic environments, confined spaces or simply in the preservation of other systems or human safety. Upcoming legislation is setting the grounds for provably safe autonomous avoidance in [1]. Recent works in UAV collision avoidance, [2] and [3], demonstrate the challenges engineers face in meeting these design constraints. High fidelity systems; both in their obstacle tracking and intelligent decision making, and ability to cope with real world factors are currently in high demand.

Currently, numerous avoidance techniques have been applied under the premise of the *free flight* concept [2]. This paper draws attention to the non-cooperative case, typically where communication has failed and a *See-And-Avoid* (SAA) approach is necessary. Interval analysis is demonstrated as an effective tool for handling navigation scenarios in the presence of measurement uncertainty and non-linearity in [4]. This work demonstrates how Interval analysis is applied in the design of a robust non-cooperative SAA algorithm [2],[5] for autonomous UAVs. Obstacle trajectories are represented as bounded regions emulating measurement uncertainty from an on board camera and range-finder system (see Figure 1). Using this description, a geometric collision avoidance problem is posed incorporating the measurement uncertainty and aircraft dynamics.

The interval obstacle trajectories are used to define an interval estimate for time to collision. The required trajectory interval is then defined geometrically to satisfy the minimum separation constraint. Finally, an interval containing the optimal avoidance manoeuvre is defined and contracted before it is handed to the UAV's trajectory controller.

This work highlights how this approach can be extended to consider multiple obstacle trajectories (and their respective uncertainties) by defining the intersection of these optimal regions. Strategies for region selection where no intersection occur are presented in the design of a unified escape trajectory. Preliminary results are presented comparing the novel algorithm against existing geometric avoidance techniques in typical air-traffic and scenarios in addition to conditions more likely seen in uncoordinated UAV flight.

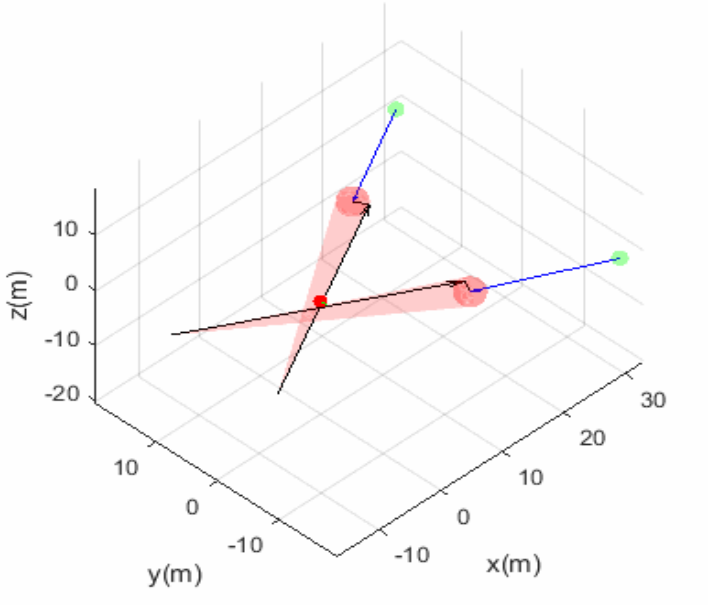


Figure 1: A depiction of the localised SAA problem, and the interval uncertainty in obstacle trajectory.

Acknowledgement

The authors gratefully acknowledge the support from the UK EPSRC under grant number EP/M506618/1.

References

[1] S. M. Veres, O. Mcaree, and J. M. Aitken, "On the complexity of formal verification of small and micro UAS," in *Proceedings of the European Control Conference 2016*, pp. 1–8, University Of Sheffield, 2016.

[2] B. M. Albaker and N. A. Rahim, "A survey of collision avoidance approaches for unmanned aerial vehicles," in *International conference for technical postgraduates (TECHPOS)*, pp. 1–7, IEEE, 2010.

[3] C. Goerzen, Z. Kong, and B. Mettler, *A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance*, pp. 65–100. Dordrecht: Springer Netherlands, 2010.

[4] L. Jaulin, "Robust set-membership state estimation ; application to underwater robotics," *Automatica*, vol. 45, no. 1, pp. 202–206, 2009.

[5] A. Zhahir, A. Razali, and M. R. Mohd Ajir, "Current development of UAV sense and avoid system," in *IOP Conference Series: Materials Science and Engineering*, vol. 152, (Universiti Putra Malaysia), pp. 012–035, 2016.

Robust Motion Planning Based on Sliding Horizon and Validated Simulation

Elliot Brendel, Julien Alexandre dit Sandretto, and Alexandre Chapoutot

ENSTA ParisTech, Université Paris-Saclay,
828, bd des maréchaux, 91762 Palaiseau Cedex, France
{brendel,alexandre,chapoutot}@ensta.fr

Keywords: Motion Planning, Validated Simulation

Introduction

Motion planning algorithms are a center piece in the control framework of mobile robots as they contribute to give them the ability of have autonomous behaviors. Furthermore, such class of algorithms is critical as a failure can cause the abort of the mission or can cause important amount of damage such as human loss. The validation of such algorithms is then mandatory in order to increase the confidence of the end users. However, those algorithms are also subject to constraints, *e.g.*, to reduce fuel consumption. So, computing a safe path is usually not enough, an optimal one is search to minimize some costs.

Moreover, one of the challenge in order to design robust and reliable motion planning algorithms is to take into account various sources of uncertainties. For example, the environment is not exactly known and some disturbance should have been considered. Mathematical models of the mobile robots are not perfect and usually come from some simplification in order to have efficient simulation activities. Lastly, computer-aided design usually produces approximated results as it is based on numerical methods which cannot produce close form solution of a problem, *e.g.*, the solution of an initial value problem for ordinary differential equations. The set-membership framework is suitable to deal with such kind of uncertainties.

Main contribution

The main contribution of this article is the combination of set-membership methods with optimizing approach. Hence, a correct-by-construction algorithm is defined with the intrinsic properties to be robust to uncertainties as it relies on set-membership approach [1]. Moreover, embedding the motion planning problem into a constraint satisfaction problem (CSP) [2], and more precisely into a global optimization framework [3], the proposed algorithm produces an optimal free-collision path with respect to a given cost function which is minimized.

Experimentation

The motion planning of an AUV which has to move the closest to the seabed is considered. Hence, the cost function is the depth of its gravity center. As safety constraints, we want to ensure that the AUV is closer to the seabed than the distance $d_{\max}$ and further than $d_{\min}$.

Dynamics of an AUV

The dynamics of gravity center of the AUV follows the ODE defined in [4] and it is such that

$$\begin{cases} \dot{x} &= v\cos\theta\cos\psi \\ \dot{y} &= v\cos\theta\sin\psi \\ \dot{z} &= -v\sin\theta \\ \dot{\psi} &= \frac{\sin\varphi}{\cos\theta}\cdot v\cdot u_1 + \frac{\cos\varphi}{\cos\theta}\cdot v\cdot u_2 \\ \dot{\theta} &= \cos\varphi\cdot v\cdot u_1 - \sin\varphi\cdot v\cdot u_2 \\ \dot{\varphi} &= -0.1\sin\varphi + \theta\cdot v\cdot(\sin\varphi\cdot u_1 + \cos\varphi\cdot u_2) \end{cases} \tag{1}$$

with $\mathbf{s} = (x, y, z, \psi, \theta, \varphi)$ is the state vector. It can be split into the vector (x, y, z) of the coordinates of the gravity center and the vector (ψ, θ, φ) of Euler angles; $\mathbf{u} = (u_1, u_2)$ is the control input vector; v is the velocity.

Note that (1) has been simplified by substituting $\tan\theta$ by θ in the definition of $\dot{\varphi}$ to avoid technical issues of the implementation.

Nonetheless, the algorithm remains valid.

Underwater environment and results

We define a function $(x, y) \mapsto \text{seabed}\,(x, y)$ which returns the depth of the seabed at the coordinates (x, y). We also define $d_{\min}$ and $d_{\max}$ two constants such that the AUV stays at a distance to the seabed between $d_{\min}$ and $d_{\max}$. Some constraints on AUV angles are considered: yaw and roll are bounded in an interval (to go in a quite straight way and to not capsize) and pitch is bounded by an extreme value (to limit the dive angle). Finally, in order to force the AUV to move forward through the x dimension, we impose $x_{\text{end}} > x_{\text{init}}$. Thus, the problem is to find the control $\mathbf{u}$ solution of

$$(P_{\text{AUV}}) : \begin{cases} \min_{\mathbf{u}} \quad z \\ \dot{\mathbf{s}} = \mathbf{f}(\mathbf{s}, \mathbf{u}) \\ z > \text{seabed}\,(x, y) + d_{\min} \\ z < \text{seabed}\,(x, y) + d_{\max} \\ x_{\text{end}} > x_{\text{init}} \\ \theta < 0.8 \\ \varphi, \psi \in [-0.5, 0.5] \end{cases}$$

Then the following seabeds are considered

- Seabed 1: $(x, y) \mapsto \text{seabed}\left(\frac{x-30}{20}, \frac{y}{2}\right) - 100$;
- Seabed 2: $(x, y) \mapsto \text{seabed}\left(\frac{y}{2}, \frac{x-30}{20}\right) - 100$;
- Seabed 3: $(x, y) \mapsto \text{seabed}\left(\frac{-x+30}{20}, \frac{y}{2}\right) - 100$;
- Seabed 4: $(x, y) \mapsto \text{seabed}\left(\frac{y}{2}, \frac{-x+30}{20}, \frac{y}{2}\right) - 100$.

Some results are given in Figure 1.

Acknowledgement

This work benefited of the "Chair Complex Systems Engineering – École polytechnique, THALES, DGA, FX, DASSAULT AVIATION,

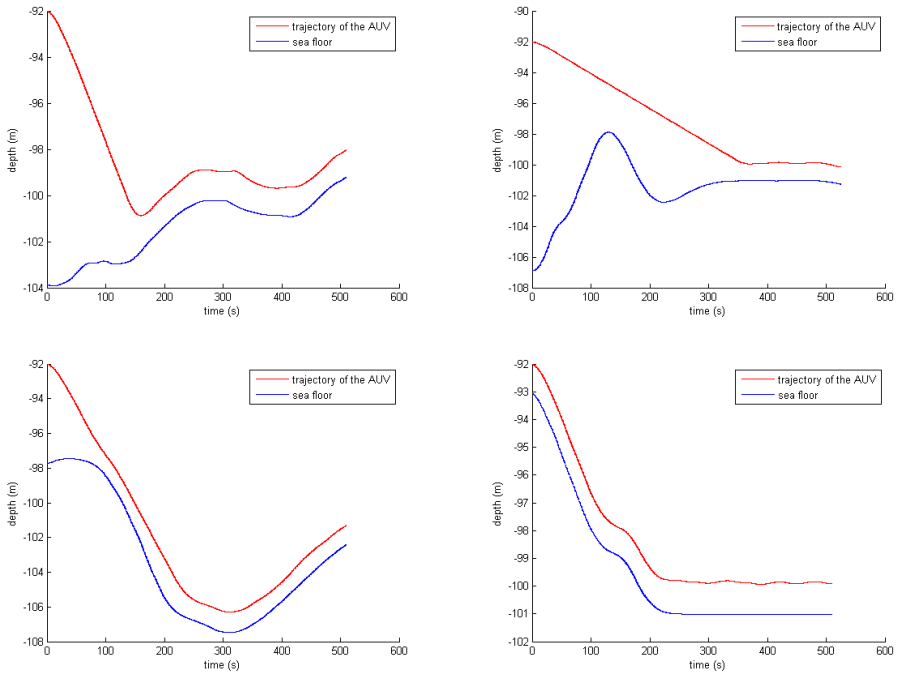


Figure 1: Results of the AUV motion planning with seabed 1 to 4 from top left to bottom right.

DCNS Research, ENSTA ParisTech, Télécom ParisTech, Fondation ParisTech and FDO ENSTA".

References

[1] L. Jaulin, M. Kieffer, O. Didrit, Olivier and É. Walter, *Applied Interval Analysis*, Springer 2001.

[2] M. Rueher, *Solving continuous constraint systems*, International Conference on Computer Graphics and Artificial Intelligence 2005.

[3] E. R. Hansen, *Global Optimization Using Interval Analysis*, Marcel Dekker Inc. 2003.

[4] L. Jaulin, *Mobile Robotics*, ISTE Press - Elsevier 2015.

Contractor Based Viability Algorithms

Stéphane Le Ménec[1]

[1] MBDA
1, Avenue Réaumur, 92358 Le Plessis-Robinson, France
stephane.le-menec@mbda-systems.com

Keywords: Viability Theory, Interval Computing, Contractor Programming

Introduction

Viability Theory [2] provides a collection of set concepts and algorithms to study the properties of dynamic systems. Thanks to paving techniques, interval computing provides powerful numerical methods to approximate sets [3]. Refined interval techniques as contractor programming and guaranteed integration allow to implement viability kernel and capture basin algorithms.

Viability Theory Definitions

Let us consider either a temporal interval $[0, T]$ with finite horizon or $[0, \infty[$ (denoted for simplicity by $[0, T]$ with $T := \infty$) with infinite horizon. We consider an (evolving) environment :

$$K(\cdot) : t \in [0, T], \;\; t \rightsquigarrow K(t) \subset X \tag{1}$$

(defined by viability constraints) and, when $T < +\infty$, an (evolving) target :

$$C(\cdot) : t \in [0, T], \;\; t \rightsquigarrow C(t) \;\subset\; K(t) \tag{2}$$

The usual case is a non evolving target for which $C(t) = \emptyset$ when $t < T$ and $C(T) = C$. An evolution $x(\cdot) : t \mapsto x(t)$ is said to be viable in $K(\cdot)$ if :

$$\forall\, t \in [0, T], \;\; x(t) \;\in\; K(t) \tag{3}$$

Finally, we introduce the right hand side $F : (x, t) \in X \times [0, T] \rightsquigarrow F(x(t), t) \in X$ of the differential inclusion :

$$\forall\, t \in [0, T], \;\; x'(t) \;\in\; F(x(t), t)) \tag{4}$$

We introduce the capture basin $\mathrm{Capt}_F(K, C)$ of the target $C(\cdot)$ viable in the environment $K(\cdot)$ as the set of initial states $x_0 \in K_0$ such that there exist :

1. a finite horizon $T^\star$,

2. at least one evolution $x(\cdot)$ starting at x_0

   (a) defined on $[0, T^\star]$,

   (b) governed by the differential inclusion $x'(t) \in F(x(t), t)$,

   (c) viable in $K(\cdot)$,

   (d) reaching the target at time $T^\star$: $x(T^\star) \in C(T^\star)$.

   Although dynamical games make sense only in finite horizon temporal case, in the infinite (or perennial) case $T = +\infty$ for which the target is empty, we introduce the viability kernel $\mathrm{Viab}_F(K)$ as the set of initial states $x_0 \in K_0$ such that there exist :

   (a) at least one evolution $x(\cdot)$ starting at x_0

      i. governed by the differential inclusion $x'(t) \in F(x(t), t)$,

      ii. viable in $K(\cdot)$,

   without closed subsets $C \subset K$, regarded as a target contained in an environment K. An evolution $x(\cdot)$ is said to be viable in K on a temporal interval $[0, T]$ if :

$$\forall\, t \in [0, T], \quad x(t) \in K(t) \tag{5}$$

Viability Algorithms

Once a differential inclusion $x'(t) \in F(x(t))$ has been discretized in time by $x_{n+1} \in \Phi(x_n)$ and "restricted" to grids of the finite dimensional vector space, then the viable capture basin $\mathrm{Capt}_\Phi(K, C)$ of elements of K from which an evolution (x_n) viable in K reaches the target C in finite discrete time can be obtained by two algorithms :

1. The *capture basin algorithm.* It is based on the formula :

$$\mathrm{Capt}_\Phi(K, C) = \bigcup_{n \geq 0} C_n \tag{6}$$

   where the *increasing* sequence of subsets $C_n \subset \mathrm{Capt}_\Phi(K, C)$ is iteratively defined by :

$$\left\{ \begin{array}{l} C_0 = C \\ \forall\, n \geq 0, \quad C_{n+1} := K \cap (C_n \cup \Phi^{-1}(C_n)) \end{array} \right. \tag{7}$$

2. And the *viability kernel algorithm.* Whenever $K \setminus C$ is *repeller* (for all $x \in K \setminus C$, all evolutions $x(\cdot)$ leave $K \setminus C$ in finite time), there is another class of general algorithms allowing to compute viable capture basins :

$$\mathrm{Capt}_\Phi(K, C) = \bigcap_{n \geq 0} K_n \tag{8}$$

where the *decreasing* sequence of subsets $K_n \supset \text{Capt}_\Phi(K, C)$ is iteratively defined by :

$$\left\{ \begin{array}{l} K_0 \;=\; K \\ \forall\, n \geq 0, \;\; K_{n+1} \;:=\; C \cup (K_n \cap \Phi^{-1}(K_n)) \end{array} \right. \tag{9}$$

Naturally, both subsets C_n and K_n are computed at each iteration subsets on a grid of the state space. The convergence of the subsets C_n and the K_n follows from convergence theorems presented in Chapter 19, p. 769, of *Viability Theory, New Directions*, [2] (see for instance Theorem 19.3, p. 774).

Contractor Based Viability Set Computing

One of the main objective of Viability Theory is to find controls such that at least one evolution satisfies environment constraints. Therefore, when defining contractor based viability algorithms, we introduce control $u \in U$. We define $C_{\Phi\,\exists u}\,([x_n,\, x_{n+1}])$ a contractor (on the box $[x_n,\, x_{n+1}]$) with quantifier (on control u) associated to the flow described hereafter.

$$C_{\Phi\,\exists u}\,([x_n,\, x_{n+1}]) \,:\exists\, u \in U \quad | \quad x_{n+1} = \Phi\,(x_n,\, u) \tag{10}$$

$C_{\Phi\,\exists u}([x_n,\, x_{n+1}])$ is an Ordinary Differential Equation (ODE) contractor [1] between state vectors x_n and x_{n+1}. Using ODE contractors with quantifiers, the sequence of sets K_n is computed, in the form of interval lattices, always in an iterative manner, as follow :

$$K_0^{\subset} \;=\; K^{\subset} = K \setminus X \tag{11}$$

$$K_{n+1}^{\subset} = \left(\bigcup_{[x] \subset K_n \,/\, C_n} \; \bigcap_{[y] \subset K_n} \left([x] - Proj_x\; (C_{\Phi\,\exists u}\,([x,\, y])) \right) \right) \bigcup K_n^{\subset} \tag{12}$$

$$n \;\geq\; 0 \tag{13}$$

with $Proj_x\,([x,\, y]) = [x]$, the projection operator along the $[x]$ dimension :

$$Proj_x \,:\, \mathbb{IR}^{2m} \;\rightarrow\; \mathbb{IR}^m \;(m : \text{state vector dimension of } x) \tag{14}$$

In a similar manner, we also re-write the capture basin algorithm using contractor programming.

Conclusion

Contractor programming is used to implement the capture basin and the viability kernel algorithms that are two types of algorithms to compute $\text{Capt}_\Phi(K, C)$. $\text{Capt}_\Phi(K, C)$ is the set of elements of K from which an evolution (x_n) viable in K reaches the target C in finite discrete time.

Acknowledgement

This work is supported by "Agence Nationale de la Recherche" through the ANR ASTRID Maturation funding scheme, VIATIC^2 project (N^o ANR-14-ASMA-0005-02) and ANR CONTREDO (N^o ANR-16-CE33-0024-06). The author would like also to thank the viability theoreticians who suggested remarks and advices to improve this work.

References

[1] J. Alexandre Dit Sandretto, and A. Chapoutot, *Contraction, propagation and bisection on a validated simulation of ODE*, Conference on Hybrid Systems: Computation and Control (HSCC 2016), Lyon, France, June 2016.

[2] J.P. Aubin, A. Bayen and P. Saint-Pierre, *Viability Theory, New Directions, Second Edition*, Springer-Verlag, ISBN 978-3-642-16683-9, 2011.

[3] L. Jaulin, M. Kieffer, O. Didrit and E. Walter, *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer, ISBN 1852332190, 2001.

Eulerian state estimation

Thomas Le Mézo, Luc Jaulin and Benoît Zerr
{thomas.le_mezo,luc.jaulin,benoit.zerr}@ensta-bretagne.fr
ENSTA-Bretagne, LabSTICC, CNRS 6285,
2 rue François Verny, 29806 Brest, France.

June 2, 2017

Abstract

In this paper, we present a tool for computing an inner and an outer approximations of the largest positive invariant set associated with a nonlinear state equation. Further, we show how to solve rigorously complex problems related to continuous time dynamical systems, such as the Eulerian state estimation problem.

1 Introduction

Invariant sets are used in nonlinear control theory [2] [3], for instance to validate *(i)* some properties of cyber-physic systems [10][27], *(ii)* to ensure the safe take off [25] of an airplane or *(iii)* to avoid collisions [6] with other aircrafts. In this paper, we deal with a dynamical system $\mathcal{S}$ defined by the following state equation:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector and $\mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^n$ is the evolution function of $\mathcal{S}$ [7, 8]. Denote by $\boldsymbol{\varphi}$ the flow map of $\mathcal{S}$, *i.e.*, with the initial condition $\mathbf{x}_0 = \mathbf{x}(0)$, the system $\mathcal{S}$ reaches the state $\boldsymbol{\varphi}(t, \mathbf{x}_0)$ at time t.

Two different types of approaches [16] are used to deal with the estimation of the solution for (1): the Lagrangian and the Eulerian. This classification is taken from the field of fluid mechanics [11]. In the Lagrangian point of view, the observer follows an individual fluid parcel as it moves through the fluid. In an Eulerian point of view, the observer stays at the same place and looks at fluid motion moving around him.

When we deal with a dynamical system such as (1), the speed of the fluid corresponds to the evolution function $\mathbf{f}(\mathbf{x}(t))$ and the position of a fluid parcel at time t corresponds to the state $\mathbf{x}(t)$. A Lagrangian approach would require simulations to find states that reach the target [20]. In the literature, this method is generally restricted to linear dynamics [1] where a closed form for the flow $\boldsymbol{\varphi}$ is available. It can also be used for nonlinear systems if we use guaranteed

integration [24] [28], but the resulting method is slow. As shown in [12] [17] [9] a Lagrangian method requires many bisections with respect to the time line (for the integration of the state equation), but also on the state space. The Eulerian methods are used for nonlinear systems [19] and try to avoid the integration of the state equation but the corresponding algorithms rely on gridding the state space [22]. To provide guaranteed results, gridding methods require the knowledge of some Lipschitz constant which are rarely available in practice [23]. Lyapunov-based methods [21][5], level-set methods [15], or barrier functions [4] can also be considered as Eulerian since they only check the constraints on the state space and do not need to perform any integration though time. Now, these methods required a parametric expression for candidate Lyapunov-like functions [26].

This paper deal with Eulerian state estimation which can be formalized as follows:

$$\begin{array}{lcl} (i) & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) & \text{(evolution)} \\ (ii) & \mathbf{x}(t_i) \in \mathbb{X}_i & \text{(event)} \\ (iii) & \forall\, (i,j) \in \mathbb{J}, t_i \leq t_j & \text{(precedence)} \end{array} \tag{2}$$

where the t_i are unknown times. Equation (i) corresponds to (1). Constraint *(ii)* tells us that at some unknown times t_i the trajectory has crosses a known set $\mathbb{X}_i$. This set corresponds to an untemporal observation such as: "the robot entered in my house". We deduce this information from the existence of wheelprints, for instance, but we do not know when this event occurred. Constraint (iii) expresses an order between event and is represented by a set $\mathbb{J} \subset \mathbb{N}^2$. For instance, if $(2,5) \in \mathbb{J}$, then the trajectory has crossed $\mathbb{X}_2$ before $\mathbb{X}_5$. Equations (ii) and (iii) can be represented graphically by a graph or a Petri net [18]. Solving such an estimation problem amounts to finding all states that are consistent with one trajectory satisfying (2).

This paper presents an approach based on invariant sets to solve this problem.

2 Invariant sets

This section presents some definitions on invariant sets. We also show that the solution set of several problems involving dynamical systems can be expressed as an algebraic expression involving maximal positive invariant sets.

Positive invariant set. A set $\mathbb{A}$ is *positive invariant* [3] if for any trajectory $\mathbf{x}(\cdot)$ of (1), we have

$$\mathbf{x}(0) \in \mathbb{A}, t \geq 0 \implies \mathbf{x}(t) \in \mathbb{A}. \tag{3}$$

The set of all positive invariant sets is a lattice, *i.e.*, the union and the intersection of two positive invariant sets is positive invariant. A consequence is that, given a set $\mathbb{X}$, the notion of *largest positive invariant set* contained in $\mathbb{X}$ and *smallest positive invariant set* enclosing $\mathbb{X}$ can be defined.

Largest positive invariant set. Given a set $\mathbb{X}$, there exists a largest

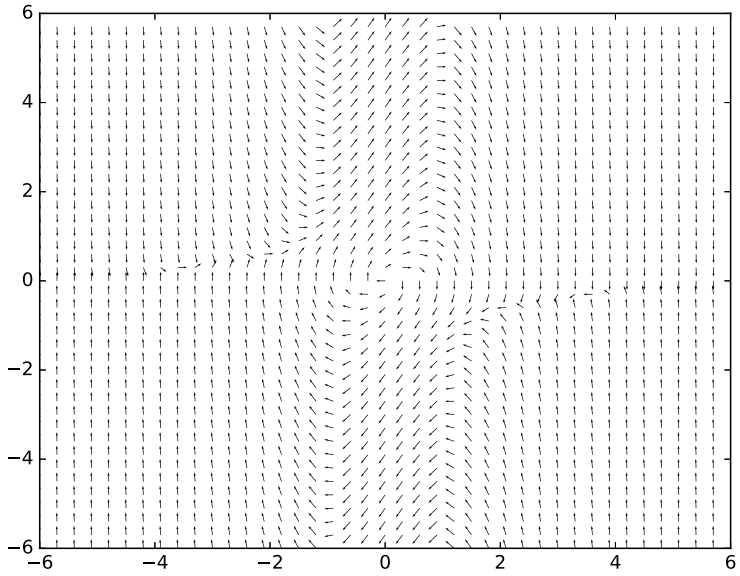


Figure 1: Vector field associated to the Van der Pol system in the box $[-6,6]^{\times 2}$

positive invariant set for (1) included in $\mathbb{X}$. It is given by

$$Inv^{+}(\mathbf{f},\mathbb{X}) = \{\mathbf{x}_0 \mid \forall t \geq 0, \boldsymbol{\varphi}(t,\mathbf{x}_0) \in \mathbb{X}\}\,. \tag{4}$$

In [14][13] a tool has been developed to compute an inner and an outer approximation for $Inv^{+}(\mathbf{f},\mathbb{X})$. As an illustration, consider the system described by the Van der Pol equation:

$$\left\{ \begin{array}{rcl} \dot{x}_1 & = & x_2 \\ \dot{x}_2 & = & \left(1-x_1^2\right)\cdot x_2 - x_1. \end{array} \right. \tag{5}$$

Figure 1 provides an illustration of its vector field.

Figure 2 shows that the largest positive invariant set in $\mathbb{X} = [-6,6]^{\times 2}$ associated to (5). All points in the magenta area will stay inside $\mathbb{X}$ forever whereas all points in the blue zone will leave $\mathbb{X}$.

Largest negative invariant set. It corresponds to the set

$$Inv^{-}(\mathbf{f},\mathbb{X}) = \{\mathbf{x}_0 \mid \forall t \leq 0, \boldsymbol{\varphi}(t,\mathbf{x}_0) \in \mathbb{X}\}\,. \tag{6}$$

Since

$$Inv^{-}(\mathbf{f},\mathbb{X}) = Inv^{+}(-\mathbf{f},\mathbb{X}), \tag{7}$$

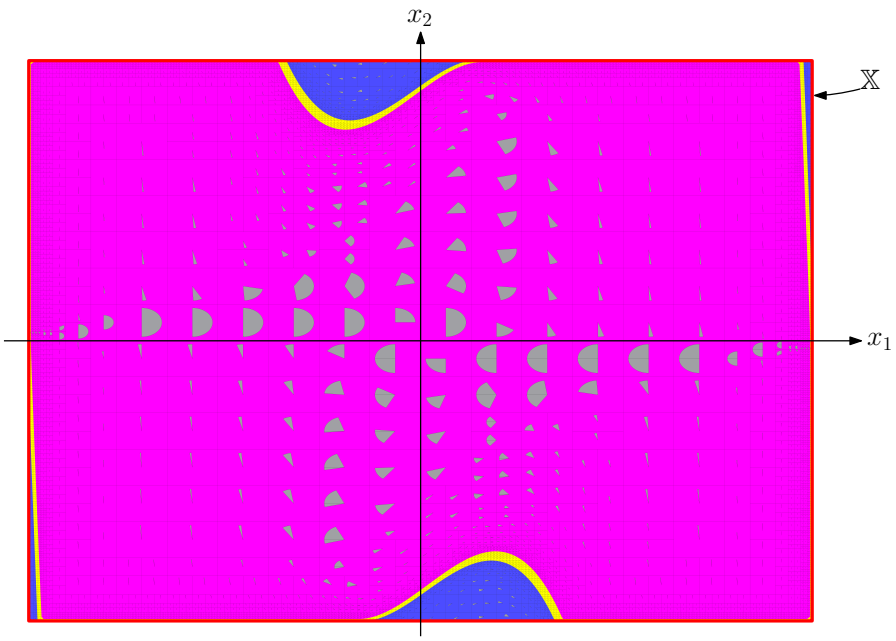


Figure 2: Largest positive invariant set $Inv^{+}(\mathbf{f}, \mathbb{X})$ where $\mathbb{X} = [-6, 6]^{\times 2}$ and $\mathbf{f}$ is the evolution function of the Van der Pol system

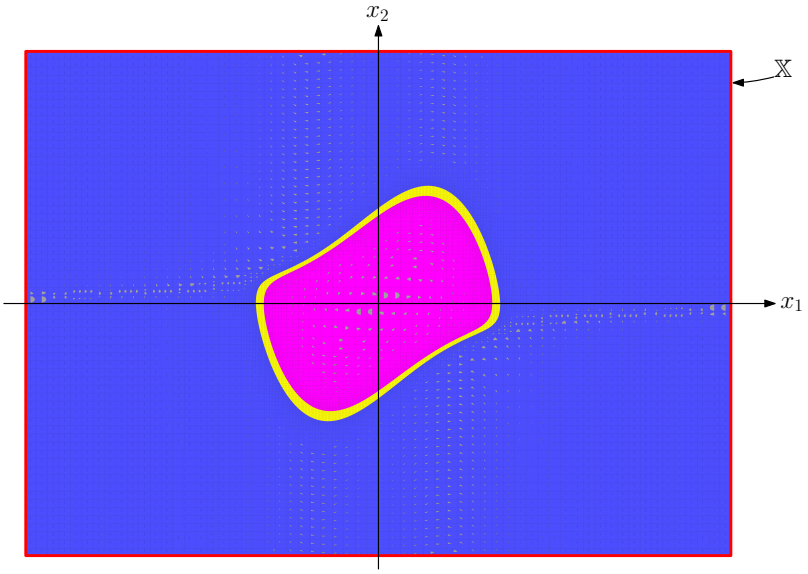


Figure 3: Largest negative invariant set $Inv^-(\mathbf{f}, \mathbb{X})$ where $\mathbb{X} = [-6, 6]^{\times 2}$ corresponding the Van der Pol system

the largest negative invariant set can be defined in terms of positive invariant sets. Figure 3 shows that the largest negative invariant set in $\mathbb{X} = [-6, 6]^{\times 2}$ associated to (5). All points in the magenta area will go to $\mathbb{X}$ in the future whereas all points in the blue zone will never reach $\mathbb{X}$.

Largest invariant set. It corresponds to the set

$$Inv(\mathbf{f}, \mathbb{X}) = \{\mathbf{x}_0 \mid \forall t \in \mathbb{R}, \boldsymbol{\varphi}(t, \mathbf{x}_0) \in \mathbb{X}\}. \tag{8}$$

We have

$$\begin{array}{rcc} Inv(\mathbf{f}, \mathbb{X}) & = & \{\mathbf{x}_0 \mid \forall t \leq 0, \boldsymbol{\varphi}(t, \mathbf{x}_0) \in \mathbb{X} \wedge \forall t \geq 0, \boldsymbol{\varphi}(t, \mathbf{x}_0) \in \mathbb{X}\} \\ & = & \{\mathbf{x}_0 \mid \forall t \leq 0, \boldsymbol{\varphi}(t, \mathbf{x}_0) \in \mathbb{X}\} \cap \{\mathbf{x}_0 \mid \forall t \geq 0, \boldsymbol{\varphi}(t, \mathbf{x}_0) \in \mathbb{X}\} \\ & = & Inv^+(-\mathbf{f}, \mathbb{X}) \cap Inv^+(\mathbf{f}, \mathbb{X}). \end{array} \tag{9}$$

Thus $Inv(\mathbf{f}, \mathbb{X})$ can be defined in terms of largest positive invariant sets. Since, in our Van der Pol example $Inv^-(\mathbf{f}, \mathbb{X}) \subset Inv^+(\mathbf{f}, \mathbb{X})$, the largest invariant set also corresponds to Figure 3.

Forward reach set. It corresponds to the set

$$Forw(\mathbf{f}, \mathbb{X}) = \{\mathbf{x} \mid \exists t \geq 0, \exists \mathbf{x}_0 \in \mathbb{X}, \boldsymbol{\varphi}(t, \mathbf{x}_0) = \mathbf{x}\}. \tag{10}$$

Since

$$\begin{array}{rcc} Forw(\mathbf{f}, \mathbb{X}) & = & \{\mathbf{x} \mid \exists t \geq 0, \exists \mathbf{x}_0 \in \mathbb{X}, \boldsymbol{\varphi}(-t, \mathbf{x}) = \mathbf{x}_0\} \\ & = & \overline{\{\mathbf{x} \mid \exists t \geq 0, \boldsymbol{\varphi}(-t, \mathbf{x}) \in \mathbb{X}\}} \\ & = & \overline{\{\mathbf{x} \mid \forall t \geq 0, \boldsymbol{\varphi}(-t, \mathbf{x}) \in \overline{\mathbb{X}}\}} \\ & = & \overline{Inv^+(-\mathbf{f}, \overline{\mathbb{X}})} \end{array} \tag{11}$$

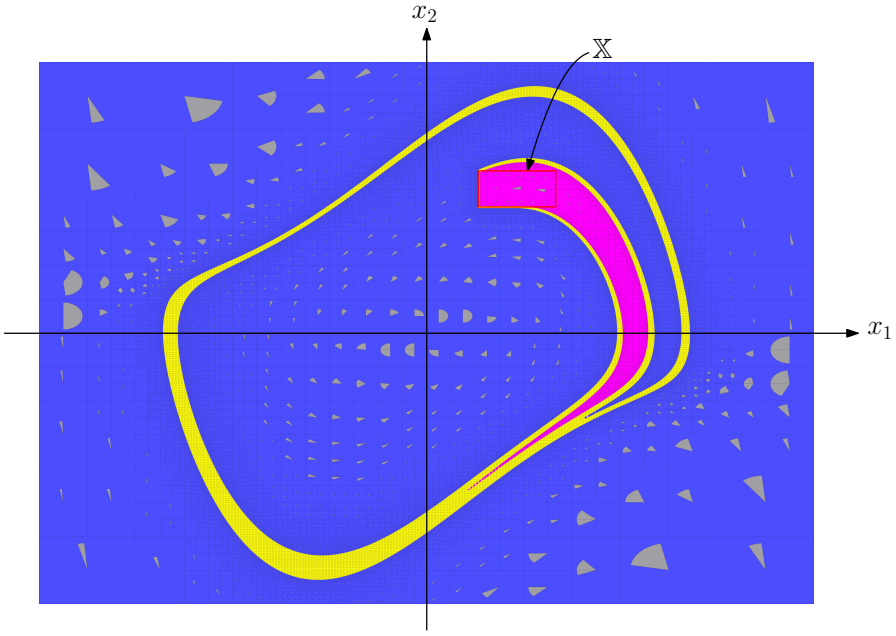


Figure 4: Forward reach set of the Van der Pol system. The frame box is $[-3,3]^{\times 2}$

the set $Forw(\mathbf{f},\mathbb{X})$ can be defined in terms of positive invariant sets. The set $Forw(\mathbf{f},\mathbb{X})$ corresponds to the smallest positive invariant set enclosing $\mathbb{X} = [0.4,1.0]\times[1.4,1.8]$. An illustration is given by Figure 4.

Backward reach set. It corresponds to the set

$$Back(\mathbf{f},\mathbb{X}) = \{\mathbf{x}_0 \mid \exists t \geq 0, \boldsymbol{\varphi}(t,\mathbf{x}_0) \in \mathbb{X}\}\,. \tag{12}$$

Since

$$\begin{array}{rcl} Back(\mathbf{f},\mathbb{X}) & = & Forw(-\mathbf{f},\mathbb{X}) \\ & = & \overline{Inv^{+}(\mathbf{f},\overline{\mathbb{X}})} \end{array} \tag{13}$$

the set $Back(\mathbf{f},\mathbb{X})$ can be defined in terms of positive invariant sets. An illustration is given on Figure 5 for $\mathbb{X} = [0.4,1.0]\times[1.4,1.8]$. All points in the magenta area will reach $\mathbb{X}$ for some $t \geq 0$.

3 Eulerian state estimation

Define ℓ sets $\mathbb{X}_0,\mathbb{X}_1,\ldots,\mathbb{X}_\ell$ of the state space. Define $\mathbb{Z}_k^{forw}$ the set of all state vectors $\mathbf{x}(t)$ inside $\mathbb{X}_k$ that have visited $\mathbb{X}_0,\mathbb{X}_1,\ldots,\mathbb{X}_{k-1}$ in the past (*i.e.*, before time t) and in the specified order. We have

$$\mathbb{Z}_{k+1}^{forw} = Forw\left(\mathbb{Z}_k^{forw}\right) \cap \mathbb{X}_{k+1}, \tag{14}$$

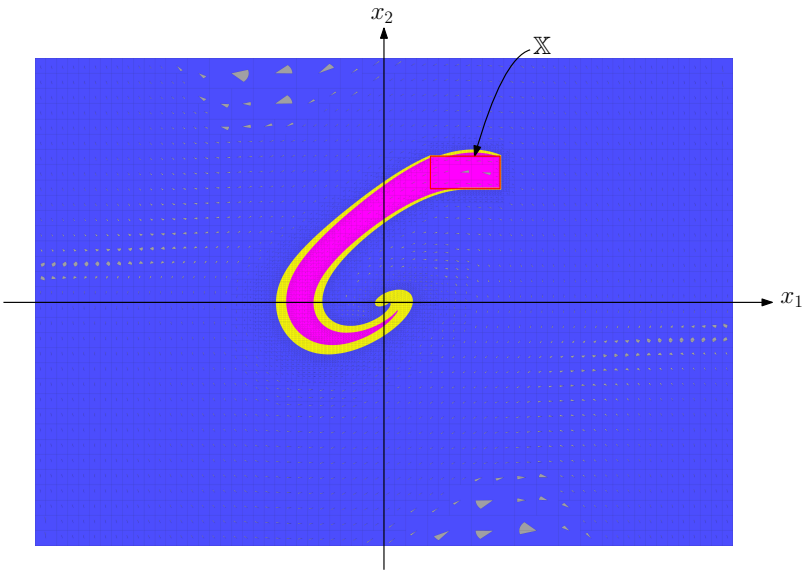


Figure 5: Backward reach set of the Van der Pol system. The frame box is $[-3,3]^{\times 2}$

with $\mathbb{Z}_0^{forw} = \mathbb{X}_0$. This sequence corresponds to what we call the *Eulerian filter*. The principle is illustrated by Figures 6,7 8 and 9. For simplicity, $Forw(\mathbf{f}, \mathbb{X})$ and $Back(\mathbf{f}, \mathbb{X})$ are denoted by $Forw(\mathbb{X})$ and $Back(\mathbb{X})$.

Define the set $\mathbb{Z}_k^{back}$ of all states $\mathbf{x}(t)$ inside $\mathbb{X}_k$ that have visited $\mathbb{X}_0, \mathbb{X}_1, \ldots, \mathbb{X}_{k-1}$ in the past and will visit $\mathbb{X}_{k+1}, \ldots, \mathbb{X}_\ell$ in the future. We have

$$\mathbb{Z}_k^{back} = Back\left(\mathbb{Z}_{k+1}^{back}\right) \cap \mathbb{Z}_k^{forw}, \tag{15}$$

with $\mathbb{Z}_\ell^{back} = \mathbb{Z}_\ell^{forw}$. The will be called the *Eulerian smoother*. The Eulerian smoother is illustrated by Figures 1011

As illustrated by Figure 12, the set of trajectories that started inside $\mathbb{X}_0$ and visited the sets $\mathbb{X}_1, \mathbb{X}_2, \ldots, \mathbb{X}_{\ell-1}$sequentially, and that ended in $\mathbb{X}_\ell$ can thus be enclosed by

$$Forw\left(\mathbb{Z}_0^{back}\right) \cap Back\left(\mathbb{Z}_\ell^{back}\right). \tag{16}$$

Example 1. Define three sets $\mathbb{X}_0, \mathbb{X}_1, \mathbb{X}_2$ and assume that we want to find the set of trajectories that started inside $\mathbb{X}_0$ and visited the set $\mathbb{X}_1$ and then finally reached the set $\mathbb{X}_2$. This problem corresponds to an Eulerian state estimation problem where $\mathbb{J} = \{(0,1),(1,2)\}$, which means that $t_0 \leq t_1 \leq t_2$. We

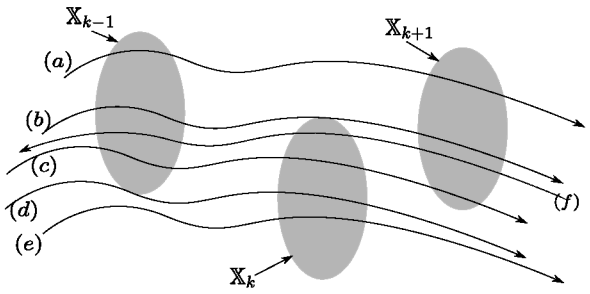


Figure 6: The trajectories (b),(c),(d) are consistent with the sets $\mathbb{X}_{k-1}, \mathbb{X}_k, \mathbb{X}_{k+1}$. The trajectory (f) does not respect the required order.

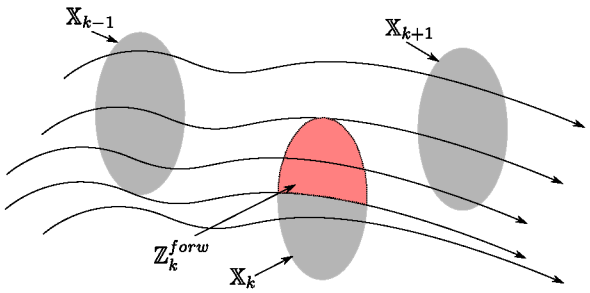


Figure 7: Set $\mathbb{Z}_k^{forw}$ of all states $\mathbf{x}(t)$ in $\mathbb{X}_k$ that have already visited $\mathbb{X}_0, \mathbb{X}_1, \ldots, \mathbb{X}_{k-1}$

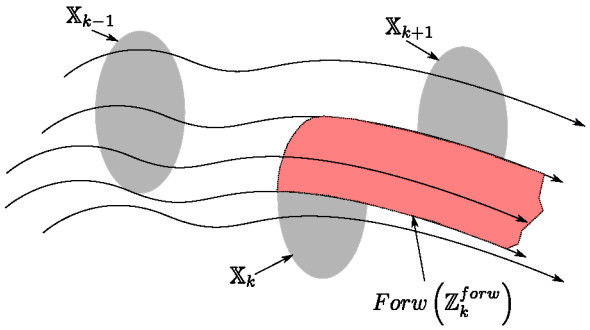


Figure 8: The set $Forw\left(\mathbb{Z}_k^{forw}\right)$ corresponds to all states $\mathbf{x}(t)$ that have visited $\mathbb{X}_0, \mathbb{X}_1, \ldots, \mathbb{X}_k$

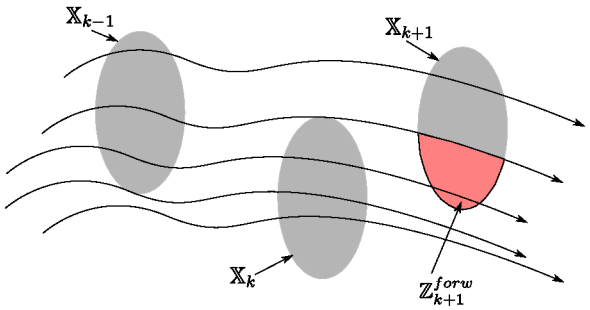


Figure 9: Set $\mathbb{Z}_{k+1}^{forw}$ of all states $\mathbf{x}(t)$ in $\mathbb{X}_{k+1}$ that have already visited $\mathbb{X}_0, \mathbb{X}_1, \ldots, \mathbb{X}_k$

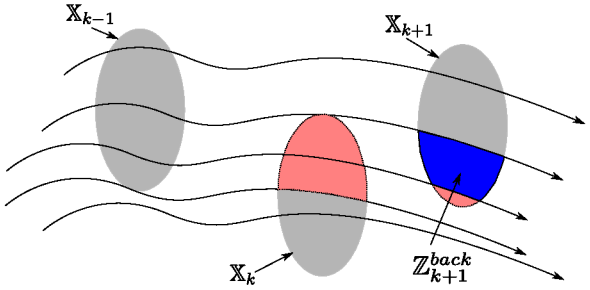


Figure 10: Set $\mathbb{Z}_{k+1}^{back}$ of all states $\mathbf{x}(t)$ inside $\mathbb{Z}_{k+1}^{forw}$ that will visit $\mathbb{X}_{k+2}, \ldots, \mathbb{X}_\ell$

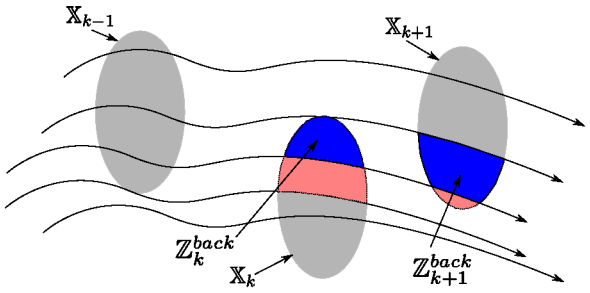


Figure 11: Set $\mathbb{Z}_k^{back}$ of all states $\mathbf{x}(t)$ inside $\mathbb{Z}_k^{forw}$ that will visit $\mathbb{X}_{k+1}, \ldots, \mathbb{X}_\ell$

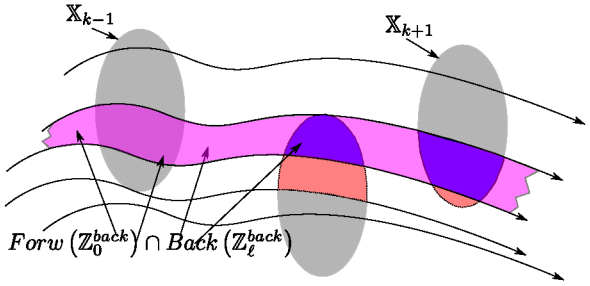


Figure 12: Set $Forw\left(\mathbb{Z}_0^{back}\right) \cap Back\left(\mathbb{Z}_\ell^{back}\right)$ enclosing the trajectory consistent with the past and future visits

have

$$\begin{array}{rcl}
\mathbb{X} & = & Forw\left(\mathbb{Z}_0^{back}\right) \cap Back\left(\mathbb{Z}_2^{back}\right) \\
\mathbb{Z}_0^{forw} & = & \mathbb{X}_0 \\
\mathbb{Z}_1^{forw} & = & Forw\left(\mathbb{Z}_0^{forw}\right) \cap \mathbb{X}_1 \\
\mathbb{Z}_2^{forw} & = & Forw\left(\mathbb{Z}_1^{forw}\right) \cap \mathbb{X}_2 \\
\mathbb{Z}_2^{back} & = & \mathbb{Z}_2^{forw} \\
\mathbb{Z}_1^{back} & = & Back\left(\mathbb{Z}_2^{back}\right) \cap \mathbb{Z}_1^{forw} \\
\mathbb{Z}_0^{back} & = & Back\left(\mathbb{Z}_1^{back}\right) \cap \mathbb{Z}_0^{forw}
\end{array}$$

In the special case where $\mathbb{X}_0 = [\mathbf{a}] = [0, 0.6] \times [0.8, 1.8]$,$\mathbb{X}_1 = [\mathbf{b}] = [0.7, 1.5] \times [-0.2, 0.2]$ and $\mathbb{X}_2 = [\mathbf{c}] = [0.2, 0.6] \times [-2.2, -1.5]$, we get the set depicted on Figure 13.

An application of Eulerian state estimation is illustrated by Figure 14, where ocean currents are represented. The objective of the mission is to find a trajectory for a buoy that follows the currents in order to visit the three red boxes. Since the time is not taken into account in the estimation process, we are in an Eulerian context.

References

[1] E. Asarin, T. Dang, and A. Girard. Reachability analysis of nonlinear systems using conservative approximation. In *Hybrid Systems: Computation and Control*, pages 20–35. Springer Berlin Heidelberg, 2003.

[2] J.P. Aubin and H. Frankowska. *Set-Valued Analysis*. Birkhäuser, Boston, 1990.

[3] Franco Blanchini and Stefano Miani. *Set-Theoretic Methods in Control*. Springer Science & Business Media, October 2007.

[4] O. Bouissou, A. Chapoutot, A. Djaballah, and M. Kieffer. Computation of parametric barrier functions for dynamical systems using interval analysis.

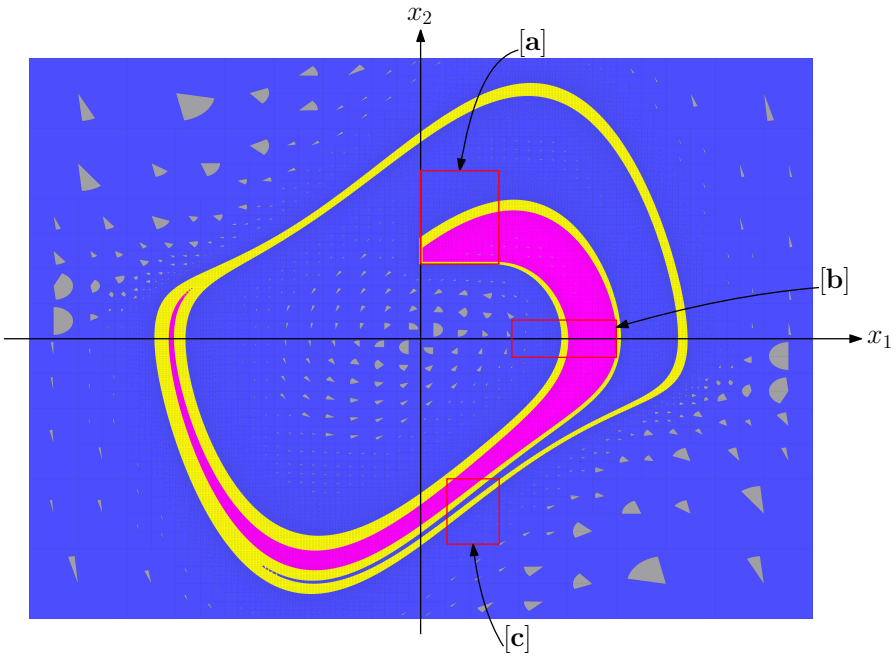


Figure 13: Feasible states associated to the Eulerian state estimation problem

In *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*, pages 753–758, December 2014.

[5] N. Delanoue, L. Jaulin, and B. Cottenceau. Attraction domain of a nonlinear system using interval analysis. In *Twelfth International Conference on Principles and Practice of Constraint Programming (IntCP 2006)*, France, Nantes, 2006.

[6] A. Desilles, H. Zidani, and E. Cruck. Collision analysis for an UAV. In *AIAA Guidance, Navigation, and Control Conference*, pages 13–16. American Institute of Aeronautics and Astronautics, 2012.

[7] L. Jaulin. *Automation for Robotics*. ISTE editions, 2015.

[8] L. Jaulin. *Mobile Robotics*. ISTE editions, 2015.

[9] L. Jaulin, D. Lopez, Le Doze, S. Le Menec, J. Ninin, G. Chabert, S. Ibn Seddik, and A. Stancu. Computing capture tubes. In M. Nehmeier, J.W. von Gudenberg, and W. Tucker, editors, *Scientific Computing, Computer Arithmetic, and Validated Numerics*, volume 9553, pages 209–224. Springer, 2016.

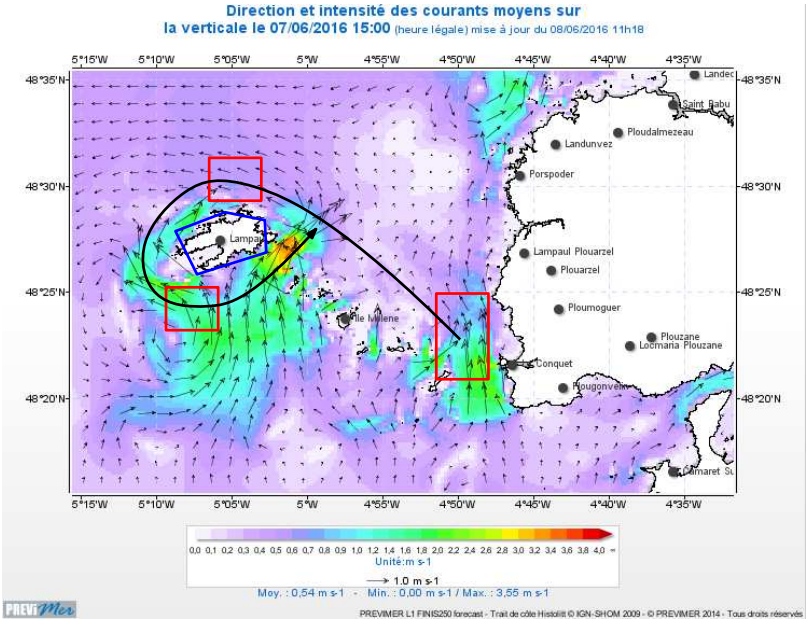


Figure 14: Visiting the three red boxes using a buoy that follows the currents is an Eulerian state estimation problem

[10] M. Konecny, W. Taha, J. Duracz, A. Duracz, and A. Ames. Enclosing the behavior of a hybrid system up to and beyond a zeno point. In *Cyber-Physical Systems, Networks, and Applications (CPSNA)*, 2013.

[11] L.D. Landau and E.M. Lifshitz. *Fluid Mechanics, Course of Theoretical Physics*. Butterworth-Heinemann, 1987.

[12] M. Lhommeau, L. Jaulin, and L. Hardouin. Capture Basin Approximation using Interval Analysis. *International Journal of Adaptative Control and Signal Processing*, 25(3):264–272, 2011.

[13] T. Le Mézo, L. Jaulin, and B. Zerr. Inner approximation of a capture basin of a dynamical system. In *Abstracts of the 9th Summer Workshop on Interval Methods*. Lyon, France, June 19-22, 2016.

[14] T. Le Mézo, L. Jaulin, and B. Zerr. An interval approach to compute invariant sets. *IEEE Transaction on Automatic Control*, 2017.

[15] Ian Mitchell, Alexandre M. Bayen, and Claire J. Tomlin. Validating a Hamilton-Jacobi Approximation to Hybrid System Reachable Sets. In Maria Domenica Di Benedetto and Alberto Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, number 2034 in Lecture Notes in Computer Science, pages 418–432. Springer Berlin Heidelberg, 2001.

[16] I.M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. In A. Bemporad, A. Bicchi, and G. Buttazzo, editors, *Hybrid Systems: Computation and Control*, pages 428–443. Springer-Verlag, 2007.

[17] D. Monnet, J. Ninin, and L. Jaulin. Computing an inner and an outer approximation of the viability kernel. *Reliable Computing*, 2016.

[18] T. Murata. Petri nets : properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[19] M. Quincampoix. Differential inclusions and target problems. *SIAM journal on control and optimization*, 30(2):324–335, 1992.

[20] N. Ramdani and N. Nedialkov. Computing Reachable Sets for Uncertain Nonlinear Hybrid Systems using Interval Constraint Propagation Techniques. *Nonlinear Analysis: Hybrid Systems*, 5(2):149–162, 2011.

[21] S. Ratschan and Z. She. Providing a Basin of Attraction to a Target Region of Polynomial Systems by Computation of Lyapunov-like Functions . *SIAM J. Control and Optimization*, 48(7):4377–4394, 2010.

[22] P. Saint-Pierre. Hybrid kernels and capture basins for impulse constrained systems. In C.J. Tomlin and M.R. Greenstreet, editors, *in Hybrid Systems: Computation and Control*, volume 2289, pages 378–392. Springer-Verlag, 2002.

[23] Patrick Saint-Pierre. Approximation of the viability kernel. *Applied Mathematics and Optimization*, 29(2):187–209, March 1994.

[24] J. Alexandre Dit Sandretto and A. Chapoutot. Validated simulation of differential algebraic equations with runge-kutta methods. *Reliable Computing*, 22, 2016.

[25] N. Seube, R. Moitie, and G. Leitmann. Aircraft Take-Off in Windshear: A Viability Approach. *Set-Valued Analysis*, 8(1-2):163–180, June 2000.

[26] Zhikun She and Bai Xue. Computing an invariance kernel with target by computing lyapunov-like functions. *IET Control Theory Applications*, 7(15):1932–1940, October 2013.

[27] W. Taha and A. Duracz. Acumen: An open-source testbed for cyber-physical systems research. In *CYCLONE'15*, 2015.

[28] D. Wilczak and P. Zgliczynski. Cr-lohner algorithm. *Schedae Informaticae*, 20:9–46, 2011.

The Box Regularized Particle Filter: A probabilistic set-membership observer

Nicolas Merlinge[1,2], Karim Dahia[2], Hélène Piet-Lahanier[2], James Brusey[1], Nadjim Horri[1]

[1] Coventry University, Faculty of Engineering and Computing,
Priory Street, Coventry, United Kingdom
name.surname@coventry.ac.uk
[2] ONERA - The French aerospace lab
Chemin de la Hunière BP 80100 FR-91123 Palaiseau, France
name.surname@onera.fr

Keywords: Box Particle Filtering, Interval Analysis, Kernel Estimation, Bayesian filtering

Introduction

State estimation of dynamic systems requires to model the uncertainty on the initial state values, disturbances on the dynamics and errors on the measurements. In probabilistic approaches [1], the initial state is modelled as a random vector and the disturbances and errors as stochastic processes with a priori pdf. The resulting state estimation is thus associated with a posterior probabilistic density. In set-membership approaches [2], the uncertainty is accounted for by considering that the values are unknown but belong to predefined sets in suitable state space, often described as bounds on the component values. Estimation in this context aims at defining the membership set where the state is guaranteed to belong. Definition of the bounds is usually simpler than determining a priori pdf but the determination of the guaranteed estimated state space boundaries may prove tedious for non-linear dynamics or observation models.

As the exact determination of the membership set for non linear models is difficult to obtain on-line, various approximation techniques have been suggested. Among them, Box Particle Filter (BPF, [3])

consists in a combination of Particle Filtering which is efficient for non linear estimation and Interval Analysis techniques which allow propagation of bounds. Instead of drawing a stochastic cloud of weighted possible states, as a classic particle filter does, the initial state set is represented as the unions of intervals resulting from a regular paving. Each box of the paving is then dynamically propagated. The measurement box is a n-dimensional box centered on the vector of measurements. Consistency between the measurement box and each predicted observation box results in a potential contraction of each box of the paving. A weight linked with the consistency is associated to each contracted box and represents its likelihood to contain the actual state.

BPF proves to be robust to dynamics and observation non-linearity while remaining computationally efficient for on-line estimation in embedded contexts. However, it can prove pessimistic and inaccurate when used with ambiguous measurements (i.e. when one measurement value can be related to several state values). Therefore, the authors have proposed a refined version of the BPF called Box Regularized Particle Filter [4].

An accurate posterior pdf evaluation from the Box Particles set

The BPF aims to estimate a bounded set guaranteed to contain the actual state to be estimated, defined as the union of the $N \in \mathbb{N}^*$ boxes. In addition, it provides an estimated posterior density function $p(\mathbf{X}_k|\mathbf{M}_k)$ of state $\mathbf{X}_k$ given measurements from time 1 to time k $\mathbf{M}_k = \{\mathbf{m}_1, ..., \mathbf{m}_k\}$. It is defined as:

$$p(\mathbf{X}_k|\mathbf{M}_k) = \sum_{i=1}^{N} w_k^i \mathcal{U}_{[\mathbf{X}_k^i]}$$

where $\mathcal{U}_{[\mathbf{X}_k^i]}$ refers to a uniform pdf whose support is the i^{th} box particle $[\mathbf{X}_k^i]$ at time k and w_k^i refers to the associated likelihood of each box to contain the actual state given the measurements. The consistency of each particle with measurement $\mathbf{m}_k$ is evaluated through its weight

w_k^i. A complete probabilistic description of the BPF can be found in [3].

In order to keep a consistent set of Box Particles, a resampling step makes it possible to replace inconsistent Box Particles with subdivisions of consistent ones. The BPF can then iteratively converge to a set containing the actual state $\mathbf{X}_k$. However, the resulting evaluation of the posterior pdf $p(\mathbf{X}_k|\mathbf{M}_k)$ is rough. In [4], the authors have derived a Kernel smoothing technique inspired from Kernel Estimation (see [5], [6]) to refine the pdf evaluation. To this end, at each resampling step, the posterior pdf is rewritten as:

$$p(\mathbf{X}_k|\mathbf{M}_k) \approx \sum_{i=1}^{N} w_k^i K_h(\mathbf{X}_k - \mathbf{C}_k^i)$$

where K_h is a bounded smooth kernel used to shift each Box Particle support around its center $\mathbf{C}_k^i$. More details about the so-called Box Regularized Particle Filter (BRPF) can be found in [4].

Main results

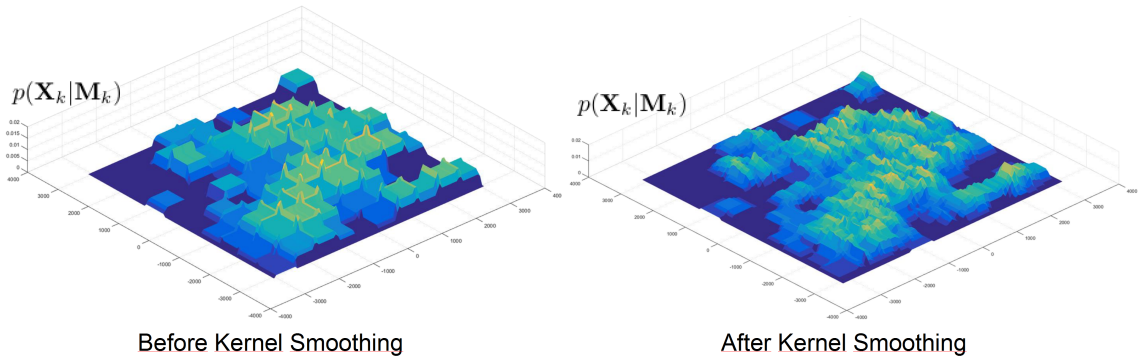


Figure 1: Kernel smoothing leads to reduce jaggedness of the posterior density

The BRPF algorithm has been applied to terrain navigation and proved far more accurate than the initial BPF. It has also been implemented on an FPGA chip and demonstrated its ability to tackle

real-time requirements on low computational ability devices. Figure 1 shows the posterior pdf before and after Kernel smoothing.

An efficient way to define a state estimate is computing the empirical expectancy of the posterior pdf: $\widehat{\mathbf{X}}_k = \mathbb{E}\left[p(\mathbf{X}_k|\mathbf{M}_k)\right]$. In order to evaluate the accuracy gain of the BRPF, a Root Mean Square Error (RMSE) over a hundred Monte-Carlo runs has been computed. The RMSE is defined by:

$$RMSE_k = \sqrt{\frac{1}{N_{MC}} \sum_{j=1}^{N_{MC}} \|\hat{\mathbf{x}}_k - \mathbf{x}_k\|^2},$$

where N_{MC} is the number of runs and $k \in \mathbb{N}$ the time-step index (each run is denoted by index $j \in \mathbb{N}$). The BRPF results in a significantly lower RMSE than the BPF. More details about the problem configuration can be found in [4].

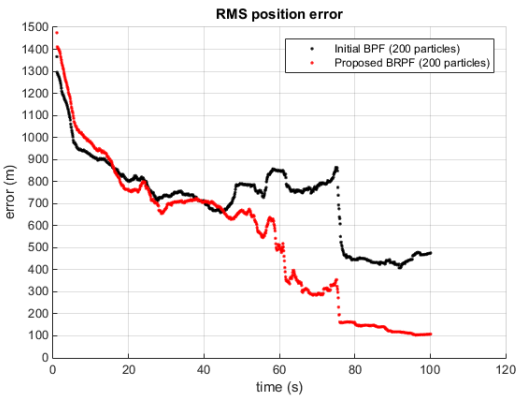


Figure 2: Localization error (m) for an aircraft performing terrain navigation. Black: BPF, Red: BRPF.

References

[1] B. Ristic, *Beyond the Kalman filter: Particle filters for tracking applications*. Boston: Artech House, 2004.

[2] L. Jaulin, "Robust set-membership state estimation," *Automatica*, vol. 45, pp. 202—-206, 2009.

[3] A. Gning, B. Ristic, L. Mihaylova, and F. Abdallah, "An introduction to box particle filtering," *IEEE Signal Processing Magazine*, pp. 166–171, 2013.

[4] N. Merlinge, K. Dahia, and H. Piet-Lahanier, "A box regularized particle filter for terrain navigation with highly non-linear measurements," *Proceeding of the IFAC ACA Conference*, vol. 49, no. 17, pp. 361–366, 2016.

[5] B. Silverman, *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall, 1986.

[6] C. Musso, N. Oudjane, and F. LeGland, "Improving regularized particle filters," *Sequential Monte Carlo Methods in Practice*, vol. 12, pp. 247–271, 2001.

Extended Quantified Set Inversion Algorithm with Applications to Control

Pau Herrero[1] and Miguel A. Sainz [2]

[1] Centre for Bio-inspired Technology, Department of Electrical and Electronic Engineering,
Imperial College, London, United Kingdom
pherrero@imperialac.uk
[2] Institut d'Informàtica i Aplicacions, Universitat de Girona,
Girona, Spain,
sainz@ima.udg.edu

Keywords: Set inversion, Constraint Satisfaction, Quantifiers, Modal Interval Analysis, Control Engineering

Introduction

The Quantified Set Inversion (QSI) algorithm is a set inversion algorithm based on Modal Interval Analysis (MIA) that allows the characterization of AE-solution set of a parametric non-linear system - i.e, quantified real constraint (QRC) problem. However, the original QSI algorithm is limited to the resolution of a subset of QRC problems where existentially quantified variables are not shared between equality constraints. This work presents an extended version of the QSI algorithm that overcomes some of these limitations. In addition, a user-friendly Matlab toolbox including a MIA arithmetic; an efficient implementation of an algorithm for performing efficient MIA computations (f^* algorithm); and the QSI algorithm is introduced. Due to the high popularity of Matlab in the scientific and engineering communities, the presented toolbox is expected to promote the use of MIA. Finally, several examples of utilization of the Matlab toolbox and an application to control engineering are presented.

Localization for Group of Robots using Matrix Contractors

Nisha Rani Mahato [1,2], Luc Jaulin [2] and Snehashish Chakaverty[1]

[1] Department of Mathematics,
National Institute of Technology Rourkela,
Odisha-769008, India
[2] ENSTA-Bretagne, LabSTICC, CNRS 6285,
2 rue François Verny, 29806 Brest, France
nisha.mahato1@gmail.com,luc.jaulin@ensta-bretagne.fr,chakravertys@nitrkl.ac.in

Keywords: Angle constraints, Matrix contractors, Azimuth angle, Distance, Localization

Introduction

Interval analysis is a tool that can be used to propagate interval uncertainties. To each variable x of $\mathbb{R}$ is associated a closed interval $[x] = [\underline{x}, \overline{x}] \in \mathbb{IR}$ such that $[\underline{x}, \overline{x}] = \{x | \underline{x} \leq x \leq \overline{x}, \underline{x}, \overline{x} \in \mathbb{R}\}$, where $\underline{x} = inf([x])$ and $\overline{x} = sup([x])$ may be referred as the infimum and supremum. Interval methods can be extended to any type of lattice $(\mathbb{L}, \leq)$ which is a partially ordered set. This is the case of the set of matrices which is a lattice, where the order relation has to be understood componentwise. An interval matrix $[\mathbf{A}] \in \mathbb{IR}^{m \times n}$ forms a sub-lattice that may be written as $\left([a]_{ij}\right)$, where $i = 1, \cdots, m$ and $j = 1, \cdots, n$.

In the literature, the localization problem is classically formalized as a Constraint Satisfaction Problem (CSP) (Jaulin et al. [1]; Araya [2]). A CSP consists of set of variables $\mathcal{V} = \{x_1, \ldots, x_n\}$ along with set of constraints or equations $\mathcal{E} = \{c_1, \ldots, c_n\}$ over interval domains$\{[x_1], \ldots, [x_n]\}$.

When dealing with groups of robots (or swarm) the localization involves the estimation of the pose of a group of robots $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n$ that are linked by a set of constraints. In this case, the constraints that

are involved can be represented by matrix equations. The ith row and the jth column of a given matrix variable represents a link between the robots $\mathcal{R}_i$ and $\mathcal{R}_j$. In this context, the variable are not anymore scalar, but is a matrix. As such, the constraints involve matrix operations such as addition, multiplication, inverse, determinant, trace, eigenvalues etc.

This paper mainly deals with construction of matrix contractors with respect to constraints that are met in robot localization. More specifically, we will consider azimuth and distance matrices that are generally involved. Section 2 introduces the notion of matrix contractors. Section 3 deals with the localization of group of robots using matrix contractors. A test-case will be presented to evaluate the efficiency of the proposed contractors.

Matrix contractors

Definition 1 (Contractor)Chabert and Jaulin [3]. A (classical) contractor associated with a set $\mathbb{X} \subset \mathbb{R}^n$ is a lattice order preserving operator

$$\mathcal{C} : \mathbb{IR}^n \to \mathbb{IR}^n$$

where, $\mathbb{IR}^n$ be the set of intervals of $\mathbb{R}^n$ (or boxes for short). The following should hold for all boxes $[\mathbf{x}] \in \mathbb{IR}^n$:

- Contraction: $\mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}]$,
- Completeness: $\mathcal{C}([\mathbf{x}]) \cap \mathbb{X} = [\mathbf{x}] \cap \mathbb{X}$

Often, the set $\mathbb{X}$ is a set which corresponds to some equations or constraints. According to Knaster–Tarski theorem (Tarski [4]), if $\mathcal{C}$ is monotonic, then $\mathcal{C} \circ \mathcal{C} \circ \cdots \circ \mathcal{C}([\mathbf{x}])$ will converge to the largest fixed point $[\mathbf{a}]$ of $\mathcal{C}$ included in $[\mathbf{x}]$. In the case where the components of $\mathbf{x}$ correspond to components of some matrices then the contractor will be called a *matrix contractor*.

Example 1. Consider the constraint "$\mathbf{S}$ is a symmetric matrix" where $\mathbf{S}$ is a $n \times n$ matrix. Define $\mathbb{S}$ the set of all $n \times n$ symmetric matrices. The minimal contractor $\mathcal{C}_{sym}$ associated with $\mathbb{S}$

$$\mathcal{C}_{sym}\left([\mathbf{S}]\right) = [\mathbf{S}] \cap [\mathbf{S}]^{\mathrm{T}}.$$

where $[\mathbf{S}] \in \mathbb{IR}^{n \times n}$. For instance

$$\mathcal{C}_{sym} \begin{pmatrix} [1,3] & [2,4] \\ [-1,3] & [-1,1] \end{pmatrix} = \begin{pmatrix} [1,3] & [2,3] \\ [2,3] & [-1,1] \end{pmatrix}$$

Example 2. Consider the matrix constraint "$\mathbf{A} + \mathbf{B} = \mathbf{C}$" where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are $n \times n$ matrices. Define $\mathbb{S}$ the set of all $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ such that $\mathbf{A} + \mathbf{B} = \mathbf{C}$. The minimal contractor $\mathcal{C}_{plus}$ associated with $\mathbb{S}$

$$\mathcal{C}_{plus} \begin{pmatrix} [\mathbf{A}] \\ [\mathbf{B}] \\ [\mathbf{C}] \end{pmatrix} = \begin{pmatrix} [\mathbf{C}] - [\mathbf{B}] \\ [\mathbf{C}] - [\mathbf{A}] \\ [\mathbf{A}] + [\mathbf{B}] \end{pmatrix}.$$

The minimality of the contractor is a consequence of the fact the constraint "$\mathbf{A} + \mathbf{B} = \mathbf{C}$" can be decomposed into a scalar form and that no dependencies exist between the entries of the matrices in the scalar decomposition. A minimal contractor $\mathcal{C}_{\mathcal{P}}$ associated with the constraint "$\mathbf{P} \in \mathbb{R}^{m \times m}$ is a positive semi-definite matrix" has been proposed in Jaulin and Henrion [5] based on the convexity of the constraint. Further, an eigenvalue contractor $\mathcal{C}_{\lambda}$ associated with the constraint "$\mathbf{S} \in \mathbb{R}^{n \times n}$ is symmetric with eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$" has been proposed in Hladík and Jaulin [6].

Localization of group of robots

This section presents the localization of group of robots subject to matrix constraints in terms of azimuth angles along with distances between the robots. Generally, localization consists of finding the pose of some robots $(x_i, y_i, \theta_i)^T, i \in \{1, 2, \ldots\}$ as illustrated in Figure 1, where (x_i, y_i) is the position of the ith robot and θ_i is its orientation.

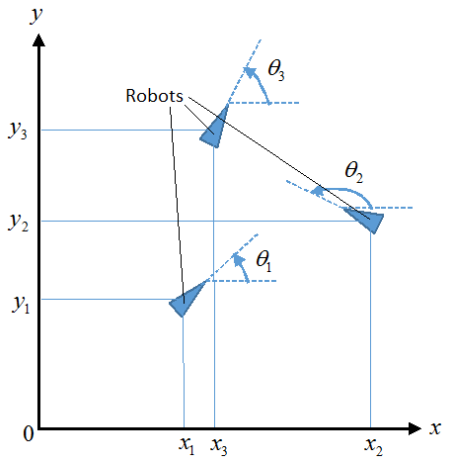


Figure 1: Pose of three robots on a planar section

Localization may be absolute or relative. The absolute localization deals with estimation of instantaneous poses $(x_i, y_i, \theta_i)^T$ of the robots. In case of group of underwater robots, the absolute localization is not possible due to the lack of absolute landmarks. As such, the localization should be relative. Then, the geometrical parameters viz. azimuth and distance may be used to represent the geometry of the group. No absolute frame nor a fixed robot is needed in this representation.

Azimuth matrix. An azimuth matrix associated with n robots is the matrix of azimuth angles α_{ij} between i^{th} robot and j^{th} robot for $i, j = 1, 2, \ldots, n$. It can be written as

$$\mathbf{A} = \begin{pmatrix} 0 & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \alpha_{1\,(n-1)} \\ \alpha_{n1} & \cdots & \alpha_{(n-1)\,1} & 0 \end{pmatrix}$$

By convention, we have taken that $\alpha_{ii} = 0$. The azimuth angles are expressed with respect to the North as depicted in Figure 2.

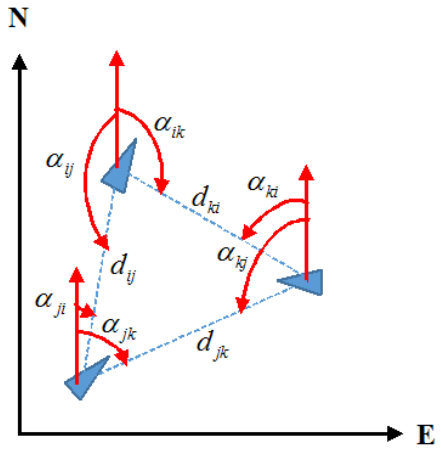


Figure 2: Azimuth angles for three robots on a planar section

In practice, the north is obtained using a compass which is uncertain and all angles are collected via goniometric sensors (like cameras or microphones) are not accurate. This uncertainty can be represented under the form of an interval matrix.

$$[\mathbf{A}] = \begin{pmatrix} 0 & [\alpha_{12}] & \cdots & [\alpha_{1n}] \\ [\alpha_{21}] & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & [\alpha_{1\,(n-1)}] \\ [\alpha_{n1}] & \cdots & [\alpha_{(n-1)\,1}] & 0 \end{pmatrix}.$$

We define the following constraint

$$\alpha \overset{2\pi}{=} \beta \Leftrightarrow \cos(\alpha - \beta) = 1$$

Using the IBEX, an optimal contractor for this relation can easily be built.

To build an azimuth contractor $\mathcal{C}_{az}([\mathbf{A}])$ consistent with the constraint "$\mathbf{A}$ is an azimuth matrix", we consider the following implication

$$azimuth\,(A) \Rightarrow \begin{cases} \alpha_{ij} - \alpha_{ji} \overset{2\pi}{=} \pi & \forall i, j \; i \neq j \\ (\alpha_{ij} - \alpha_{ik}) + (\alpha_{ki} - \alpha_{kj}) + (\alpha_{jk} - \alpha_{ji}) \overset{2\pi}{=} \pi & \forall i, j, k \; i \neq j \neq k \\ (\alpha_{ij} - \alpha_{jk}) \overset{2\pi}{=} (\alpha_{ik} - \alpha_{ij}) + (\alpha_{kj} - \alpha_{ki}) & \forall i, j, k \; i \neq j \neq k \end{cases}$$

Now, the distance matrix associated with n robots is given by $\mathbf{D} = ([d]_{ij})$ for $i, j = 1, 2, \ldots, n$. To build distance contractor consistent with the constraint "$\mathbf{D}$ is a distance matrix", we consider the following:

$$distance(\mathbf{D}) \Rightarrow \begin{cases} d_{ij} = d_{ji} & \forall i, j \\ d_{ij} \leq d_{ik} + d_{kj} & \forall i, j, k \end{cases}$$

Moreover, we may build the following mixed constraint

$$distance{-}azimuth(\mathbf{D}, \mathbf{A}) \Rightarrow \sin(\alpha_{ik} - \alpha_{ij}) d_{ij} = \sin(\alpha_{ki} - \alpha_{kj}) d_{kj} \; \forall i, j, k \; i \neq j \neq k$$

Example 5. Let us consider the azimuth matrix with respect to five robots:

$$[\mathbf{A}] = \begin{pmatrix} 0 & [2, 2.1] & [-2.8, 3] & [2.5, 3.3] & -[2.2, 2.3] \\ [5.2, 5.3] & 0 & -[2, 2.1] & [3.1, 3.2] & -[1.5, 1.6] \\ [0.1, 0.2] & [1, 1.1] & 0 & -[2.6, 2.7] & -[1.4, 1.5] \\ [5.5, 6.5] & [6.2, 6.3] & [0.5, 0.6] & 0 & -[1, 1.4] \\ [0.8, 0.9] & [1.5, 1.6] & [1.7, 1.8] & [1.7, 2.1] & 0 \end{pmatrix}$$

The contracted azimuth matrix using matrix contractors up to the fixed point is

$$\begin{pmatrix} 0 & [2.05, 2.1] & -[2.94, 3] & [2.5, 3.3] & -[2.24, 2.3] \\ [5.2, 5.24] & 0 & -[2.1, 2.04] & [3.1, 3.15] & -[1.54, 1.6] \\ [0.14, 0.2] & [1.04, 1.1] & 0 & -[2.6, 2.64] & -[1.4, 1.44] \\ [5.6, 6.4] & [6.2, 6.3] & [0.5, 0.54] & 0 & -[1.04, 1.45] \\ [0.84, 0.9] & [1.54, 1.6] & [1.7, 1.74] & [1.74, 2.1] & 0 \end{pmatrix}$$

Acknowledgement

The first author is thankful for the support by Raman-Charpak Fellowship 2016, Indo-French Center for the Promotion of Advanced Research, New Delhi, India to carry out the present research. The authors are also thankful to the Science and Engineering Research Board, Department of Science and Technology, New Delhi, India and NIT Rourkela for the support to continue the Ph. D. studies of the first author.

References

[1] L. Jaulin, M. Kieffer, O. Didrit and E. Walter, *Applied Interval Analysis: With Examples in Parameter and State Estimation, Roust Control and Robotics*, Springer, London, 2001.

[2] I. Araya, B. Neveu and G. Trombettoni, Exploiting common subexpressions in numerical CSPs. *In International Conference on Principles and Practice of Constraint Programming*, 342–357, 2008.

[3] G. Chabert and L. Jaulin, Computing the pessimism of inclusion functions, *Reliable Computing*, 13, 489–504, 2007.

[4] A. Tarski, A lattice-theoretical fixpoint theorem and its applications, *Pacific Journal of Mathematics*, 5, 285–309, 1955.

[5] L. Jaulin and D. Henrion, Contracting optimally an interval matrix without loosing any positive semi-definite matrix is a tractable problem, *Reliable Computing*, 11, 1–17, 2005.

[6] M. Hladík and L. Jaulin, An eigenvalue symmetric matrix contractor. *Reliable Computing*, 16, 27–37, 2011.

Bandwidth efficient concurrent ranging and communication for localisation in underwater acoustic networks

Jan Śliwka[1], Andrea Munafò [2], Roberto Petroccia [1]

[1] NATO STO-Centre for Maritime Research and Experimentation (CMRE), Viale San Bartolomeo 400, La Spezia (SP) 19126, Italy.
{jan.sliwka,roberto.petroccia}@cmre.nato.int
[2] Marine Autonomous & Robotic Systems (MARS), National Oceanography Centre, European Way, Southampton, SO14 3ZH, UK.
andmun@noc.ac.uk

Keywords: Localisation, Navigation, Ranging, Low Bandwidth, Interval Methods, Underwater Communications, AUV

Introduction

Underwater navigation is essential for Autonomous Underwater Vehicles (AUVs) to perform their mission. In some scenarios, the AUVs are part of a larger system, namely an acoustic underwater network, where the cooperation of multiple nodes and vehicles can be exploited to enhance the localisation capability of each node. In this context, acoustic messages can be effectively exchanged in the network to obtain range measurements [1, 2]. However, acoustic communication is low bandwidth [3](as low as tens or hundreds bits per second). Reducing the overhead introduced by the ranging strategy is therefore really important. This paper investigates an original method to efficiently encode localisation specific messages in a controlled way which can be adapted to the mission profile. The proposed method enables therefore to obtain a good compromise between the required bandwidth and localisation accuracy.

Ranging

In general, measuring the distance between two nodes breaks down to measuring the sound velocity in water (c_s) and the time-of-flight (TOF) of the acoustic messages. Figure 1 shows a two-way message exchange between node i and node j. The absolute times of emission and reception of the messages are denoted $t_1, .., t_4$. In the example,

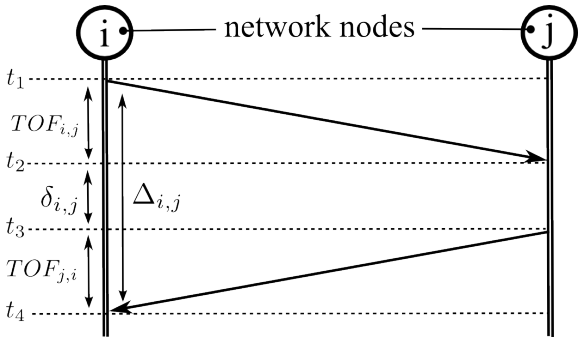


Figure 1: Measuring the two-way TOF between nodes i and j

in order for Node i to compute the TOF, Node j has to include in its response either the timings t_2, t_3 or their relative difference $\delta_{i,j} = t_3 - t_2$. In this paper we assume that $\delta_{i,j}$ is being transmitted. The TOF can be then computed at Node i as $(t_4 - t_1 - \delta_{i,j})/2$. Finally, the range is obtained using this formula:

$$r_{i,j} = c_s \frac{(t_4 - t_1 - \delta_{i,j})}{2}. \tag{1}$$

Reducing the overhead: encoding the delta

The objective of this paper is to effectively reduce the amount of bits used to encode delta while maintaining a sufficient localisation accuracy, according to the mission profile. We denote by $h_{i,j}$ the encoded time value to be sent, u the selected unit of time and n the number of bits used to encode the delta. The value $h_{i,j}$ is obtained using the following formula:

$$h_{i,j} = [\frac{\delta_{i,j}}{u} + \frac{1}{2}] \pmod{2^n}, \tag{2}$$

where $[x]$ is the integer value of the variable x. On the receiving side, the ranging equation 1 can be rewritten as

$$\exists p \in \mathbb{N}, \exists w \in [\frac{-1}{2}, \frac{1}{2}], r_{i,j} = c_s \frac{(t_4 - t_1 - h_{i,j} * u - p * u * 2^n - u * w)}{2}, \tag{3}$$

where the quantization error w and the modulo quotient p are unknown to the receiver. Because of the quantisation, the range is measured with a *resolution* of $\varepsilon = \frac{u*c_s}{2}[m]$. Since p is unknown, the range is also measured with an *ambiguity* of $a = \varepsilon * 2^n[m]$. As an example, using $n = 5[bits]$ to encode each delta value and considering an ambiguity of $a = 500[m]$, the resolution of the range is $\varepsilon \approx 16[m]$ which can be acceptable depending on the mission. Using the proposed encoding scheme, the parameters a and ε and n can be adapted to the mission requirements. When accurate ranging is possible, a lower value of ambiguity can be used (lowering n). Alternatively, if the AUV does not require high precision localisation, a low resolution of range can be used (again lowering n).

Localisation

In this section we show that even after encoding the deltas in an ambiguous way, they can still be used for AUV localisation. Figure 2 shows experimental results from the COLLAB_NGAS14 sea trial with the proposed encoding applied in post-processing. The ambiguity was set to $500[m]$. The localisation is performed using interval methods, as seen in [4,5], as they are suitable for non linear problems with multiple solutions. The result shows indeed multiple possible non overlapping solutions. Since usually underwater vehicles know their position when the mission starts (*e.g.* GPS fix on the surface), having non overlapping solutions allows the vehicle to correctly track the true position during the mission or until the next fix. The ambiguity therefore does not affect position tracking and the mobile vehicle is able to distinguish between the true solution and the ghost solutions.

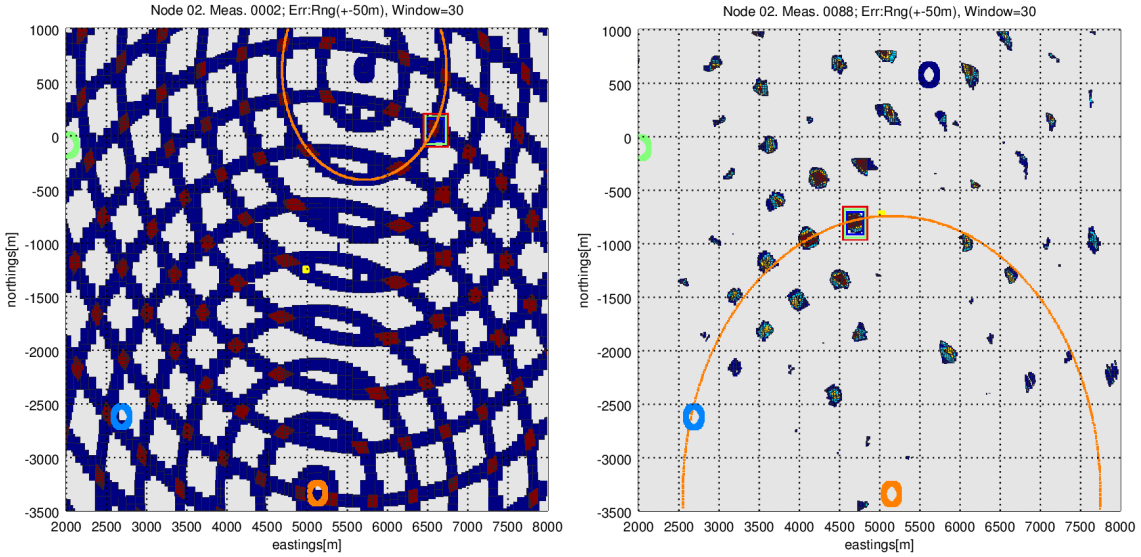


Figure 2: Localisation results: The squares show the actual position of the localised node while the bold circles are baseline node positions. The set of possible positions compatible with most of the measurements is coloured in red. The left image shows the result after 2 measurements. The right image shows the result after several measurements.

Acknowledgement

This work was supported by the Office of Naval Research Global under grant no. N62909-16-1-2095.

References

[1] A. Munafò and G. Ferri, "An acoustic network navigation system," *Journal of Field Robotics*, March 21 2017. [Online]. Available: http://dx.doi.org/10.1002/rob.21714

[2] O. Kebkal, K. Kebkal, and R. Bannasch, "Long-Baseline Hydro-Acoustic Positioning Using D-MAC Communication Protocol," in *Proceedings of MTS/IEEE OCEANS 2012*, Yeosu, Korea, May 21–24 2012, pp. 1–7.

[3] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 34–43, October 2007.

[4] A. Caiti, A. Garulli, F. Livide, and D. Prattichizzo, "Set-membership acoustic tracking of autonomous underwater vehicles," *Acta Acustica united with Acustica*, vol. 5, no. 88, pp. 648–652, 2002.

[5] J. Sliwka, A. Munafò, G. Ferri, and J. Alves, "Interval methods based AUV localization in the context of an acoustic network with experimental results," in *OCEANS'14 -St. John's*, 2014, pp. 1–6.

Comparison between Particle Filter and Interval Analysis for Wind Farm Targets Detection by Multistatic Radar System

Overview:

Wind farms cover large areas of land or sea, with large wind turbines structures that can reach up to 180m height, particularly in offshore environments. These installations can affect nearby radar systems including civilian radar, such as Air Traffic Control (ATC), marine, coastal, weather radars as well as defence installations. The radar's ability to detect, and position targets accurately might be degraded. The negative impact is caused by the high Radar Cross Section (RCS) from the turbine structure and/or the Doppler shift created by the turbine blades. Those impact factors make the wind turbine a complex target to accurately detect. Therefore, an alternative approach of radar system such as multistatic radar is considered to maximize on the advantages of multiple sensors and design flexibility to improve turbines detectability.

Range-Only target detection approach is often considered to achieve flexibility in design and reduction in cost and complexity of the radar system. However, this approach may require advanced signal processing techniques to effectively associate measurements from multiple sensors to estimate targets positions. This issue proved to be more challenging for the complex detection environment of a wind farm due to the increase in number of measurements from the complex radar scattering of each turbine.

The basic principles of multistatic radar are extended to use advanced target detection and tracking methods for multiple sensors to allow the use of multistatic radar in high target densities. The approaches used include, Interval Analysis (IA) and/or the statistical approach of Particle Filter (PF). To critically evaluate these techniques a computer simulation tool for Multistatic radar system was developed during the course of this research. The model was used alongside integrated wind farm model which computes the radar returns from individual turbines. This made it feasible to model the impact of wind farms and compare different multistatic radar processing techniques in this complex radar environment.

Comparison of IA vs. PF:

The comparison between PF and IA was performed according to three criterions: real targets detection rate, false targets detection rate and detection accuracy. These criterions were tested based on the two categories; performance analysis for multiple targets detection, and performance analysis for the detection of a 5x5 wind farm formation. The comparison results for each criterion were consistent for the two categories for both PF and IA.

Real targets detection rate:

The real target detection performance of the PF was relatively low. The technique of multiple results analysis for ghost elimination and the approach of particle nomination were the reason for low detection rate. The fluctuation in PF results because of the variance in particle placement for each run made it difficult to analyse multiple results with generic rule for matching detections without eliminating real targets detection. There was a need for trade-off between reducing the number of ghost targets and increasing the number of real targets detection.

IA showed high detection rate of real targets. This result was due to the process of measurements association within this approach. IA considers all range measurements of a bistatic pair as one group set and associates all group sets by finding all possible intersections among them. Therefore, real targets positions are more likely to be included within the estimated results by the intersection process. However, this does not guarantee all targets will be detected for every scenario as shown in the results which indicated that on average *90%* of the real targets were detected. IA can lose a target when multiple targets fall within the same range resolution or during the process of ghost detections elimination. Unlike PF, results of IA are consistent for every detection run of the same target and sensors layout but results from different subgroups of sensors can shift slightly according to the change in range resolution between sensors. This shift in results will cause a mismatch between multiple detection results and the intersection will be disregarded as ghost detection.

Ghost targets detection rate:

IA showed higher ghost detection rate than PF. The same reason for having high percentage of real target detection applies for this result. That is, solving for every possible intersection among measurements is likely to produce more ghost targets. Alternatively, the low number in ghost detection of PF is due to the iterative process of particles sampling and the multiple results analysis technique. Particle sampling eliminates ghost detection by relocating particles from the position of false estimation to locations where a real target is more likely to exist.

Detection Accuracy:

The accuracy of the PF was calculated based on the distance between target's real position and the nominated particle within the same range cell. If multiple particles were nominated, then the closest particle is considered.

The detection results from the IA approach were represented in interval of a rectangular shapes in X and Y dimensions. All possible combinations of x and y coordinates within that interval vector are valid as an estimated position for that target. Therefore, the detection accuracy is equal to the diagonal distance of the interval vector that includes real target position.

The interval method accuracy level was low because of the post detection processing on interval vectors, whereby the process of merging adjacent vectors and/or combining nearby vectors will increase the diagonal distances of interval vectors. PF shows better detection accuracy because the nominated particle can be no farther than the system range resolution.

Results from the comparison between PF and IA:

This section showcase the comparison between IA and PF performance according to detection results of 5x5 wind farm across *30* consecutive time steps. The results from each approach are shown according to the three comparison criterions mentioned before.

IA approach achieved high percentage of real target estimation throughout the detection window of *30* time steps, in contrast to PF low percentage around *50%* as shown in the percentage bar chart of figure 1.

High percentages in real targets detection of the IA approach were due to accounting for all possible measurements intersections between all bistatic pairs. On the other hand, the low numbers in real targets detection of PF were down to non-accurate particle placement, sampling process or ghost targets elimination technique.

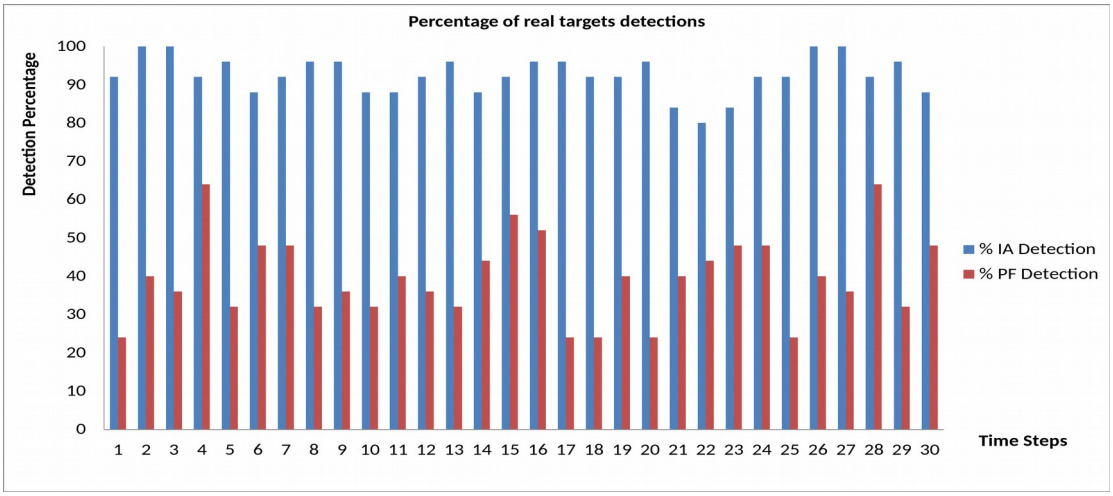


Figure 1: Percentage of real targets detection of IA and PF of a 5x5 wind farm.

The comparison between the two approaches based on number of ghost detection shows higher count of ghost targets detected by the IA than PF throughout all the time steps as shown in figure 2. The plot was produced according to the count of ghost detection at each time step individually without the tracking history log for ghost targets elimination. In the case of IA the appearance of ghost targets was due to solving for every possible intersection among measurements which is bound to produce more ghost targets. Alternatively, the low number in ghost detection of PF is due to particles sampling iterative process. Particle sampling eliminates ghost detection by relocating particles from the false estimation position to locations where more potential estimation of a real target exists.

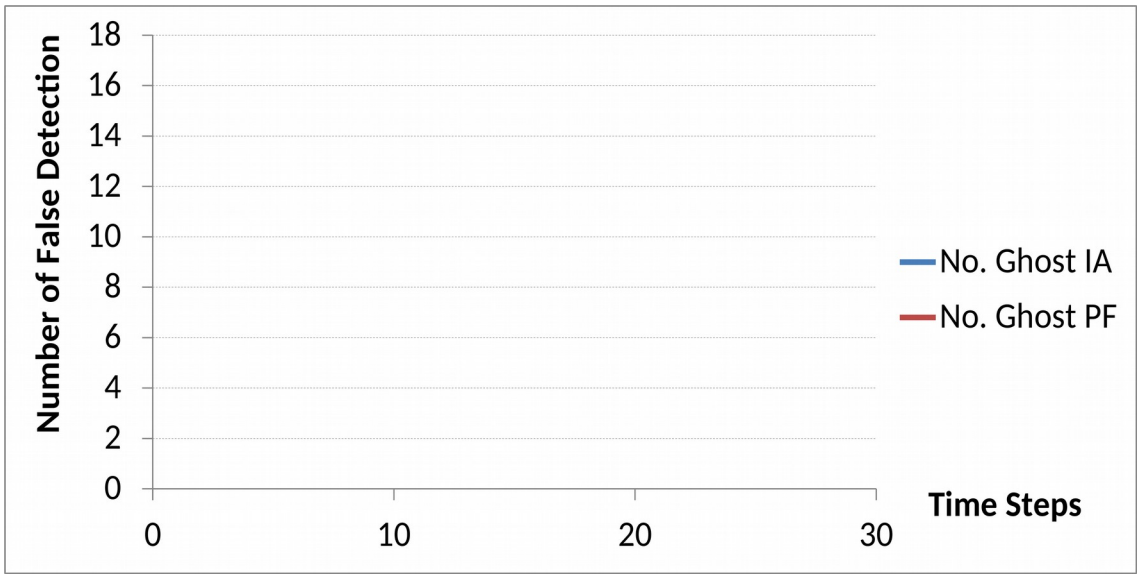


Figure 2: Number of false targets detection of IA and PF from a 5x5 wind farm.

The estimated target accuracy in PF is higher than IA, according to figure 3 and 4. Despite the complex detection scenario, the last step of PF is to nominate single particle for each range cell. Therefore, the estimated position will represent single position within the limits of that range cell resolution. In contrast, IA post detection processing of merging and combining intervals can result into extended accuracy range. Figure 5 combines the average values of accuracy from the late two plots in figure 3 and 4 for further illustration of the difference in the detection accuracy between the two methods.

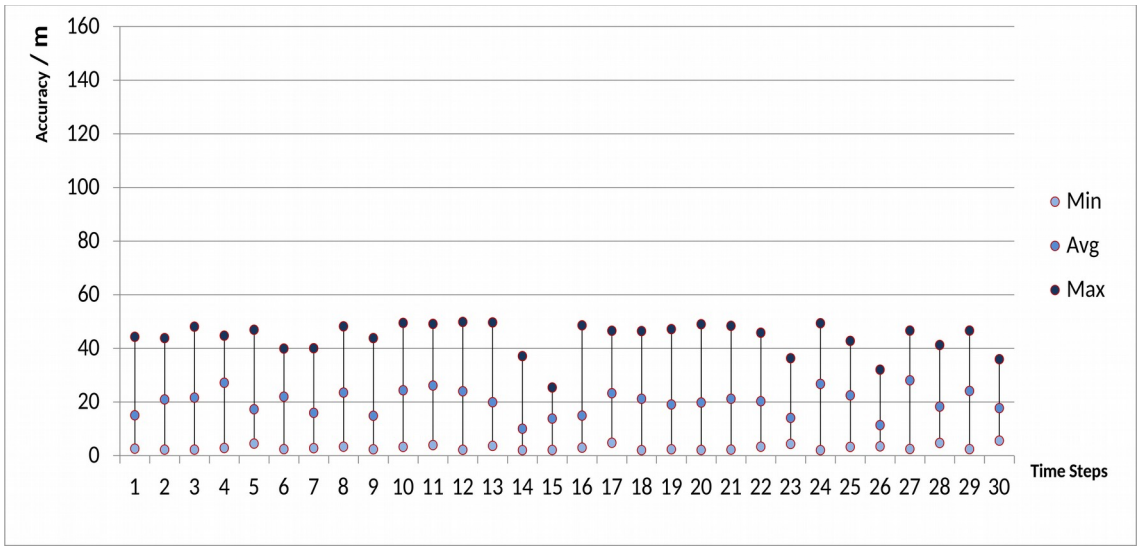


Figure 3: PF accuracy of maximum, Average and minimum detection accuracy.

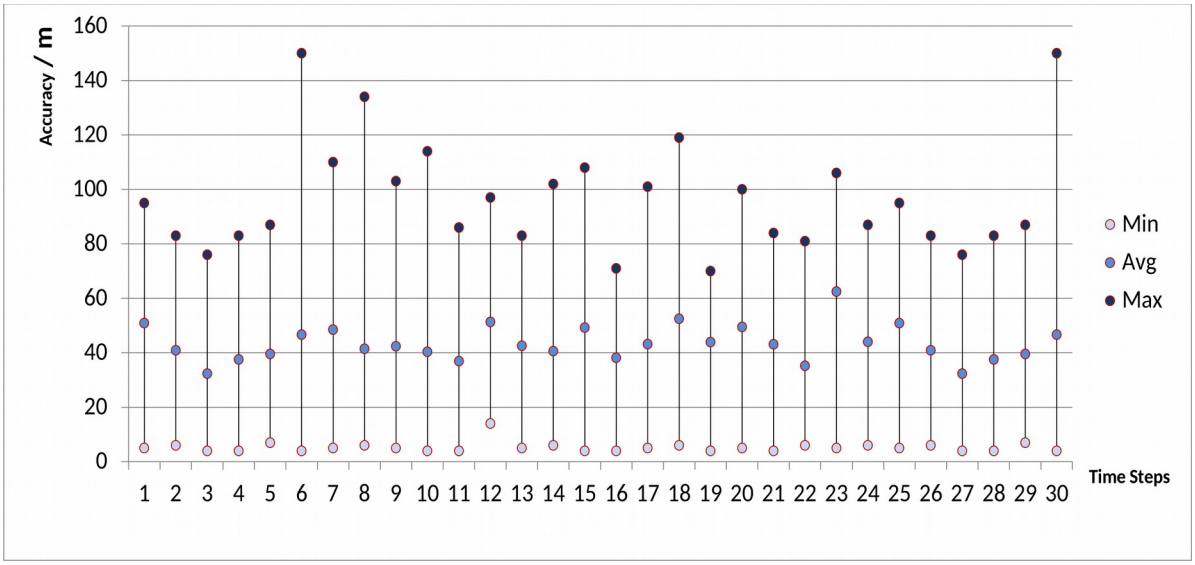


Figure 4: IA accuracy of maximum, Average and minimum detection accuracy.

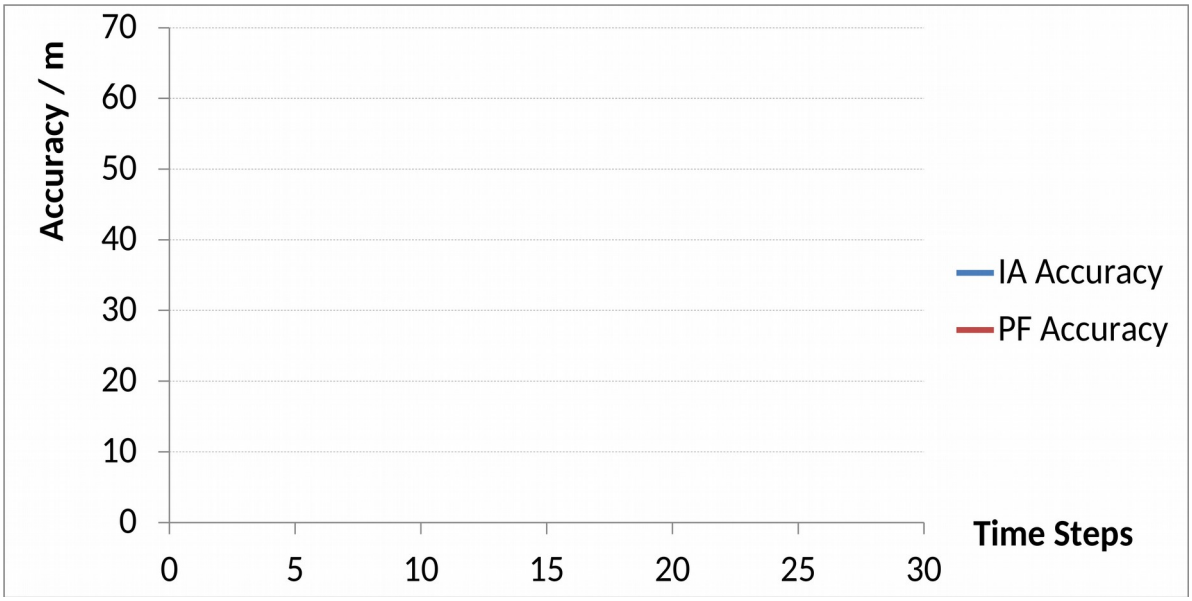


Figure 5: Comparison between PF and IA detection accuracy

Interval–state cellular automata and their applications to image segmentation

Irina Voiculescu[1], Imre Boros[2,3], Nicolae Popovici[3], Laura Dioşan[3] and Anca Andreica[3]

[1] Department of Computer Science
University of Oxford, UK
irina@cs.ox.ac.uk
[2] T. Popoviciu Institute of Numerical Analysis
Romanian Academy, Cluj–Napoca, Romania
imre.boros@math.ubbcluj.ro
[3] Department of Computer Science
Babeş–Bolyai University, Cluj–Napoca, Romania
{popovici,lauras,anca}@cs.ubbcluj.ro

Keywords: Interval comparisons, multi–state cellular automata, image segmentation, multi–agent systems

Introduction

In previous studies [1, 2] we have shown how multi–state Cellular Automata (CA) can be used for image segmentation. Our main aim is to segment features of semantic interest out of simple greyscale images, with a long–term goal of applying CA to the segmentation of medical data. We do this by imagining each pixel as an agent having to decide its membership to a particular geographic region, and changing its own value according to the values of its surrounding neighbours.

Comparisons of intervals

Our CA update rule relies heavily on comparisons of pixel values such as `if x<y then...` When `x` and `y` are intervals, the comparison gets tricky because of potential interval overlap.

Whilst for scalars there are only three elementary comparison operators ($<$, $=$ and $>$), there are as many as 18 well–defined relative

positions for intervals (see Figure 1). It is not evident how to name these, nor how to map them over the three scalar comparisons. Chiriaev and Walster [3] refer to 'certainly relations' and 'possibly relations.'

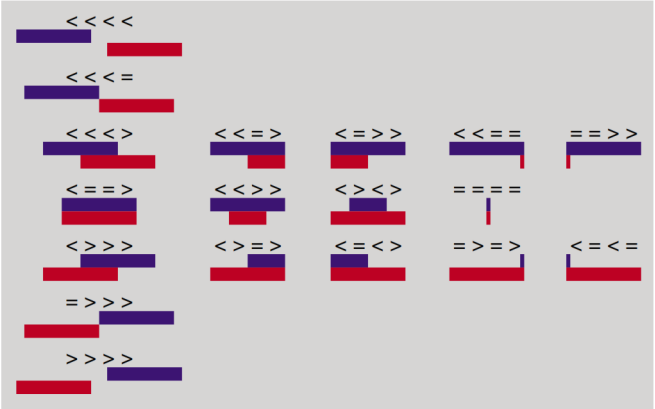


Figure 1: Encoding of 18 relative positions of intervals. Scalar relations of the four pairs of end points are listed in a fixed sequence.

CA rule based on interval components

Intervals can be used conveniently to classify pixels in an image into semanticly meaningful regions. Instead of scalar greyscale values, each pixel in an image can be represented as a range of greyscale values within $[0, 255]$. A CA can then be run over the set of pixels using a rule which requires at every iteration step for a pixel to use information from its Moore neighbourhood (3×3 or 5×5). It compares its own range to its neighbours' range, and decides how to change its range either to match or to distance itself from regions in its neighbourhood.

In order to compare intervals, we overload the usual operators. For example, based on the 18 meaningful relations in Figure 1, we can regroup the comparisons in the following way:

$<_{[]}$ <<<<, <<<=, <<<>, <<=>, <=>>, <<==, ==>>

$=_{[]}$ <==>, <<>>, <><>, ====

$>_{[]}$ <>>>, <>=>, <=<>, =>=>, <=<=, =>>>, >>>>

Interval $M = [\underline{M}, \overline{M}]$ which belongs to the current pixel in one iteration is adjusted according to the update rule in the following iteration:

- C_S is the interval closest in value to M where $C_S <_{[]} M$
- C_G is the interval closest in value to M where $C_G >_{[]} M$
- F_S is the interval furthest in value to M where $F_S <_{[]} M$
- F_G is the interval furthest in value to M where $F_G >_{[]} M$

In order to measure how far in value two intervals $\mathbf{x} = [\underline{x}, \overline{x}], \mathbf{y} = [\underline{y}, \overline{y}]$ are, we use the Hausdorff distance $d_H(\mathbf{x}, \mathbf{y}) = \max(|\underline{x} - \underline{y}|, |\overline{x} - \overline{y}|)$. If, for example, we denote $dC = d_H(C, M)$, then we can define update rules for the interval associated with each pixel. One of the simplest update rules for the close intervals can be defined thus:

if $dC_S < dC_G$ **then**
$$\underline{M} = \min(\overline{C_S}, \underline{M})$$
$$\overline{M} = \max(\overline{C_S}, \underline{M})$$
else
$$\underline{M} = \min(\underline{C_G}, \overline{M})$$
$$\overline{M} = \max(\underline{C_G}, \overline{M})$$

Any such rule constrains the greyscale range for each pixel to a set of discrete integer values. In the generic case, there would be no such overall requirement: it is easy to envisage a model whereby the greyscale range can become a true range over a real continuum. Indeed, one such model features within our future projects.

Results

We have obtained promising results using the above rule on synthetic images with an appropriate amount of noise, as well as on conventional real–world images. Initial experiments were carried out on a dozen images from the Berkeley Segmentation Dataset [4], which usefully provides the corresponding ground–truth human segmentations.

Figure 2 illustrates in left columns the raw data, and in right columns the corresponding machine segmentation generated after 30 iterations of our interval–based algorithm. The interval pixels obtained

after the final iteration were thresholded with a method similar to Otsu's [5], where greyscale values are split according to the peaks in the image hystogram, thus generating semantically interesting regions in the image. The methods can be improved further by using conventional pre– and post–processing steps. We have intentionally applied the CA in the absence of any conventional filtering methods so as to highlight the viability of the CA–based segmentations.

Figure 2: Raw data and corresponding machine segmentations obtained after 30 iterations, with tresholding.

References

[1] Laura Dioşan, Anca Andreica, Imre Boros and Irina Voiculescu. "Avenues for the Use of Multi–State Cellular Automata." Proc. Evolutionary Algorithms and Complex Systems. EvoStar 2017.

[2] Laura Dioşan, Anca Andreica and Irina Voiculescu. "Parameterized Cellular Automata in Image Segmentation." SYNASC 2016.

[3] Dmitri Chiriaev, G. William Walster. "Interval Arithmetic Specification." Technical Report 1998.

[4] D. Martin, C. Fowlkes, D. Tal, J. Malik. "A Database of Human Segmented Natural Images." Proc 8th Intl Conf Comp Vis 2001.

[5] Nobuyuki Otsu. "A Threshold Selection Method from Gray-Level Histograms." IEEE Trans Syst Man Cybernetics 9.1 (1979): 62-66.

Primitive Shape Recognition using Interval Methods

Salvador Pacheco-Gutierrez

The University of Manchester, Faculty of Engineering and Physical Sciences, Oxford Road, Manchester, United Kingdom

salvador.pachecogutierrez@manchester.ac.uk

Keywords: point cloud, parameter estimation, geometrical shapes

Introduction

In recent years, the development of laser scanner technologies has allowed us to gather real-world information from a three dimensional perspective; this is, by means of point clouds.

This work is focused on the characterisation of basic geometrical shapes such as lines, circles, planes, spheres, and cylinders immersed in a point cloud. Such shapes, specially cylinders and planes may represent objects such as vessels or containers. The automated characterisation of shapes is, among other applications, required for blue print retrieval particularly in cases where the access is limited or the environment is hazardous.

In this work, the information is retrieved using the structured-light scanner Kinect, whose measurement's statistical error distribution is not adequately defined by the inventors. Due to this, interval methods results to be the appropriate tool since the only requirement to guarantee the parameter detection is that the error is bounded.

Shape Detection using Interval Methods

The determination of the parameter vector $\mathbf{p}$ of a function $\mathbf{f}(\mathbf{p}, \mathbf{y})$, being $\mathbf{y}$ a set of measurements with known uncertainty can be approached

using interval analysis. Such a problem can be characterised as follows [1,2]:

$$\mathbb{P} = \bigcap_{i=\{1\dots m\}} \mathbb{P}_i \tag{1a}$$

$$\mathbb{P}_i = \{\mathbf{p} \in \mathbb{R}^{n_p}, \exists [\mathbf{y}] \in [\mathbf{y}]_i \mid \mathbf{f}(\mathbf{p}, \mathbf{y}) = \mathbf{0}\} \tag{1b}$$

being $\mathbf{p}$ the parameter vector, $[\mathbf{y}]_i \in \mathbb{R}^{n_y}$ is the ith measurement box and $\mathbf{f}$ is the model function. In this manner, Eq. 1b represents the posterior[1] parameter vector $\mathbb{P}_i$ that satisfies $\mathbf{f}$ at the $[\mathbf{y}]_i$ measurement box.

In order to counteract the effect of outliers acquired during the data gathering process, a relaxation procedure is implemented [1]. Let us consider the boxes $[\mathbf{x}]_1 \dots [\mathbf{x}]_m \in \mathbb{R}^n$. The q-relaxed intersection can be expressed as follows:

$$[\mathbf{x}]^{\{q\}} = \bigcap_{i=1\dots m}^{\{q\}} [\mathbf{x}]_i \tag{2}$$

which represents the set of all $x \in \mathbb{R}^n$ belonging to all $[\mathbf{x}]_i$'s except q at most. In a similar fashion, expression 1a can be redefined as follows:

$$\mathbb{P}^{\{q\}} = \bigcap_{i=\{1\dots m\}}^{\{q\}} \mathbb{P}_i \tag{3}$$

Case Studies

2D Line

Using the Hessian normal form, a two-dimensional line can be determined by the set $\mathcal{P}_L$ such that $\mathbf{n}^T \mathbf{y}_i = d$, being $\mathbf{n} = \{n_x, n_y\}^T$ a unit vector pointing from the origin to the line, and $d \geq 0$ the distance

[1] In this context, the word *posterior* refers to the parameter vector obtained after taking into consideration the model constraint $\mathbf{f}(\mathbf{p}, \mathbf{y}) = \mathbf{0}$.

from the origin to the line. The determination of the parameter vector $\mathbf{p}_L = \{n_x, n_y, d\}^T$ can be cast as a constraint satisfaction problem (CSP) as follows:

$$\mathcal{H}_L \ : \ \begin{cases} (\mathbf{n}^T \cdot \mathbf{y}_i - d = 0, \quad \mathbf{y_i} \in [\mathbf{y}]_\mathbf{i}) \\ ||\mathbf{n}|| = 1 \end{cases} \tag{4}$$

where $||\cdot||$ is the vector norm and $\cdot$ the dot product. Using $\mathcal{H}_L$, the parameter vector $\mathbf{p}_L$ can be characterised using SIVIA.

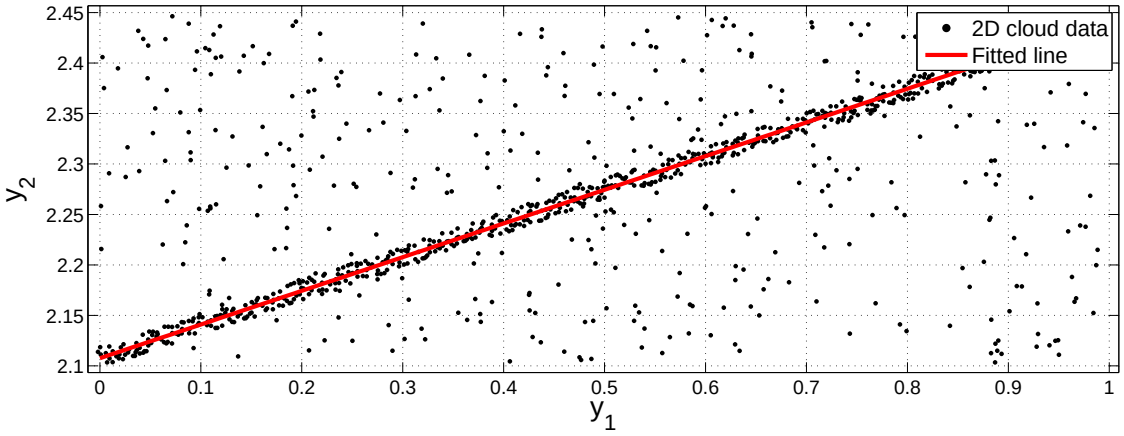


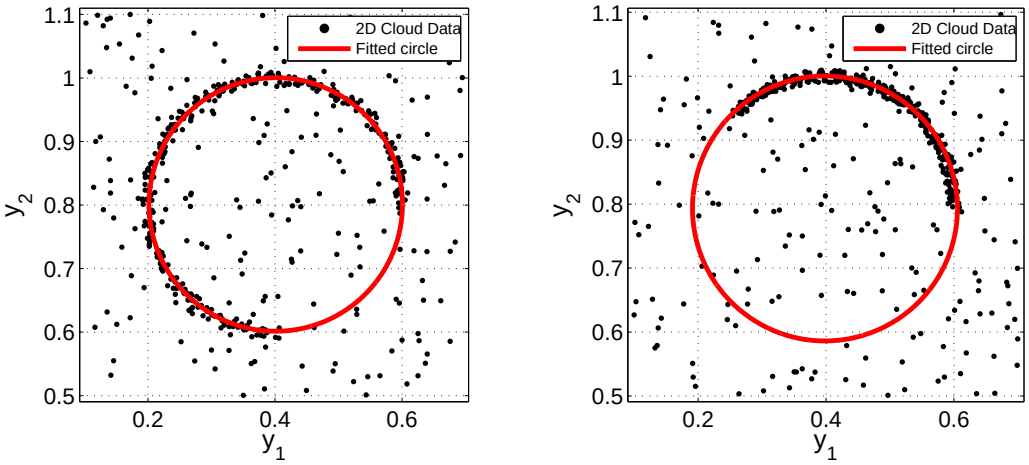
Figure 1: Two-dimensional line fitted using interval analysis. The total number of points is $m = 1000$ with ± 1 cm of noise and 40% of outliers. The true parameter vector is $\mathbf{p}_L = \{-0.3162, 0.9487, 2.0\}$. The result obtained was $[\mathbf{p}_L] = \{[-0.3475, -0.2856], [0.9296, 0.9678], [1.9716, 2.0273]\}$. The red line represents the model at mid($[\mathbf{p}_L]$).

2D Circle

The parameter vector for a 2D circle defined as $\mathbf{p}_C = \{p_1, p_2, p_3\}$ represents the x and y coordinates of the centre of the circle and its radius respectively. The determination of $\mathbf{p}_C$ can be cast as a CSP as follows

$$\mathcal{H}_C \ : \ \mathbf{f}_C(\mathbf{p}_C, \mathbf{y}_i) = \mathbf{0}, \ \mathbf{y}_i \in [\mathbf{y}]_i \tag{5a}$$

$$f_C(\mathbf{p}_C, \mathbf{y}) = (y_1 - p_1)^2 + (y_2 - p_2)^2 - p_3^2 \tag{5b}$$



(a) 75%. (b) 30%.

Figure 2: Two-dimensional circle fitted using interval analysis with different amount of descriptive inliers. The percentage represents the portion of the circle perimeter covered by the inliers. The true parameters are $\mathbf{p}_C = [0.4, 0.8, 0.2]$ with $m = 500$ points, ± 1 cm of noise and 40% of outliers. The red circle represents the model at mid($[\mathbf{p_C}]$).

Planes

A plane can be interpreted as an extension of the Hessian form of a line. In other words, the contractor presented in Eq. 4 can be used to determine the parameter vector $\mathbf{p}_P = \{n_x, n_y, n_z, d\}$, which in this case, includes the component over the Z axis of the direction vector.

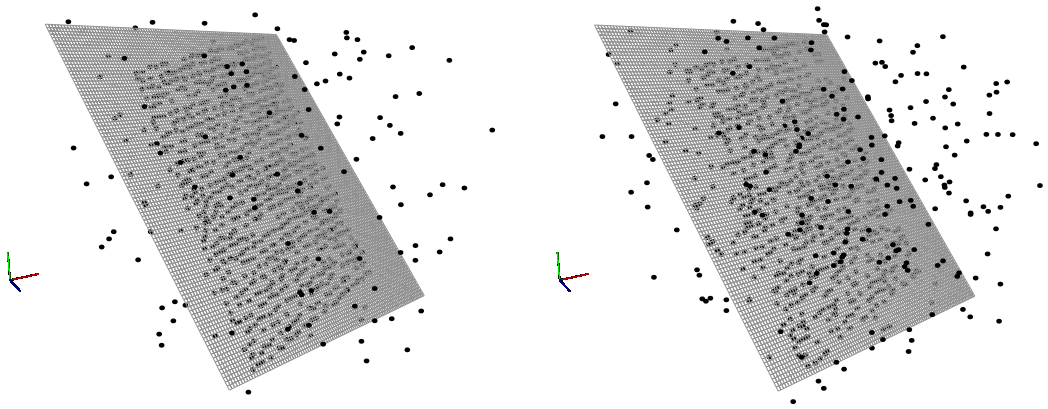
Sphere

Following the approach used with the 2D circle and its CSP expressed in Eq. 5a, a sphere can be characterised as follows:

$$\mathcal{H}_S \ : \ \mathbf{f}_S(\mathbf{p}_S, \mathbf{y}_i) = \mathbf{0}, \ \mathbf{y}_i \in [\mathbf{y}]_i \tag{6a}$$

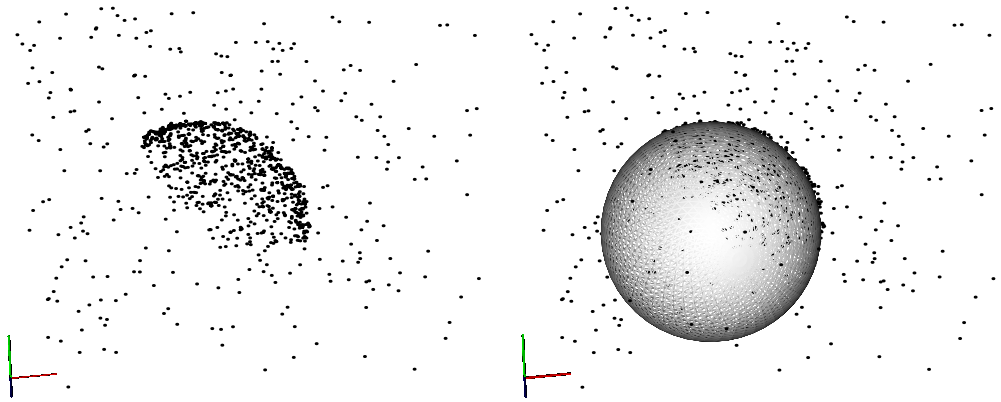
$$f_S(\mathbf{p}_S, \mathbf{y}) = (y_1 - p_1)^2 + (y_2 - p_2)^2 + (y_3 - p_3)^2 - p_4^2 \tag{6b}$$

where $\mathbf{p}_S = \{p_1, p_2, p_3, p_4\}$ is the parameter vector representing the spatial coordinates of the centre of the sphere and its radius respectively.



(a) 20% of outliers. (b) 40% of outliers.

Figure 3: Synthetic data used for plane detection. The true parameter vector is $\mathbf{p}_P = [0.3333, 0.6667, 0.6667, 1.0]$ with $m = 900$ points and ± 1 cm of noise. The grid plane represents the model at $\text{mid}([\mathbf{p}]_P)$.



(a) One third of sphere data. (b) Model detected.

Figure 4: Data sets used for sphere detection. For (a)–(d), the true parameters are $\mathbf{p}_S = [0.5, 1.0, 1.5, 0.3]$ with $m = 1000$ points, ± 1 cm of noise and 30% of outliers. For (e)–(f), $m = 2069$, the noise was estimated to be ± 3 mm and $\epsilon = 0.01$. The grid sphere model represents the model at $\text{mid}([\mathbf{p}_S])$.

Cylinder

A circular cylinder can be defined geometrically by its radius r, a vector $\mathbf{v} = \{v_x, v_y, v_z\}$ that defines the direction of its central axis and a pivot point $\mathbf{c} = (c_x, c_y, c_z)$ lying on $\mathbf{v}$. These parameters integrate the parameter vector $[\mathbf{p}_W] = \{\mathbf{v}, \mathbf{c}, r\}$.

Considering the geometrical relations between the measurements $[\mathbf{y}]_\mathbf{i}$ and the parameters that define the cylinder, it is possible to determine $[\mathbf{p}_W]$ by establishing the following constraint satisfaction problem:

$$\mathcal{H}_W \;:\; \begin{cases} r = \frac{||\mathbf{v}\times(\mathbf{y}_i - \mathbf{c})||}{||\mathbf{v}||}, & \mathbf{y_i} \in [\mathbf{y}]_\mathbf{i} \\ \mathbf{v}\cdot\mathbf{c} = 0 \end{cases} \tag{7}$$

Although this procedure may lead to successful results, determining $[\mathbf{p}_W]$ is highly computationally expensive because the SIVIA algorithm would be required to perform bisections along the seven dimensions of $[\mathbf{p}_W]$; this are the elements of $[\mathbf{p}_W]$ which are $v_x, v_y, v_z, c_x, c_y, c_z$ and r. Due to this, a dimensionality reduction in the parameter vector is conducted. Details about this procedure will be provided during the workshop.

Acknowledgement

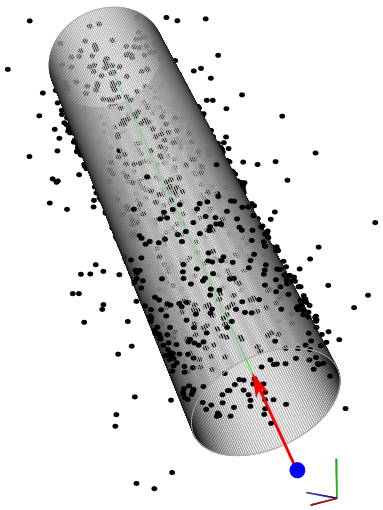
Special thanks to The National Council for Science and Technology (CONACyT), Dalton Nuclear Institute (DNI) and the Aerospace Institute of the University of Manchester.

References

[1] L. Jaulin and S. Bazeille, Image Shape Extraction using Interval Methods, *Sysid 2009* Saint Malo, France, 2009.

[2] L. Jaulin and E. Walter, Guaranteed bounded-error parameter estimation for nonlinear models with uncertain experimental factors, *Automatica* 35(5):849–856, 1999.

(a) Original and projected data over the plane defined by the cylinder axis.



(b) Cylindrical model detected

Figure 5: Synthetic data used for cylinder detection. The set is composed by $m = 1000$ points, ± 1 cm of noise and 20% of outliers. In (b), the blued dot represents $\text{mid}([\mathbf{c}])$ and the red arrow represents the direction vector $\mathbf{v}$. The cylindrical grid model represents the model with parameters $\mathbf{p}_W = \text{mid}([\mathbf{p}_W])$.

Set-membership functional diagnosability through linear functional independence

Carine Jauberthie[1], Nathalie Verdière[2],
Louise Travé-Massuyès[1]

[1] LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France, cjaubert,louise@laas.fr
[2] Normandie Univ, UNIHAVRE, LMAH, FR-CNRS-3335, ISCN, 76600 Le Havre, France, nathalie.verdiere@univ-lehavre.fr

Keywords: set-membership diagnosability, linear functional independence

Introduction

Before a system is put into operation and eventually diagnosed, diagnosability analysis is an important stage. Diagnosability indeed guarantees that the sensored values delivered by the available instrumentation can be processed into an appropriate set of symptoms allowing to discriminate a reasonable set of faulty situations.

A fault is considered as an additional parameter that impacts the behavior of some components of the system. Its effects may be linear or non-linear. Functional diagnosability, introduced in [1], and extended to set-membership (SM) functional diagnosability in [2] was analyzed through parameter identifiability. In the proposed work, SM-functional diagnosability is assessed from the linear independence of *SM-functional fault signatures*, which results in a much more direct test.

The system is assumed to be represented by the following model :

$$\Gamma \begin{cases} \dot{x}(t) = g(x(t), u(t), f, p), \ x(t_0) = x_0, \\ y(t) = h(x(t), u(t), f, p), \end{cases} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ and $y(t) \in \mathbb{R}^m$ denote the state variables and the outputs respectively. $u(t)$ is the input vector. The function g is real

and analytic on an open set of $\mathbb{R}^n$. p denotes the parameter vector belonging to a connected set $P \in \mathcal{U}_\mathcal{P}$, where $\mathcal{U}_\mathcal{P}$ is an open set of $\mathbb{R}^p$.

The fault vector $f \in \mathbb{R}^e$ belongs to a connected set $F \subseteq \mathcal{F}_{SYS}$, where $\mathcal{F}_{SYS}$ is the exhaustive set defining the fault domain. $Y(P, F, u)$ denotes the set of outputs, solution of Γ with the input u, the parameter vector P and the fault vector F. Considering a connected set of faults $F \subseteq \mathcal{F}_{SYS}$, let us denote by F_j the vector derived from F by setting all the components equal to zero except the jth component. F_j, called a "bounded fault', is a connected set describing a faulty situation characterized by the occurence of a fault whose magnitude is assumed to belong to the bounded set F_j.

Definitions and role of analytical redundancy relations

Using elimination theory, some differential polynomials or analytical redundancy relations (ARR) linking system inputs, outputs and their derivatives can be obtained. The use of ARRs makes possible to detect, isolate, and estimate the characteristics of a fault acting on the system. Specific ARRs indexed by $i = 1, ..., m$ can be obtained [1] :

$$\begin{aligned} w_i(y, u, f, p) = \; & m_{0,i}(y, u, p) - \textstyle\sum_{k=1}^{n_i} \gamma_k^i(f, p) m_{k,i}(y, u) \\ & = w_{0,i}(y, u, p) - w_{1,i}(y, u, f, p) \end{aligned} \tag{2}$$

where $(\gamma_k^i(f, p))_{1 \leq k \leq n_i}$ are rational in f and p, $\gamma_v^i \neq \gamma_w^i$ $(v \neq w)$ and $(m_{k,i}(y, u))_{1 \leq k \leq n_i}$ are differential polynomials with respect to y and u.

$w_{0,i}(y, u, p)$ is equal to $m_{0,i}(y, u, p)$, hence the first part of the polynomial does not contain components of f. It corresponds to the residual *computation form* whereas $w_{1,i}(y, u, f, p)$ is known as the residual *internal form*. Let us notice that (2) can as well be interpreted for bounded faults (vector F) and uncertain but bounded parameters (vector P).

One may be interested in distinguishing types of faults, independently of their magnitude. For instance, it may be important to detect a leakage on a pipe but the amount of derived flow may not be relevant. Hence the notion of *SM-functional diagnosability* that comes through the notion of *SM-functional signature*.

Definition 1. *The SM-functional signature of a bounded fault F_j is a function $FSig_{SM}(F_j)$ which associates to F_j the interval vector:*

$$(w_{1,i}(Y(P,u),u,F_j,P))_{i=1,...,m}.$$

$FSig^{(i)}_{SM}(F_j)$ denotes the ith component of $FSig_{SM}(F_j)$ and corresponds to an interval function. $FSig^{(i)}_{SM}(F_j)$ consists in a set of trajectories generated in the presence of F_j and can be viewed as a tube of trajectories on the time interval $[t_0, T]$.

To distinguish the two tubes of trajectories generated by two different bounded faults, we propose the following definitions. The first one refers to *weak* SM-functional diagnosability and permits an intersection of the two tubes on a time subinterval whereas the second one refers to *strong* SM-functional diagnosability and requires the two tubes to be totally disjoint on $[t_0, T]$.

Definition 2. *Two bounded faults F_j and F_k are SM-functionally discriminable if $FSig_{SM}(F_j)$ and $FSig_{SM}(F_k)$ are distinct, which includes the two following cases:*

- *there exists at least one index $i^* \in \{1, ..., m\}$ and a time interval $[t_1, t_2] \subseteq [t_0, T]$ such that for all $t \in [t_1, t_2]$, $FSig^{(i*)}_{SM}(F_1) \cap FSig^{(i*)}_{SM}(F_2) \neq \emptyset$ and $FSig^{(i^*)}_{SM}(F_1) \not\subseteq FSig^{(i^*)}_{SM}(F_2)$ or $FSig^{(i^*)}_{SM}(F_2) \not\subseteq FSig^{(i^*)}_{SM}(F_1)$, in which case F_j and F_k are said to be **weakly SM-functionally discriminable.***
- *there exists an index $i^* \in \{1, ..., m\}$ and a time interval $[t_1, t_2]$ such that for all $t \in [t_1, t_2]$, $FSig^{(i^*)}_{SM}(F_1) \cap FSig^{(i^*)}_{SM}(F_2) = \emptyset$, in which case F_j and F_k are said to be **strongly SM-functionally discriminable**.*

Definition 3. *The model Γ given by (1) is weakly (resp. strongly) SM-functionally diagnosable for $\mathcal{F}_{SYS}$ if any two bounded faults F_j, $F_k \subseteq \mathcal{F}_{SYS}$ are weakly (resp. strongly) SM-functionally discriminable.*

Analysis of SM-functional diagnosability

To analyse SM-functional diagnosability, a criterion testing the linear independence of SM-functional fault signatures is proposed.

Definition 4. *Rewriting* $FS_{i,j} := FSig^{(i)}_{SM}(F_j)$*, the SM-signature matrix is defined as* $MSig = (FS_{i,j})_{1\leq i\leq m,\ 1\leq j\leq e}$*. Define also the extended following matrix for the case* $i = 1$*:*

$$W_{MSig} = \begin{pmatrix} FS_{1,1} & FS_{1,2} & \dots & FS_{1,e} \\ FS'_{1,1} & FS'_{2,2} & \dots & FS'_{1,e} \\ \vdots & \vdots & \vdots & \vdots \\ FS^{(e-1)}_{1,1} & FS^{(e-1)}_{1,2} & \dots & FS^{(e-1)}_{1,e} \end{pmatrix}. \tag{3}$$

Proposition 1. *1. Consider the case* $i = 1$ *– If* $det(W_{MSig}) \not\equiv 0$*, then the faults of the set* $\mathcal{F}_{SYS} = \{F_1, \dots, F_e\}$ *are discriminable, as well as any subset of faults within this set* [1]*.*

2. Consider the case $i = m > 1$ *and assume that* $m \leq e$ *– If the minor* $M_{I,J}$ *of order* α*,* $2 \leq \alpha \leq m$*, of* $MSig$ *is not zero then the faults in the set* $\{F_j, j \in J\} \subset \mathcal{F}_{SYS}$ *are discriminable, as well as any subset of faults within this set.*

Note that Proposition 1 gives conditions on the rank of the interval matrices. Proofs are omitted due to lack of space.

Conclusion

A new approach to test the concept of SM-functional diagnosability is proposed. It leads to a test based on the rank of an interval matrix formed from the SM-functional signatures of the faults.

References

[1] N. Verdière, C. Jauberthie, L. Travé-Massuyès, Functional diagnosability and detectability of nonlinear models based on analytical redundancy relations, Journal of Process Control, Vol. 35, pp. 1–10, 2015.

[2] C. Jauberthie, N. Verdière, L. Travé-Massuyès, Set-Membership diagnosability: definitions and analysis, In Proceedings of International Conference on Control and Fault-Tolerant Systems, 6 pages, Barcelona, Spain, 2016.

[3] E. Goubault, M. Kieffer, O. Mullier, S. Putot. General Inner Approximation of Vector-valued Functions. Reliable Computing, Vol. 18, pp. 117-143, 2013.

[1] Note that the determinant of W_{MSig} is the classical wronskian.

Examples on Verified Diagnosis of Safety Critical Dynamic Systems Based on Kaucher Interval Arithmetik

Stefan Schwab[1], Oliver Stark[1] and Soeren Hohmann[1]

[1] Karlsruhe Institute of Technologie, Institute of Control Systems,
Kaisterstr. 12, 76131 Karlsruhe, Germany
{stefan.schwab@kit.edu}

Keywords: Verification, safety critical, Kaucher interval arithmetic

Introduction

Safety critical systems are of high importance in various domains (e.g. automated driving, aviation, battery management systems, medical devices). Because a failure in a safety critical system can have severe implications high safety integrity levels have to be met, i.e. the system always has to work correctly. Recent developments use interval arithmetic to assess properties of safety critical systems. There is a wide range of theory for state observers [14, 8, 3, 7], identification [10] and simulation [9, 4, 5]. Approaches that use interval arithmetic in the diagnosis setting are given by [12, 13, 2, 6]. Those methods are aiming on the guaranteed detection of failures. The provided frameworks will never create false positives (type I errors, "false alarms"). As they are using classical interval arithmetic their solutions are minimum outer inclusions of the real solution. It is widely known that this calculation method produces so called spurious solutions that are included in the solution interval but are actually no solution of the genuine problem [1]. There might be false negatives as those methods cannot differentiate between real and spurious solutions. Erroneous behaviour can thus be hidden in the spurious solution set.

The authors proposed a new method for the diagnosis of dynamical

systems based on input and output measurement data at the IFAC World Congress 2017 [11]. The proposed verification method uses a system specification in which the system behaviour is defined by its dynamic parameters. The method uses the notation of Kaucher interval arithmetic to handle inevitable measurement noise. Known sensor properties are used to set up the interval inclusion of distorted measurement that is guaranteed to include the true value. The introduced method can be extended to handle unsharp requirements, e.g. due to limited knowledge or wide tolerances. These can be modelled directly in the specification using the proposed interval notation. The Kaucher interval arithmetic calculation and noise handling lead to mathematically guaranteed results which represents a new quality of the verdicts compared to existing methods in the literature.
This paper summarizes the main ideas of the proposed procedure and shows several examples calculated by the new method.

Acknowledgement

This work was founded by ITK Engineering AG and the State of Baden-Wuerttemberg.

References

[1] G. Alefeld and G. Mayer. Interval analysis : theory and applications. *Journal of Computational and Applied Mathematics*, 121:421–464, 2000.

[2] J. Armengol, J. Vehi, M. Angel Sainz, P. Herrero, and E. R. Gelso. Squaltrack: A tool for robust fault detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):475–488, April 2009.

[3] D. Efimov, T. Raïssi, Perruquetti W., and A. Zolghadri. Estimation and control of discrete-time lpv systems using interval ob-

servers. In *IEEE 52nd Annual Conference on Decision and Control (CDC)*, 2013.

[4] J. Heeks, E.P. Hofe, B. Tibken, and K. Thorwart. An interval arithmetic approach for discrete time nonsmooth uncertain systems with application to an antilocking system. In *Proceedings of the 2002 American Control Conference*, 2002.

[5] P. Herrero, P. Georgiou, C. Toumazou, B. Delaunay, and L. Jaulin. An efficient implementation of SIVIA algorithm in a high-level numerical programming language. *Reliable Computing*, 16:239–251, 2012.

[6] A. Ingimundarson, J. M. Bravo, V. Puig, T. Alamo, and P. Guerra. Robust fault detection using zonotope-based set-membership consistency test. *International Journal of Adaptive Control and Signal Processing*, 23(4):311–330, 2009.

[7] S. Krebs, C. Schnurr, M. Pfeifer, J. Weigold, and S. Hohmann. Reduced-order hybrid interval observer for verified state estimation of an induction machine. *Control Engineering Practice*, 57:157 – 168, 2016.

[8] R. Lamouchi, M. Amairi, T. Raïssi, and M. Aoun. Interval observer design for linear parameter-varying systems subject to component faults. In *24th Mediterranean Conference on Control and Automation (MED)*, 2016.

[9] A. Rauh, E.P. Hofer, and E. Auer. Valencia-ivp: A comparison with order initial value problem solvers. In *12th GRAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics*, 2006.

[10] A. Rauh, L. Senkel, and H. Aschemann. Experimental comparison of interval-based parameter identification procedures for uncertain odes with non-smooth right-hand sides. In *20th International*

Conference on Methods and Models in Automation and Robotics (MMAR), 2015.

[11] S. Schwab, O. Stark, and S. Hohmann. Verified diagnosis of safety critical dynamic systems based on kaucher interval arithmetic. In *IFAC 2017 World Congress, Toulouse, France*, 2017, to appear.

[12] F. Wolff. *Konsistenzbasierte Fehlerdiagnose nichtlinearer Systeme mittels Zustandsmengenbeobachtung.* Schriften des Instituts fuer Regelungs- und Steuerungssysteme, Karlsruher Institut fuer Technologie ; 09. KIT Scientific Publishing, Karlsruhe, 2010. Zugl.: Karlsruhe, KIT, Diss., 2010.

[13] S. Zaiser, Buchholz, M., and Dietmayer, K. Black-box modeling with uncertain parameters from measurement data with unknown, but bounded errors. *at - Automatisierungstechnik*, 62(9):607–618, 2014.

[14] G. Zheng, D. Efimov, F.J. Bejarano, and Perruquetti W. On the interval estimation for nonlinear singular system. In *IEEE 54th Annual Conference on Decision and Control (CDC)*, 2015.

Fault Tolerant Control using Viability Theory

Vicenç Puig [2], Majid Ghaniee Zarch[1] and Javad Poshtan[1]

[1] Electrical Engineering Department, Iran University of Science and Technology (IUST), Narmak, 16846-13114 Tehran, Iran
[2] Institut de Robòtica i Informàtica Industrial (CSIC-UPC). Carrer Llorens Artigas, 4-6, 08028 Barcelona
vicenc.puig@upc.edu

Keywords: Fault-tolerant control, viability theory

Introduction

Modern control systems are developed taking into account the demand for reliability, safety and fault tolerance. Consequently, it is necessary to design control systems which are capable of tolerating potential faults. A closed-loop control system which can tolerate component malfunctions, while maintaining desirable performance and stability properties is said to be a fault tolerant control system [7].

Viability theory develops mathematical and algorithmic methods for investigating the adaptation to viability constraints of evolutions governed by complex systems under uncertainty [1]. Viability is a theory that until now has mostly used in safety verification in control systems [5]. Viability theory has also been found useful in areas different from automatic control as e.g. economics [3] or biology [2]. This theory provides some concepts that are actually more general than what is used in set and set-invariance theory. Viability kernel is an accepted tool for safety verification. However, the problem with this theory is how to compute the different sets involved. Nowadays, several algorithms have been proposed that can approximate these sets effectively. Some of these algorithms are surveyed [5]. Finding the viability theory concepts that can be used in fault tolerance evaluation is a major contribution of this paper. This paper will also try to relate

these concepts with set-based concepts introduced to address the admissibility evaluation defined in [6] in the context of Model Predictive Control (MPC).

Faults will cause changes in the set of feasible solutions. This causes that the set of admissible solutions for the control objective could be empty. Therefore, the admissibility of the control law facing faults can be determined knowing the feasible solution set. One of the aim of this paper is to provide methods to compute this set and the evaluate the admissibility of the control law.

This paper will focus on the fault tolerant evaluation of a given fault configuration, considering a nonlinear predictive control law with constraints. The method proposed in this paper is not of analytical but of computational nature. It follows the idea proposed by [6]. Faults can cause changes in the constraints related to the control signals (inputs), which modifies the set of feasible solutions of the MPC controller. This can cause the set of admissible solutions for a given control objective to be empty. An algorithm based on viability theory concepts will be provided to evaluate the admissibility of the control law for a given fault configuration.

Fault-tolerance assessment using viability theory

Let us denote the sequence variables over the time horizon N

$$\begin{aligned} \tilde{x} &= (x_k)_0^N = (x_0, x_1, ..., x_N), \\ \tilde{u} &= (u_k)_0^{N-1} = (u_0, u_1, ..., u_{N-1}). \end{aligned}$$

Thus, in the case of a model predictive control law, the triple $\langle O, C, U \rangle$ is defined by

$$O : \min_{\tilde{u}} \ J(\tilde{x}, \tilde{u}) \tag{1}$$

subject to

$$C : \begin{cases} x_{k+1} = f\left(x\left(k\right), u\left(k\right)\right) \\ u_k \in U \qquad k \in [0, N-1] \subset \mathbb{N} \\ x_k \in X \qquad k \in [0, N] \subset \mathbb{N} \end{cases} \tag{2}$$

where

$$U \triangleq \left\{u \in \mathbb{R}^m \,|\, u_{\min} \leq u \leq u_{\max}\right\} \tag{3}$$

$$X \triangleq \left\{x \in \mathbb{R}^n \,|\, x_{\min} \leq x \leq x_{\max}\right\}, \tag{4}$$

The control law belongs to the set U and it is obtained using the *receding horizon philosophy* [4]. This technique consists on taking only the first value from the sequence $\tilde{u}$ computed at each time instant by solving the previous optimization problem. The initial states x_0 are updated from measurements or state estimation.

Definition 1. *[**Feasible solution set**] The feasible solution set of the MPC problem (3)-(4) is given by*

$$\Omega = \left\{\tilde{x}, \tilde{u} | \left(x\left(k+1\right) = f\left(x\left(k\right), u\left(k\right)\right)\right)_0^{N-1}\right\}$$

The subset Ω gives the input and state sets compatible with system constraints which originate the set of predictive states.

Definition 2 (**Feasible control objective set**)**.** *The feasible control objective set is given by*

$$\Gamma_\Omega = \left\{J\left(\tilde{x}, \tilde{u}\right) \in \mathbb{R} | \left(\tilde{x}, \tilde{u}\right) \in \Omega\right\}$$

and corresponds to the set of all values of J obtained from feasible solutions.

Consider the system with fault as:

$$x\left(k+1\right) = f\left(x\left(k\right), u\left(k\right), \theta_f\right) \tag{5}$$

In this case, feasible solution set Ω converts to Ω_f and feasible control objective set Γ_Ω converts to Γ_{Ω_f}.

Definition 3 (**Admissible solution set**). *Given the following subsets*

- Ω_f, *defined as the feasible solution set*
- Γ_Ψ, *defined as the admissible control objective set,*

the admissible solution set is given by

$$\Psi = \{\tilde{x}, \tilde{u} \in \Omega_f | J(\tilde{x}, \tilde{u}) \in \Gamma_\Psi\}$$

and corresponds to the feasible solution subset that produces control objectives in Γ_Ψ. *If* $\Psi = \emptyset$, *then the system* (5) *is not fault tolerant.*

Based on the viability concepts introduced in [1], it can be readily deduced that there are some similarities that allow us to use viability theory in fault tolerance evaluation. Actually, an equivalency between feasible solution set and viability kernel can be considered

$$Viab_S(K) \equiv \Omega$$

Note that for finding both of them, constraints of the system is considered. But in the viability kernel definition, there is an extra limitation that the system must have at least one evolution that remains in the set. This is close to the concept of Lyapanov theory for stability. Therefore, viability kernel is more reliable to provide safe areas of work for the system. It can be deduced that if reference of the system is inside viability kernel, it is achievable. Actually, there is a control signal that can bring the system to reference.

The equivalence between viability kernel and feasible solution set leads us to relate the capture basin with the set of admissible performance

$$Capt_S(K, C) \equiv \Gamma_\Omega$$

In the definition of capture basin, the target C can be regarded as objective J that must be reached. It means that if there is a limited time to achieve the target after fault occurs while the states of the system must be in the capture basin.

Note that in definition of viability kernel and capture basin, despite feasible solution set and feasible control objective set, there is no direct mention regarding the control signal. Therefore, regulation map can be used as complementary concept to deal with the control signal.

Algorithm

Now, after finding those equivalency, fault-tolerance evaluation that is proposed in [6] can be addressed by viability theory concepts. The admissibility evaluation starts obtaining the viability kernel $viab_S(K)$ given a set of initial states K_0 and the system dynamics. This procedure is described in the Appendix.

After finding viability kernel based on constraints of states and inputs, the capture basin can be obtained. In this manner, it is possible to consider viability kernel or a part of the set (based on steady state or a predefined objective trajectory) as target to find capture basin.

Given a fault in the system, the admissible solution set can be obtained from the above algorithms using revised system dynamics and constraints. In this manner, the new viability kernel can define the set of admissible states of the system after fault occurs. Therefore, it is possible to investigate if the reference is achievable or not.

On the other hand, finding capture basin with new dynamics allows to determine in at least how many steps the system can reach the target. The target can be considered as small set near steady state inside viability kernel or a small set around a predefined trajectory. *Algorithm 1* shows the procedure for admissibility evaluation using viability theory concepts. In the final version of the paper, the algorithm will be tested using a mobile robot.

Acknowledgement

This work has been partially funded by the Spanish Government (MINECO) through the project CICYT ECOCIS (ref. DPI2013-48243-C2-1-R), by

Algorithm 1 Fault-tolerance evaluation using viability theory

find $Viab_S(K)$
if the reference x_{ref} is inside $Viab_S(K)$ **then**
  x_{ref} is achievable
else
  x_{ref} is not achievable
end if
find $Capt_S(K, C)$ of a target C inside $Viab_S(K)$
if the target C is achievable in finite time T **then**
  system is fault tolerant
else
  system is not fault tolerant
end if

MINECO and FEDER through the project CICYT HARCRICS (ref. DPI2014- 58104-R).

References

[1] Jean-Pierre Aubin, Alexandre M Bayen, and Patrick Saint-Pierre. *Viability theory: new directions*. Springer Science & Business Media, 2011.

[2] Guillaume Deffuant and Nigel Gilbert. *Viability and resilience of complex systems: concepts, methods and case studies from ecology and society*. Springer Science & Business Media, 2011.

[3] H-W Lorenz and J-P Aubin. Dynamic economic theory: a viability approach, 1999.

[4] J.M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, Great Britain, 2002.

[5] John N Maidens, Shahab Kaynama, Ian M Mitchell, Meeko MK Oishi, and Guy A Dumont. Lagrangian methods for approximat-

ing the viability kernel in high-dimensional systems. *Automatica*, 49(7):2017–2029, 2013.

[6] C Ocampo-Martinez, P Guerra, V Puig, and J Quevedo. Actuator fault-tolerance evaluation of linear constrained model predictive control using zonotope-based set computations. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 221(6):915–926, 2007.

[7] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual reviews in control*, 32(2):229–252, 2008.

Improving Guaranteed Coverage Assessment of a Robotic Survey in the Translation Invariant Case

Vincent Drevelle

Université de Rennes 1, IRISA, INRIA Rennes-Bretagne Atlantique,
Lagadic project
Campus de Beaulieu, 35042, Rennes, France
vincent.drevelle@irisa.fr

Keywords: Interval analysis, robotics, explored area

Introduction

Robots are often employed for tasks that consists in covering an given area, like survey missions, search and rescue, or other tasks such as lawn-mowing.

In practice, the robot trajectory is known with an uncertainty, which means the zone that has been actually covered during the mission is also uncertain. Assessing the coverage of the mission is an important task [1], to ensure there will be no untreated zone.

Interval analysis can be employed to compute an upper and lower bound of the zone covered during a mission [2,3]. We show how in the case of translation independent covered domains, Minkowski difference of sets can be employed to improve computation of the explored zone.

Problem statement

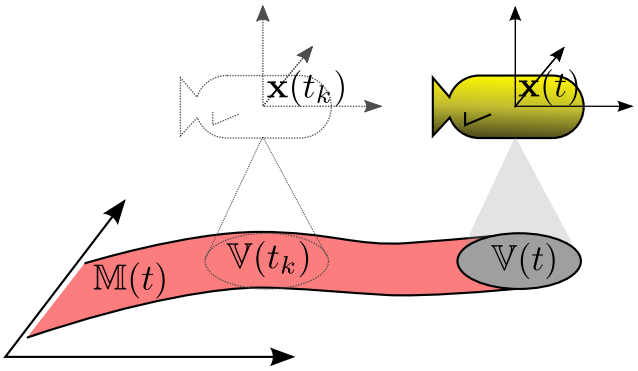
Zone covering by a mobile robot can be modeled as follows Let us consider a mobile robot equipped with actuators and sensors. The robot is classically represented by the following state equations

$$\left\{ \begin{array}{rcl} \dot{\mathbf{x}}(t) & = & \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \\ \mathbf{y}(t) & = & \mathbf{g}\left(\mathbf{x}(t)\right), \\ \mathbb{M} & = & \bigcup_t \mathbb{V}(\mathbf{x}(t)) \end{array} \right. \tag{1}$$

where $\mathbf{x}$ denotes the robot's state vector (e.g. position, velocity...), $\mathbf{u}$ is the input vector (to robot actuation) and $\mathbf{y}$ is the observation vector (measurements from the navigation sensors). The robot's evolution is modeled by the function $\mathbf{f}$ and $\mathbf{g}$ is the observation function. $\mathbb{V}$ is the set-valued function that returns the zone which is in the robot's field of view, i.e the *visible zone*. The *covered zone* $\mathbb{M}$ is the union of all visible zones during the whole mission.

Visible zone is defined by a visibility predicate $V(\mathbf{z}, \mathbf{x}(t))$ that indicates if the point $\mathbf{z}$ is in the range of the sensor for the given the robot state $\mathbf{x}(t)$:

$$\mathbb{V}(t) = \left\{ \mathbf{z} \in \mathbb{R}^2 : V(\mathbf{z}, \mathbf{x}(t)) \right\}. \tag{2}$$



Assuming bounded-error knowledge of $\mathbf{u}$ and $\mathbf{y}$, i.e $\mathbf{u}(t) \in [\mathbf{u}](t)$ and $\mathbf{y}(t) \in [\mathbf{y}](t)$, let us define the set of admissible trajectories

$$\mathcal{T} = \{\mathbf{x} : \mathbb{R} \to \mathbb{R}^n \mid \forall t,\ \dot{\mathbf{x}}(t) \in \mathbf{f}(\mathbf{x}(t), [\mathbf{u}](t)),\ \mathbf{g}(\mathbf{x}(t)) \in [\mathbf{y}](t)\}.$$

Ideally, $\mathbb{M}$ can be bracketed between the *guaranteed covered zone* (also called *clear zone*)

$$\mathbb{M}^{\forall} = \left\{ \mathbf{z} \in \mathbb{R}^2 \mid \forall \mathbf{x} \in \mathcal{T},\ \exists t,\ V(\mathbf{z}, \mathbf{x}(t)) \right\}, \tag{3}$$

and the *possibly covered zone* (also called *non-dark zone*)

$$\mathbb{M}^{\exists} = \left\{ \mathbf{z} \in \mathbb{R}^2 \mid \exists \mathbf{x} \in \mathcal{T},\ \exists t,\ V(\mathbf{z}, \mathbf{x}(t)) \right\}. \tag{4}$$

$\mathbb{M}^{\exists} \backslash \mathbb{M}^{\forall}$ is called the *penumbra* (the zone which is not known to have been covered or not due to uncertainty).

Approach

A fast approach to compute an inner approximation of $\mathbb{M}^{\forall}$ and an outer approximation of $\mathbb{M}^{\exists}$ has been presented in [2]. It consists in first contracting the tube $[\mathbf{x}](t)$ with the constraints of Eq. 1. Then, the covered zone interval is obtained as the union of visible zone intervals, using a set-inversion method, and symbolic interval arithmetic to efficiently deal with the penumbra. However, since it only consider visible zones at each time independently, trajectory smoothness information is lost, which leads to a very pessimistic characterization of the covered zone. An improved approach to deal with line scanning sensors, like laser or side-scan sonar, is presented in [3].

Using Minkowski difference in the translation invariant case

In most scenarios, the space to cover is a subspace corresponding to several components of the robot state (e.g latitude and longitude coordinates). Moreover, since the sensor is attached to the robot body, there is a tight relation between translation of the robot and translation of the visible zone.

Let's split the robot state vector components in two parts such that $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)^T$, where $\mathbf{x}_a$ corresponds to the part of the robot state that coincides with the sensed space. The visibility predicate is said to be translation invariant if

$$\forall \mathbf{a} \in \mathbb{R}^p,\ V(\mathbf{z} + \mathbf{a}, \mathbf{x} + \begin{pmatrix} \mathbf{a} \\ \mathbf{0}_{n-p} \end{pmatrix}) = V(\mathbf{z}, \mathbf{x}). \tag{5}$$

In other words, a translation in the robot state results in the same translation of the visible zone:

$$\mathbb{V}(\mathbf{x} + \begin{pmatrix} \mathbf{a} \\ \mathbf{0}_{n-p} \end{pmatrix}) = \mathbf{t}_{\mathbf{a}}\mathbb{V}(\mathbf{x}). \tag{6}$$

Now, let's find a set of trajectories $\mathcal{T}^*$ and a domain $\mathbb{A} \subset \mathbb{R}^q$ such that $\mathcal{T} \subset \mathcal{T}^{[]}$, where $\mathcal{T}^{[]}$ is the set of trajectories generated from

translations with a vector $(\mathbf{a}, \mathbf{0}_{n-p})^T$, $\mathbf{a} \in \mathbb{A}$ of the trajectories in $\mathcal{T}^*$:

$$\mathcal{T}^{[]} = \{\mathbf{x}^*(\cdot) + \begin{pmatrix} \mathbf{a} \\ \mathbf{0}_{n-p} \end{pmatrix}, \mathbf{x}^*(\cdot) \in \mathcal{T}^*, \mathbf{a} \in \mathbb{A}\}.$$

From (6) and (3), if the visibility predicate is translation invariant, the guaranteed covered zone for the set of trajectories $\mathcal{T}^{[]}$ is given by

$$\mathbb{M}^{\forall}_{\mathcal{T}^{[]}} = \mathbb{M}^{\forall}_{\mathcal{T}^*} \ominus -\mathbb{A},$$

where $\ominus$ denotes the Minkowski difference of two sets. And since $\mathcal{T} \subset \mathcal{T}^{[]}$, we have $\mathbb{M}^{\forall} \supset \mathbb{M}^{\forall}_{\mathcal{T}^{[]}}$, which means $\mathbb{M}^{\forall}_{\mathcal{T}^{[]}}$ can be used as a lower bound of the guaranteed covered zone.

Practical implementation consists in first computing an inner subpaving of the guaranteed covered zone with $\mathcal{T}^*$. If $\mathcal{T}^*$ can be well approximated by a tube, the existing fast method will have less pessimism than with the full trajectory set. The obtained domain is then eroded with $-\mathbb{A}$ to get an inner approximation of the guaranteed zone. Choosing A as a box enables very fast computation.

Results and comparison with previous methods will be presented on different test-cases.

References

[1] M. Couillard and D. P. Williams, Performance assessment of an adaptive AUV survey, *in International Conference on Detection and Classification of Underwater Targets,* p. 158, 2012.

[2] V. Drevelle, L. Jaulin and B. Zerr, Guaranteed Characterization of the Explored Space of a Mobile Robot by Using Subpavings, *In proceedings of NOLCOS 2013*, p. 44–49, 2013.

[3] B. Desrochers and L. Jaulin, "Computing a guaranteed approximation of the zone explored by a robot," IEEE Transactions on Automatic Control, 2016.

Guaranteed SLAM
An Interval Approach

Mohamed Mustafa[1], Eduard Codres[1], Nicolas Delanoue[2] and Alexandru Stancu[1]

[1] The University of Manchester, Faculty of Engineering and Physical Sciences, Oxford Road, Manchester, United Kingdom
{mohamed.mustafa, eduard.codres, alexandru.stancu}@manchester.ac.uk
[2] Laboratoire Angevin de Recherche en Ingénierie des Systèmes
The University of Angers, Angers, France
nicolas.delanoue@univ-angers.fr

Keywords: Nonlinear models, Real analysis, SLAM convergence, Interval Methods

Introduction

This paper proposes a new approach, *interval Simultaneous Localization and Mapping* (i-SLAM), which addresses the robotic mapping problem in the context of interval methods, where the robot sensor noise is assumed bounded. With no prior knowledge about the noise distribution or its probability density function, we derive and present necessary conditions to guarantee the map convergence even in the presence of nonlinear observation and motion models.

Concept

To describe the SLAM problem mathematically, consider the robot pose $\mathbf{s}_k$ at timestep k, and the landmark position $\mathbf{m}_i$, where $i \in \{1, \ldots, n_l\}$ and n_l is the number of observed landmarks. The robot *motion* and *observation* models are defined as follows, respectively:

$$\mathbf{s}_k = \mathbf{h}\left(\mathbf{s}_{k-1}, \mathbf{u}_k\right) + \mathbf{q}_k, \tag{1}$$

$$\mathbf{z}_{i,k} = \mathbf{g}\left(\mathbf{m}_i, \mathbf{s}_k\right) + \mathbf{r}_{i,k}, \tag{2}$$

where $\mathbf{h}$ and $\mathbf{g}$ are generally nonlinear functions, and $\mathbf{q}_k$ and $\mathbf{r}_{i,k}$ represent the additive noises associated with the motion and observation, respectively.

Assume that the robot motion uncertainty $\mathbf{q}_k$ in (1) is bounded by a box $[\mathbf{q}]$ such that $\forall k > 0, \mathbf{q}_k \in [\mathbf{q}]$, and let observation model noise in (2) be bounded such that $\mathbf{r}_k \in [\mathbf{r}]$. Then, the SLAM problem can be seen as a constraint satisfaction problem (CSP) where the set of variables are the SLAM unknown parameters and they are contained in the vector $\mathbf{x}$ as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{m}_1^T & \dots & \mathbf{m}_{n_l}^T & \mathbf{s}_0^T & \dots & \mathbf{s}_{n_f}^T \end{bmatrix}^T, \tag{3}$$

where n_f is the number of timesteps. Note that $\mathbf{u}_k$ and $\mathbf{z}_{i,k}$ are vector values obtained from the sensors. The constraints that govern the variables in $\mathbf{x}$ are derived directly from the robot motion model and the observation model as follows:

$$\mathbf{s}_k - \mathbf{h}\left(\mathbf{s}_{k-1}, \mathbf{u}_k\right) \in [\mathbf{q}], \tag{4}$$

$$\mathbf{z}_{i,k} - \mathbf{g}\left(\mathbf{m}_i, \mathbf{s}_k\right) \in [\mathbf{r}], \tag{5}$$

for all $i \in \{1, \dots, n_l\}$, and $k \in \{1, \dots, n_f\}$.

The *interval SLAM* (i-SLAM) is an algorithm that solves the SLAM problem using interval methods. Contractors are examples of interval methods that are capable of solving nonlinear CSPs with large number of variables.

Define the *measurement space* to include all observations and control inputs, e.g. $\mathbf{z}_{i,k}$ and $\mathbf{u}_k$ for all $i \in \{1, \dots, n_l\}$, and $k \in \{1, \dots, n_f\}$. Also define the *parameter space* to include all unknown SLAM variables in $\mathbf{x}$ as shown in (3). Then, provided that there is a bijective map [1] from the parameter space to measurement space, at the limit, the solution obtained by contractor is comparable to that obtained by SIVIA algorithm [2], with a considerable reduction in computational cost. From a practical point of view, the bijection assumption from the parameter space to the measurement space is expected to hold true at the limit especially when the robot traverses a cyclic environment with distinct landmarks [3]. Thus, using contractors is sufficient to solve,

in polynomial time [2], the SLAM problem in the presence of a large number of observations.

Algorithm 1 summarizes the i-SLAM procedure for known data association, where U_{n_f} and Z_{n_f} are the sets of all control inputs and all observations, respectively.

Algorithm 1 i-SLAM with known data association

1: **function** I-SLAM($[\mathbf{x}], U_{n_f}, Z_{n_f}, [\mathbf{q}], [\mathbf{r}]$)
2: **while** $[\mathbf{x}]$ is not fixed-point **do**
3: **for all** $k \in \{1, \ldots, n_f\}$ **do**
4: **for all** $i \in \{1, \ldots, n_l\}$ **do**
5: $[\mathbf{y}_1] \leftarrow [\mathbf{q}] \cap ([\mathbf{s}_k] - [\mathbf{h}]([\mathbf{s}_{k-1}], \mathbf{u}_k))$
6: $[\mathbf{s}_k] \leftarrow [\mathbf{s}_k] \cap ([\mathbf{h}]([\mathbf{s}_{k-1}], \mathbf{u}_k) + [\mathbf{y}_1])$
7: $[\mathbf{s}_{k-1}] \leftarrow [\mathbf{s}_{k-1}] \cap \left(\left[\mathbf{h}^{-1}_{\mathbf{s}_{k-1}}\right]([\mathbf{s}_k] - [\mathbf{y}_1], \mathbf{u}_k)\right)$
8: $[\mathbf{y}_2] \leftarrow [\mathbf{r}] \cap (\mathbf{z}_{i,k} - [\mathbf{g}]([\mathbf{m}_i], [\mathbf{s}_k]))$
9: $[\mathbf{m}_i] \leftarrow [\mathbf{m}_i] \cap \left(\left[\mathbf{g}^{-1}_{\mathbf{m}_i}\right](\mathbf{z}_{i,k} - [\mathbf{y}_2], [\mathbf{s}_k])\right)$
10: $[\mathbf{s}_k] \leftarrow [\mathbf{s}_k] \cap \left(\left[\mathbf{g}^{-1}_{\mathbf{s}_k}\right](\mathbf{z}_{i,k} - [\mathbf{y}_2], [\mathbf{m}_i])\right)$
11: **return** $[\mathbf{x}]$

References

[1] W. Rudin, *Functional analysis*, 2nd ed., ser. International Series in Pure and Applied Mathematics. New York: McGraw-Hill Inc., 1991.

[2] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*. Sprinter, ISBN: 978-1852332198, 2001.

[3] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. The MIT Press, ISBN: 978-0262015356, 2011.

A new method for computing an inner and outer approximation for guaranteed integration with uncertain initial conditions.

Alexandru Stancu[1], Eduard Codres [1] , Mario Martinez Guerrero[1] and Vicenç Puig[2].

[1] The University of Manchester, Faculty of Engineering and Physical Sciences, Oxford Road, Manchester, United Kingdom
{alexandru.stancu, eduard.codres, mario.martinezguerrero}@manchester.ac.uk
[2] Institut de Robòtica i Informàtica Industrial (CSIC-UPC). C/. Llorens i Artigas, 4-6, 08028 Barcelona, Spain
vicenc.puig@upc.edu

Keywords: Invariant set, Interval methods, Euler method, Viability kernel

Introduction

One of the most important issues in control is determining the stability of a system. Since the 1960's, Lyapunov-based methods have been developed to determine the stability of linear and nonlinear systems. However, when the system is nonlinear, and the initial condition is not known exactly, or when some uncertainties occur, stability analysis is challenging, and no reliable methods have been developed. In this paper, we propose a new method to bracket a capture tube. Then we will show how this method can be applied for computing an inner and outer approximation of the Viability Kernel.

In this paper, we consider the characterisation of invariant sets for nonlinear and continuous systems of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$. In the literature, there are two classes of methods to compute invariant sets. The first class of methods uses the concept of guaranteed integration

[1]. Starting from an initial box $[\mathbf{x}](0)$, guaranteed integration methods compute a box-valued function $[\mathbf{x}](t)$ or tube, which contain all true trajectories of the nonlinear system. The second class of methods is based on Lyapunov theory and do not require any integration. These methods approximate the dynamics of a system using an energy function [2] that is decreasing with time. Both classes of methods have some drawbacks which make them difficult or even impossible to apply for any nonlinear system. In case of guaranteed integration methods the main drawbacks are a) the interval integration is too conservative; b) because is too conservative the initial box must be divided into many small boxes increasing in this way the computational time; hence these methods cannot be used in case of higher dimensional systems; c) after the interval integration is difficult to obtain the invariant set which contains all intermediate steps. On the other hand, in case of Lyapunov methods, the main drawback is that we have to know a-priori a Lyapunov function which is only possible for a small class of systems.

Basic properties

In this paper, we propose a novel method to compute the invariant sets. Moreover, we are able to compute an outer and an inner approximation of the invariant set which will bracket in this way the real invariant set. The proposed method is a mixed method in the sense that it is a combination of trajectory propagation using a simple Euler method and interval-based methods. In this way, we can avoid the main drawbacks of guaranteed integration and use trajectory propagation instead. However, the trajectory propagation using Euler is not guaranteed, and we will combine it with interval-based methods in order to compute the inner and outer approximation of the invariant set. The proposed method is in discrete time and consists of a prediction-correction algorithm at every time instant. The prediction step consists on trajectory propagation using Euler, and the correction step is based on interval methods which provide a guaranteed method to compute the invariant set. In this way, the proposed method avoids

guaranteed integration which is susceptible to wrapping effect [3], but we use trajectory propagation based on Euler method to compute an initial guess for a Lyapunov function. The initial guess is then corrected using interval-based methods in order to obtain the inner and outer approximation for the invariant set.

Capture tubes and cross out condition

We define the capture tube for all the states starting from an initial set $\mathcal{D}_0$. A tube $\mathbb{G}$ is an interval of trajectories and it can be represented using inequalities:

$$\mathbb{G}(t) = \{\mathbf{x}, \mathbf{g}(\mathbf{x}, t) \leq 0\} \tag{1}$$

A tube is said to be a *capture tube* if the fact that $\mathbf{x}(t) \in \mathbb{G}(t)$ implies that $\mathbf{x}(t + t_1) \in \mathbb{G}(t + t_1)$for all $t_1 > 0$. The following theorem [4] will be used to find an arbitrary small capture tube for the solution set of our dynamical system.

Theorem 1. ***(Capture Tubes)*** *Consider a tube* $\mathbb{G}(t) = \{\mathbf{x}, \mathbf{g}(\mathbf{x}, t) \leq 0\}$ *where* $\mathbf{g} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ *If the cross out condition:*

$$\begin{cases} \dfrac{\partial g_i}{\partial \mathbf{x}}(\mathbf{x}, t).\mathbf{f}(\mathbf{x}) + \dfrac{\partial g_i}{\partial t}(\mathbf{x}, t) \geq 0 \\ g_i(\mathbf{x}, t) = 0 \\ \mathbf{g}(\mathbf{x}, t) \leq 0 \end{cases} \tag{2}$$

is inconsistent for all $\mathbf{x}$*, all* $t > 0$*, and all* $i \in \{1, ..., m\}$ *then* $\mathbb{G}(t)$ *is a capture tube for the time-dependent system* $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$*.*

Because our dynamical system is time invariant, the capture tube will also be defined as time invariant:

$$\mathbb{G} : \{\mathbf{x}, \mathbf{g}(\mathbf{x}) \leq 0\} \tag{3}$$

As it can be seen from the above theorem, in order to find the capture tube for the solution set of the dynamical system starting from $\mathcal{D}_0$ we have to find a function $\mathbf{g}(\mathbf{x})$ for which the *cross out* condition is inconsistent.

Capture tube estimation using trajectory propagation

This method is fast and easy to use, but it only gives us an inner approximation of the solution set and we want to find an outer guaranteed approximation. While some guaranteed methods for finding this outer enclosure exist, they all suffer from the well-known wrapping effect which leads to instability. We will use trajectory propagation as an initial guess for the Lyapunov function. This initial guess can be used to construct the function $\mathbf{g}(\mathbf{x})$ which defines the capture tube. There are many methods than can interpolate a data set consisting of vertices using some function. But one critical requirement for our method is that the conditions of the capture tube theorem have to be fulfilled. One of the most general and promising methods to generate interpolating functions is to use RBF (radial basis function) networks. The main advantage of this approach is that the cross out condition can be included in the training process. We can then use the resulting RBF network function to define the capture tube. There are some drawbacks though:

- The training process can be slow and it can converge to local minima's.
- Depending on the complexity of the dynamical system a big number of RBF's might be required.
- Interval methods do not cope well with multiple occurrences of the variables. RBF networks rely on having a big number of basis functions which means multiple occurrences cannot be avoided.

Another approach is to approximate the capture tube using hyperplanes:

$$g_i(\mathbf{x}) = \mathbf{N}_i \times \mathbf{x} + d_i, \quad i \in \{1, ..., m\} \tag{4}$$

$\mathbf{N}_i$ - hyperplane normal;
d_i - distance from hyperplane to origin;

m - total number of hyperplanes defining the capture tube;

This is the method used in this paper. In 2D space the hyperplanes are segments, in 3D are triangles, in 4D are prisms and so on. In this way, in our approach, we are only interested to compute the boundary of the capture tube and not on the inside of the tube.

Main results

Case study I:Computing an outer and inner approximation for invariant sets

Capture tube generated for a 2D dynamical system System Equations (pendulum)

$$\begin{cases} \dot{x_1} = x2 \\ \dot{x_2} = -sin(x_1) - 0.65x_2 \end{cases} \tag{5}$$

Initial box:$[0.6, 0.8] \times [0.4, 0.6]$ The capture tube was computed for 6s.

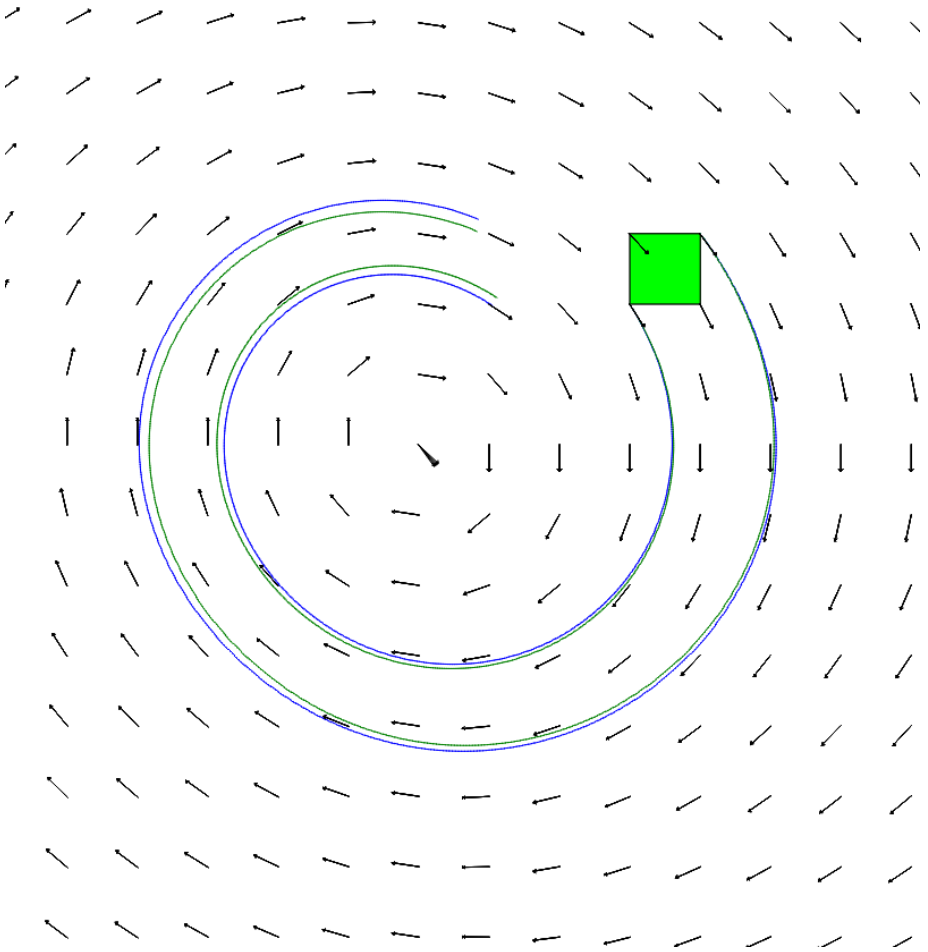


Figure 1: Box evolution in a 2D system

Capture tube generated for a 3D dynamical system System equations:

$$\begin{cases} \dot{x_1} = -x_2 x_3 + 1 \\ \dot{x_2} = x_1 x_3 - x_2 \\ \dot{x_3} = x_3^2(1 - x_3) \end{cases} \tag{6}$$

Initial box:$[0.6, 0.8] \times [0.4, 0.6]$ With red we represented the point used as initial condition for the trajectories propagation. They are on the sides of the initial cube where the trajectories are crossing the cube from inside to outside.

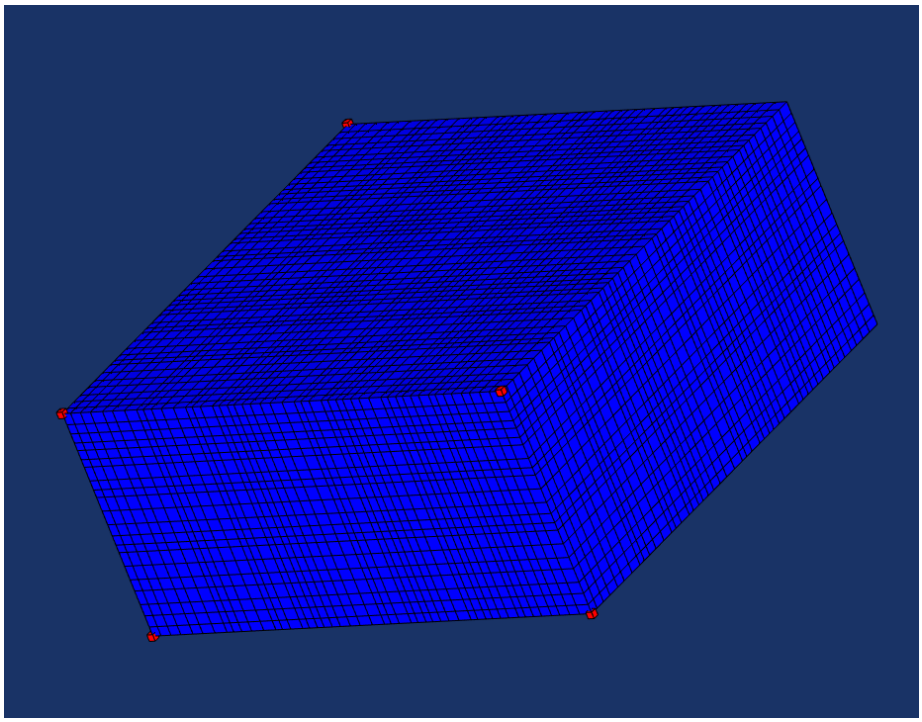


Figure 2: Initial Box for the 3D system

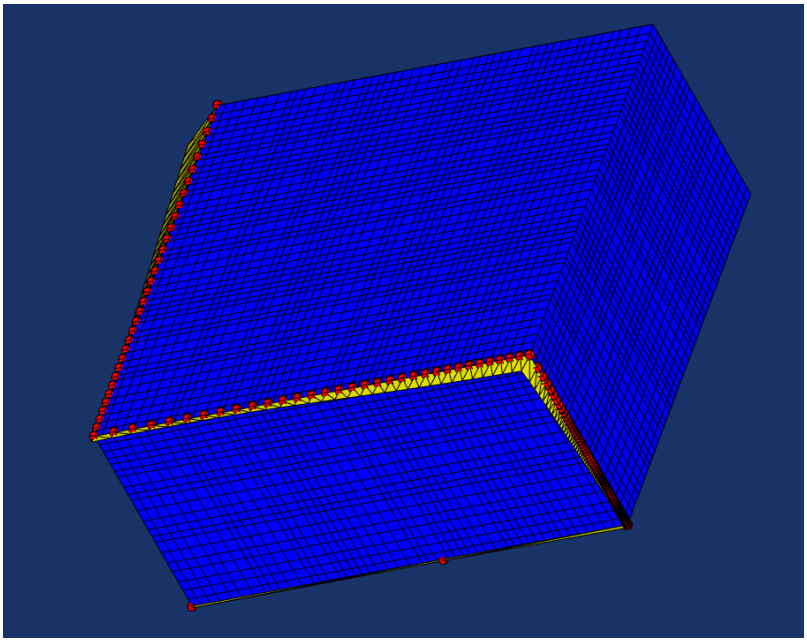


Figure 3: Tube computation

The figures below, show the inner and outter approximation of the capture tube from different perspectives.

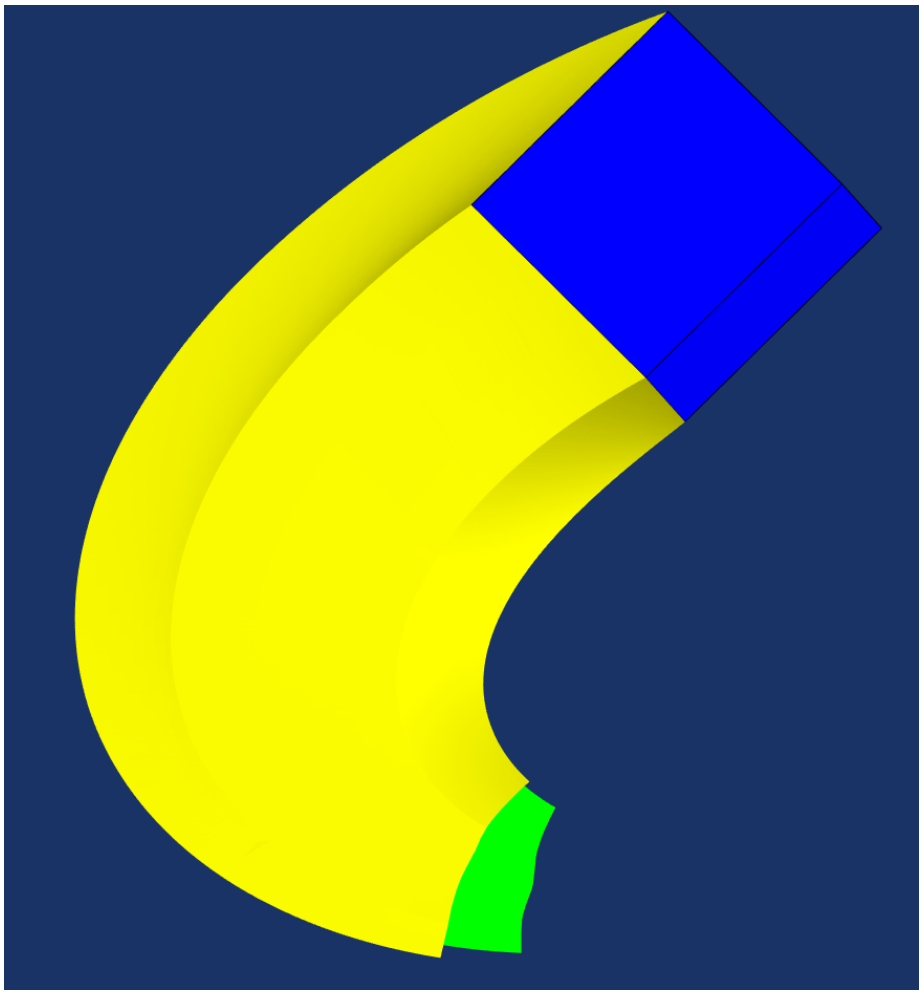


Figure 4: Different persepctives of the inner and outter approximation

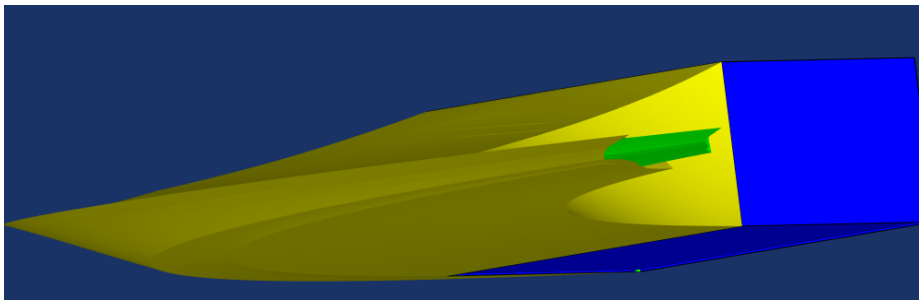


Figure 5: Different persepctives of the inner and outter approximation

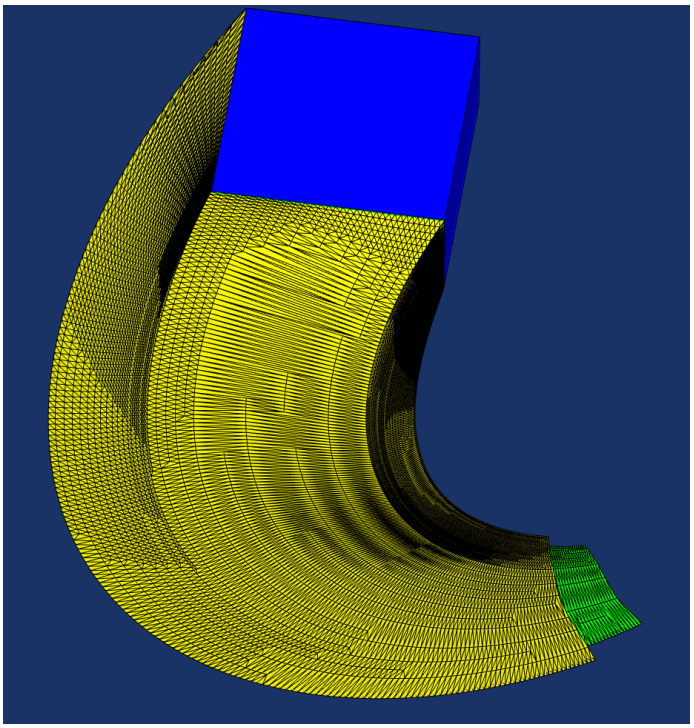


Figure 6: Different persepctives of the inner and outter approximation

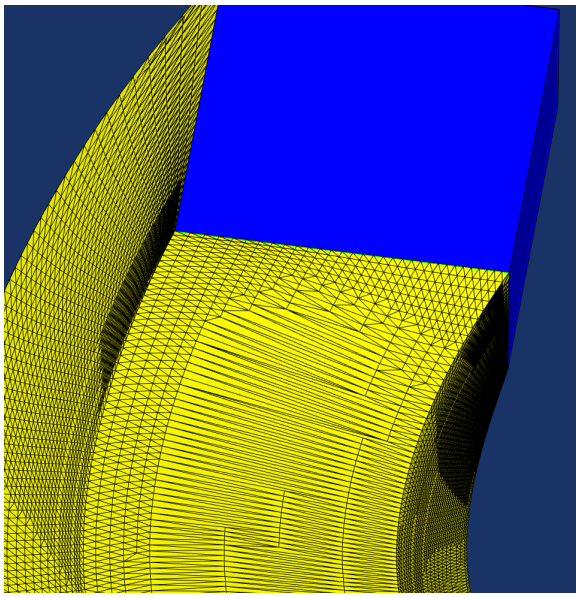


Figure 7: Different persepctives of the inner and outter approximation

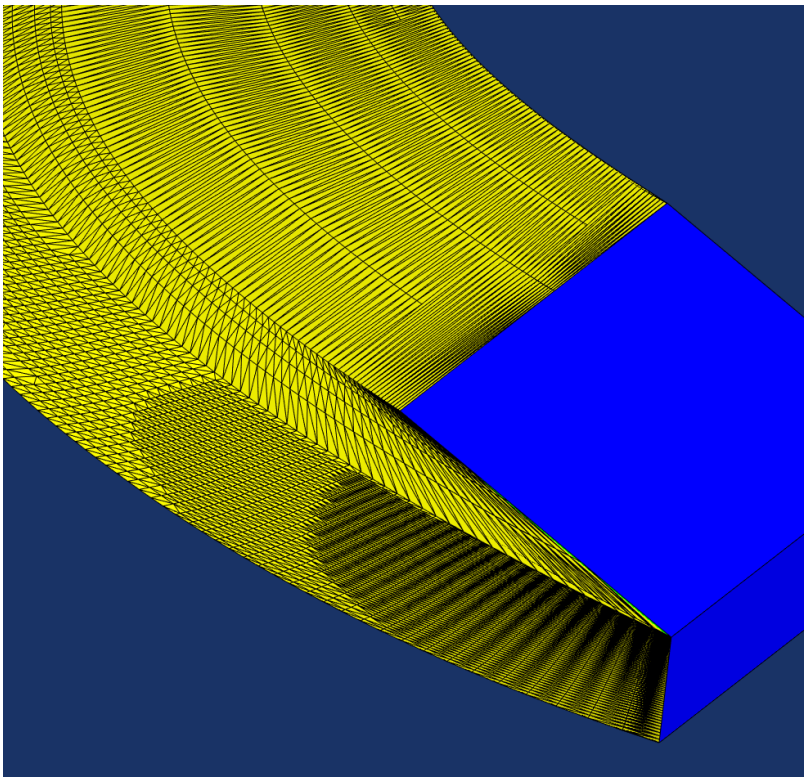


Figure 8: Different persepctives of the inner and outter approximation

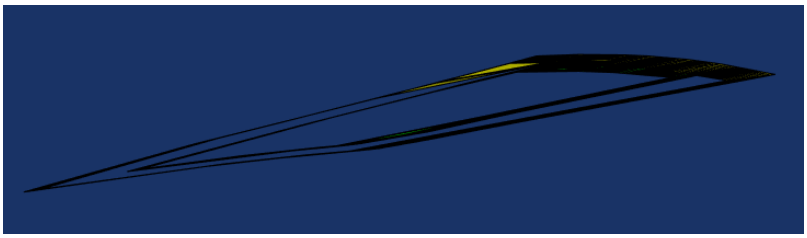


Figure 9: Cross Section to see the inner and outter approximations

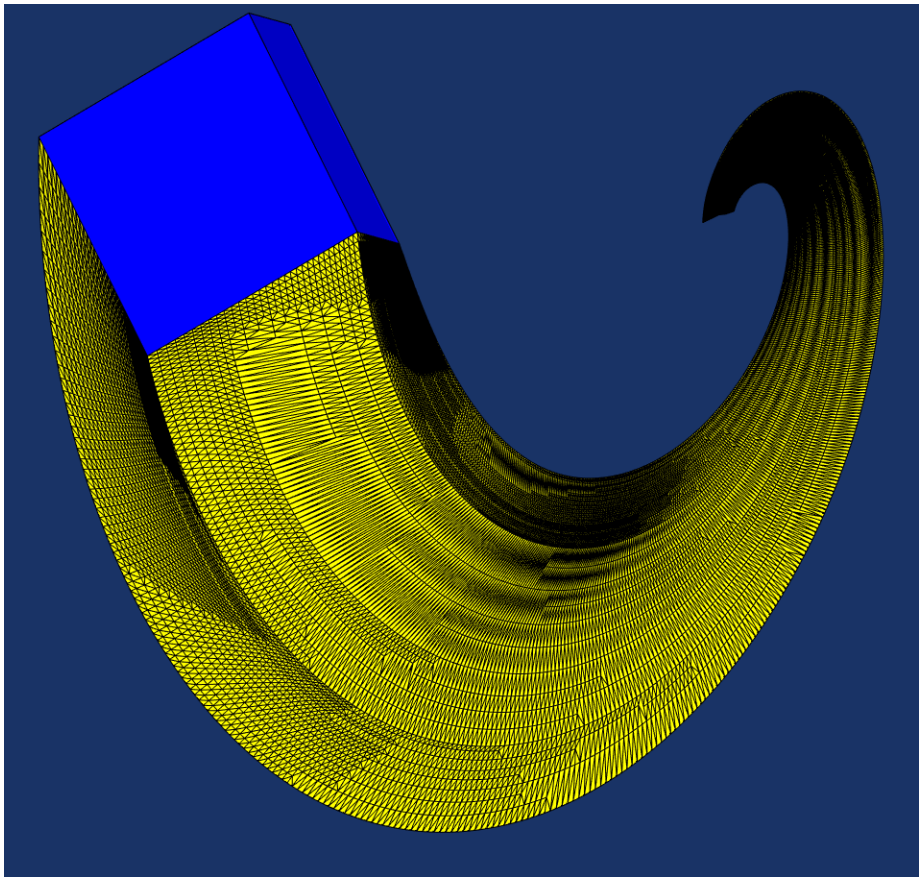


Figure 10: Different persepctives of the inner and outter approximation

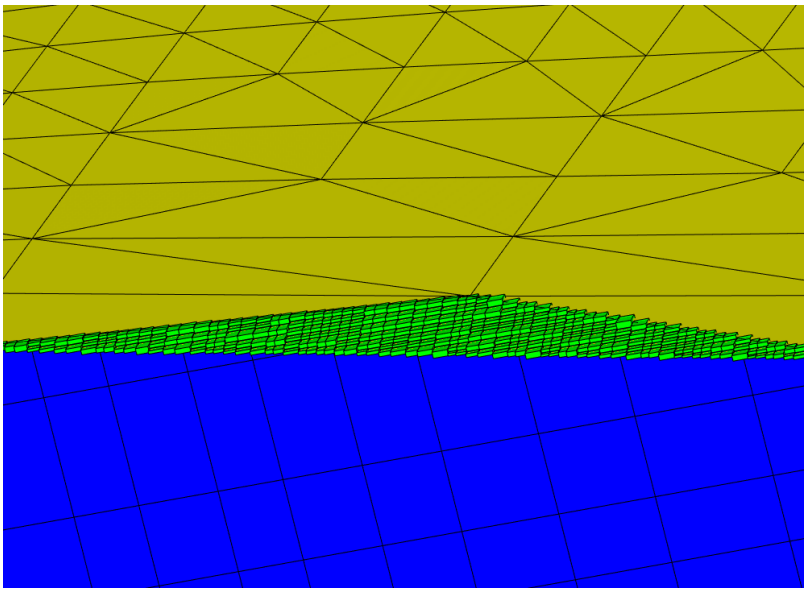


Figure 11: Zoom on some triangles at a particular time instance

Case study II:Computing an outer and inner approximation for a Viability kernel
This example is given for "Car on the hill" system proposed in [5]. The initial set K and with red are the segments which need to be adjusted such that to have at least one control which keeps the system inside $\mathbb{K}$.

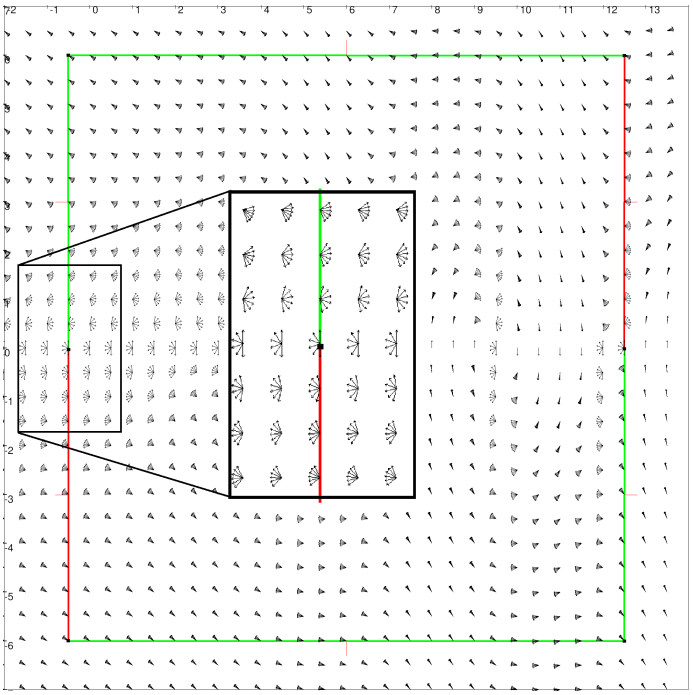


Figure 12: Vector field at the border for the first iteration

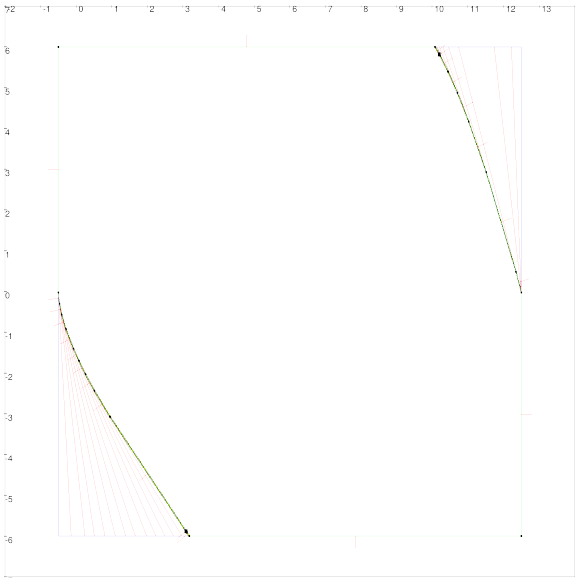


Figure 13: The segments are adjusted iteratively

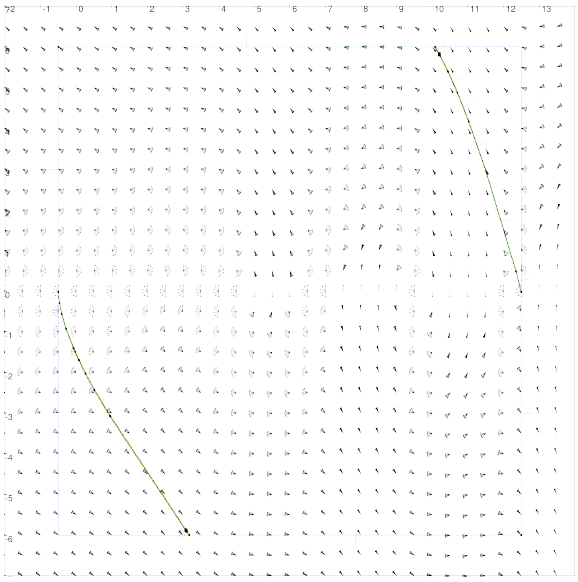


Figure 14: Viability Kernel obtained with the vector field

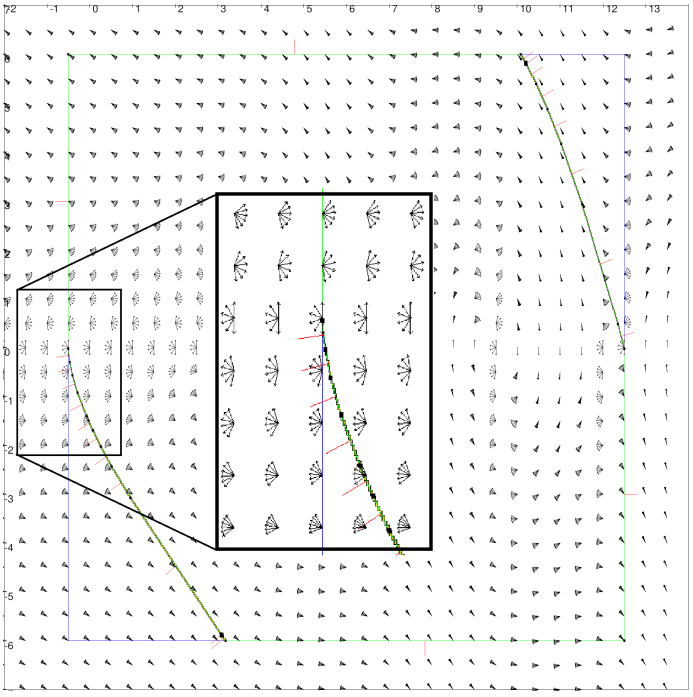


Figure 15: Boxes at the border of the Viability Kernel

References

[1] M. Berz, C. Bischof, G. Corliss, and G. A., Eds., Computational Differentiation: Techniques, Applications and Tools. Philadelphia, Penn.: SIAM, 1996.
[2] L. Jaulin and F. L. Bars, "An Interval Approach for Stability Analysis; Application to Sailboat Robotics," IEEE Transaction on Robotics, vol. 27, no. 5, 2012.
[3] Alexandru Stancu, Vicenç Puig, Pep Cugueró, Joseba Quevedo, "Benchmarking on Approaches to Interval Observation Applied to Robust Fault Detection", Global Optimisation and Constraint Satisfaction, Lecture notes in computer science, ISSN 0302-9743 , Volume 3478, 2005, Pages 171-192, Springer Berlin.
[4] A. Stancu, L. Jaulin, and A. Bethencourt. Stability analysis for time-dependent nonlinear systems ; an interval approach. Internal report, University of Manchester, 2015.
[5] D. Monnet, J. Ninin and L. Jaulin, Computing an inner and an outer approximation of viability kernels, Reliable Computing 22, 138-148, 2016.

Computing an Inner Approximation of the Viability Kernel Using Capture Tubes

Eduard Codres[1], Joaquim Blesa [2] , Mario Martinez Guerrero[1], Alexandru Stancu[1].

[1] The University of Manchester, Faculty of Engineering and Physical Sciences, Oxford Road, Manchester, United Kingdom
{eduard.codres, mario.martinezguerrero}@manchester.ac.uk
[2] Institut de Robòtica i Informàtica Industrial (IRI), CSIC-UPC, C/. Llorens i Artigas 4-6, 08028 Barcelona, Spain
jblesa@iri.upc.edu

Keywords: Viability kernel, Interval analysis, Guaranteed Integration.

Introduction

Viability theory [1] is a promising area of research for the design of reliable control systems in the presence of uncertainties and faults. In this work, following the ideas presented in [2], we propose a method to compute the inner and outer approximation of the viability kernel using interval analysis and guaranteed integration techniques.

Basic properties

Following the notation in [1] we will consider a dynamic system $\mathcal{S}$ defined by

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ u(t) &\in \mathbb{U} \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$, $\mathbb{U}$ is a compact subset of $\mathbb{R}^m$, $u \in \mathcal{U} = u : \mathbb{R}^+ \longmapsto \mathbb{U}$, $f : \mathbb{R}^n \times \mathbb{U} \longmapsto \mathbb{R}^n$ being f a continuous and locally Lipschitzian function bounded in $\mathbb{R}^n \times \mathbb{U}$ and φ is the flow map of $\mathcal{S}$ that computes the

reached state $\varphi(t, x_0, u)$ given an initial state $x_0 = x(t)$ and a control function $u(t)$.

Then, the viability kernel [1] in [2] is defined as

Definition 1 *Let $\mathcal{S}$ a system defined by Eq. (1) and let $\mathbb{K} \subseteq \mathbb{R}^n$ be a compact set. The viability kernel of $\mathbb{K}$ under $\mathcal{S}$, is the set $Viab\mathcal{S}(\mathbb{K})$ of initial states $x \in \mathbb{K}$ from which at least one evolution does not leave $\mathbb{K}$ for all $t \geq 0$. We have*

$$Viab\mathcal{S}(\mathbb{K}) = \{x_0 \in \mathbb{K} | \exists u \in \mathcal{U}, \forall t \geq 0, \varphi(t, x, u) \in \mathbb{K}\},$$

with the purpose of computing an inner approximation of the viability kernel using interval analysis we propose:

Proposition 1. *Given a system $\mathcal{S}$ defined by Eq. (1), an unknown initial state x_0 bounded by a box $[x_0]$ (i.e. $x_0 \in [x_0]$), an interval time horizon t_H and a control vector u in the time horizon t_H; the evolution of the state x of the system $\mathcal{S}$ can be bounded by a tube $\mathbb{T}_{\mathcal{S}}([x_0], t, u)$ such that*

$$\varphi(t, x_0, u) \in \mathbb{T}_{\mathcal{S}}([x_0], t, u) \quad \forall x_0 \in [x_0] \quad \forall t \in [0, t_H].$$

This Tube can be computed by discretizing Eq. (1) and using guaranteed integration techniques.

Proposition 2. *From the Tube $\mathbb{T}_{\mathcal{S}}([x_0], t, u)$ a boundary set $\mathbb{B}_{\mathcal{S}}([x_0], t_H, u)$ can be obtained, by using the final points of the tube at t_H (i.e. $\mathbb{T}_{\mathcal{S}}([x_0], t_H, u)$).*

The Tube $\mathbb{T}_{\mathcal{S}}([x_0], t, u)$ and the Boundary $\mathbb{B}_{\mathcal{S}}([x_0], t_H, u)$ satisfy the following conditions:

$$\varphi(t, x_0, u) \cap \mathbb{T}_{\mathcal{S}}([x_0], t, u) = \emptyset, \tag{2}$$
$$\varphi(t, x_0, u) \cap \mathbb{B}_{\mathcal{S}}([x_0], t_H, u) \neq \emptyset. \tag{3}$$

For a $\mathbb{R}^2$ system this can be depicted as the line segment between the two final points of the tube in Figure 1

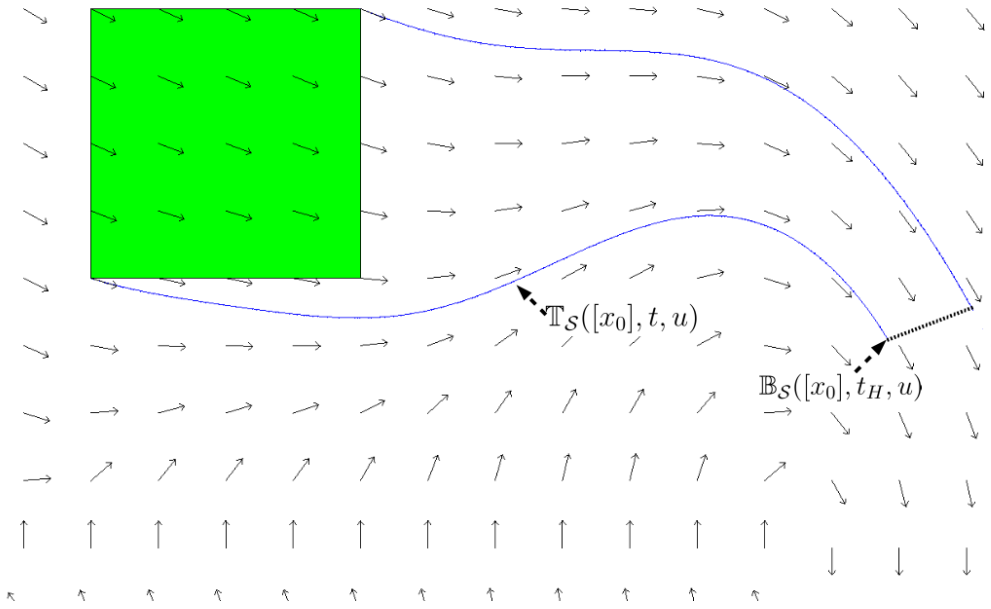


Figure 1: Integration Tube.

Proposition 3. *Given that the set $\mathbb{V}_{inner}$ is non-convex, if there exists a set $\mathbb{V}' \subset \mathbb{K}$ generated by the tube $\mathbb{T}_{\mathcal{S}}([x], t_H, u)$ and the set $\mathbb{V}_{inner}$, then the set $\mathbb{V}' \subset \mathbb{V}_{inner}$.*

Proof. Lets suppose:

$$\varphi(t, x, u) \cap (\mathbb{V}_{inner} \cup \mathbb{T}_{\mathcal{S}}([x], t, u)) = \emptyset, \quad \forall x \in \mathbb{V}' \tag{4}$$

Taking into account the properties of the system and proposition 2, $\forall x \in \mathbb{V}'$

$$\varphi(t, x, u) \cap \mathbb{V}_{inner} \neq \emptyset, , \tag{5}$$

$$or \quad \varphi(t, x, u) \cap \mathbb{T}_{\mathcal{S}}([x], t, u)) \neq \emptyset, , \tag{6}$$

$$or \quad \varphi(t, x, u) \cap \mathbb{V}' \neq \emptyset. \tag{7}$$

Where Eq. (5) and Eq. (6) contradicts Eq. (4), Eq. (7) states the posibility of an equilibrium point inside the set $\mathbb{V}'$. Therefore, $\mathbb{V}' \subset \mathbb{V}_{inner}$. Figure 2 depicts a graphical illustration for proposition 3 in a $\mathbb{R}^2$ system. ■

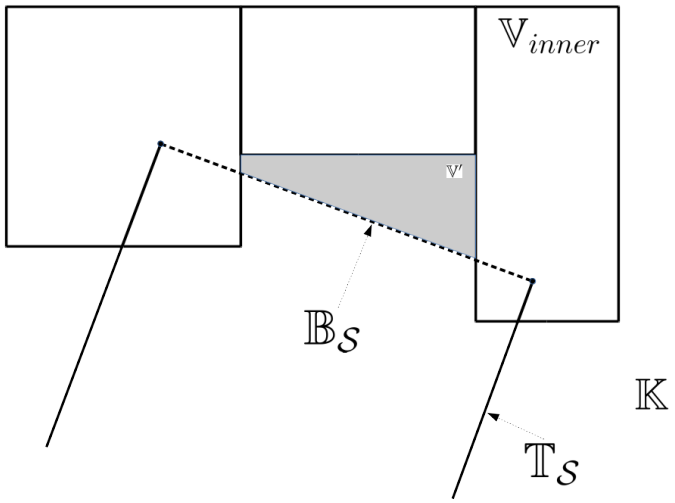


Figure 2: Case defined in Proposition 3.

Main results

Given an initial inner approximation of the viability kernel $\mathbb{V}_{inner}$, the tube $\mathbb{T}_{\mathcal{S}}([x_0], t, u)$ and its boundary set $\mathbb{B}_{\mathcal{S}}([x_0], t_H, u)$ defined in previous Section can be used to compute an inner approximation of the viability kernel $Viab\mathcal{S}(\mathbb{K})$ by means of algorithm 1 that follows the ideas proposed in [2]. Figures 4 and 3 depict the two different cases presented in algorithm 1.

Algorithm 1 Computation of an inner approximation of $Viab\mathcal{S}(\mathbb{K})$

Require: $\mathcal{S}, \mathcal{U}, \mathbb{K}, t_H$ and initial set $\mathbb{V}_{inner}$

$\mathbb{H} = \emptyset$ and $\mathbb{S} = \mathbb{K} \setminus \mathbb{V}_{inner}$
while $\mathbb{S} \neq \emptyset$ **do**
  for $[x_i] \in \mathbb{S}$ **do**
    Choose $u \in \mathcal{U}$
    if $\mathbb{T}_{\mathcal{S}}([x_i], t_H, u) \subseteq \mathbb{K}$ and $\mathbb{B}_{\mathcal{S}}([x_i], t_H, u) \subseteq \mathbb{V}_{inner}$ **then**
      $\mathbb{V}_{inner} := \mathbb{V}_{inner} \cup [x_i], \quad \mathbb{S} := \mathbb{S} \setminus [x_i]$
      Compute $\mathbb{V}_T = (\mathbb{T}_{\mathcal{S}}([x_i], t_H, u) \cap \mathbb{S})_{inner}$
      $\mathbb{V}_{inner} := \mathbb{V}_{inner} \cup \mathbb{V}_T, \quad \mathbb{S} := \mathbb{S} \setminus \mathbb{V}_T$
    end if
  end for
  Bisect boxes of $\mathbb{S}$
end while
return $\mathbb{V}_{inner}$ and $\mathbb{H}$

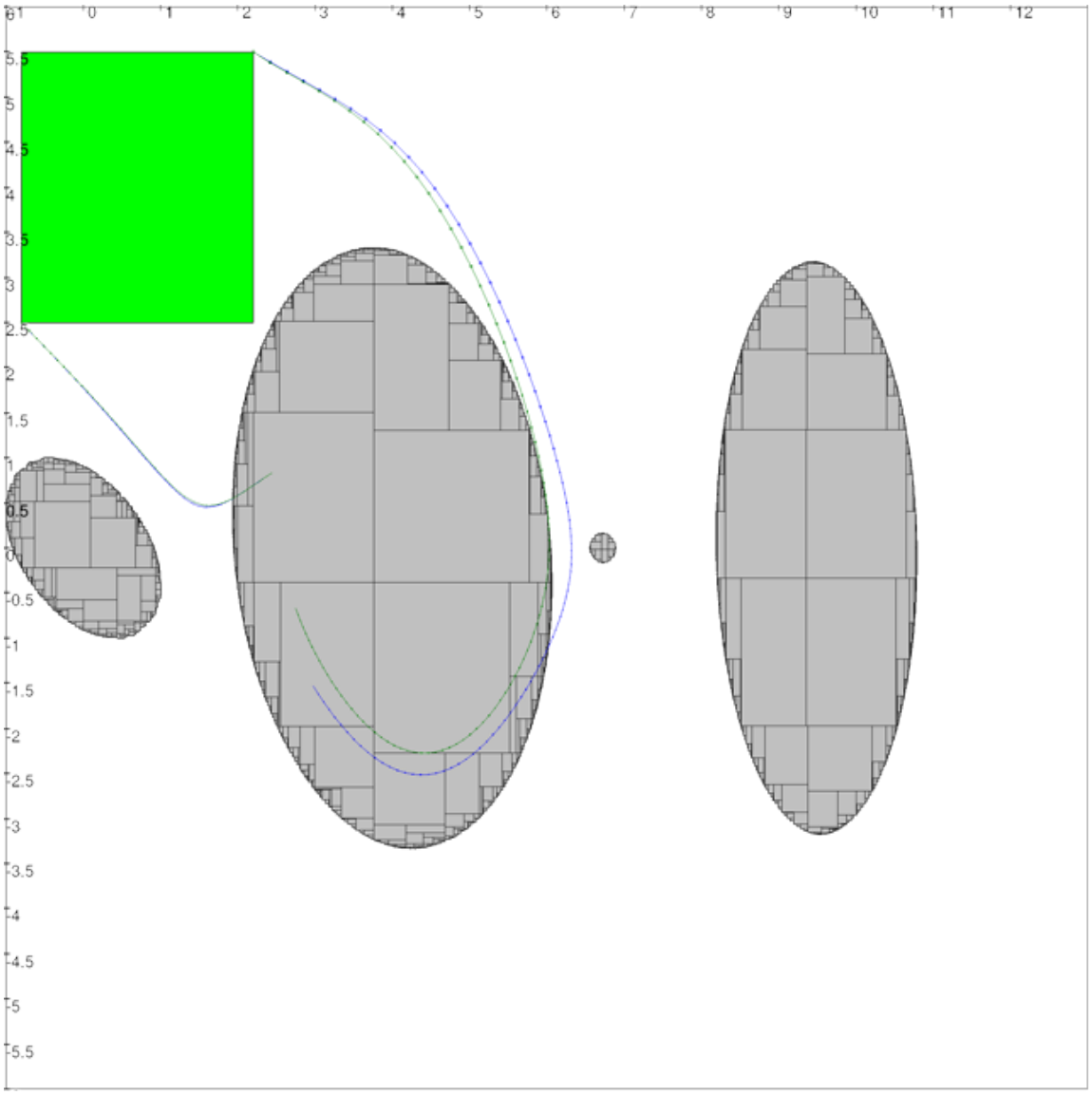


Figure 3: Box evolving towards the Viable Set

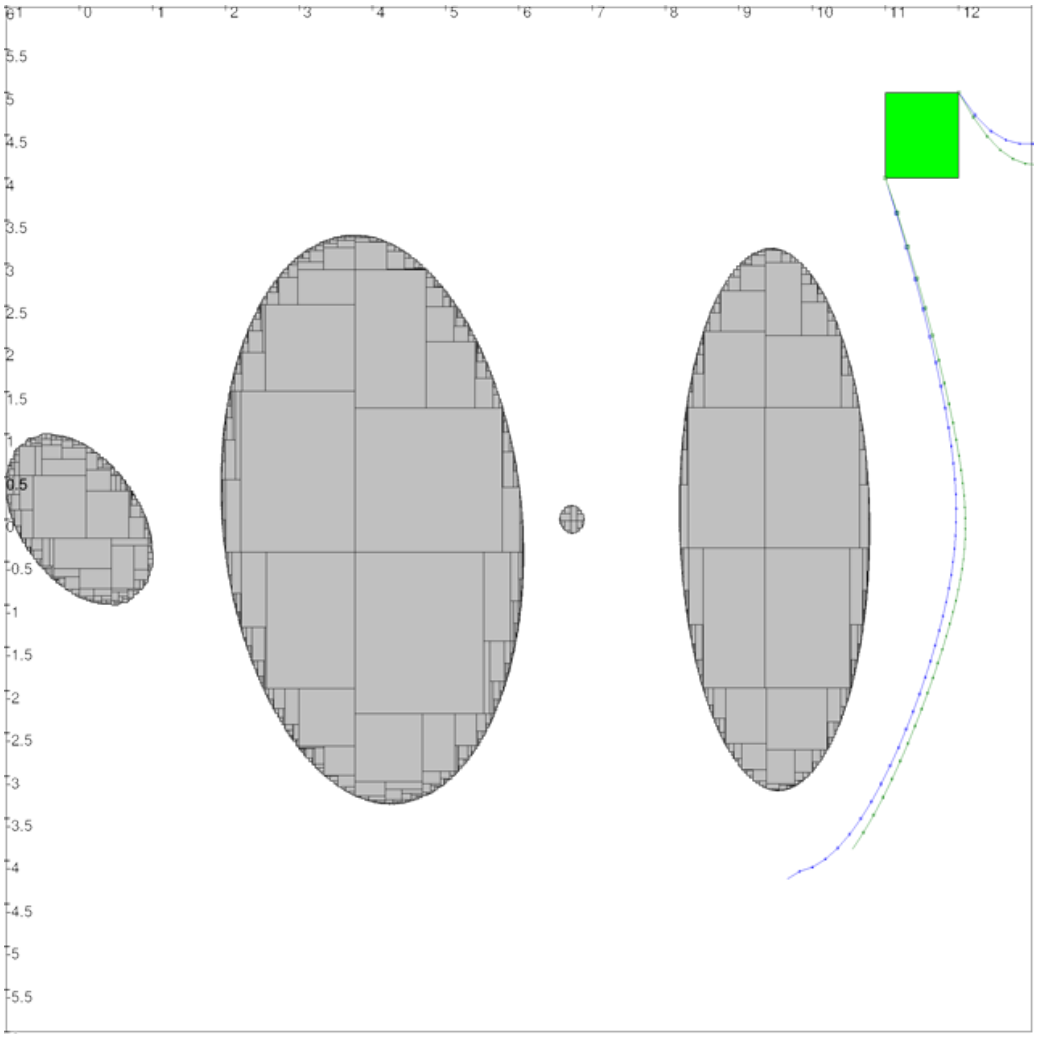


Figure 4: Box evolving towards the Non Viable Set

Acknowledgement

Thanks to the University of Manchester, Autonomous Research Theme for the support during the realization of the project.

This work has been partially funded by the Ministerio de Economía, Industria y Competitividad (MEICOMP) of the Spanish Government and FEDER through the project DEOCS (ref. DPI2016-76493-C3-3-R) and through the grant IJCI-2014-20801.

Thanks to the National Council of Science and Technology (CONACyT) for their support in the realization of this project.

References

[**1**] JP. Aubin, *Viability Theory*, Birkhauser, Boston, 1991.

[**2**] D. Monnet, J. Ninin and L. Jaulin, Computing an inner and an outer approximation of viability kernels, *Reliable Computing* 22, 138-148, 2016.