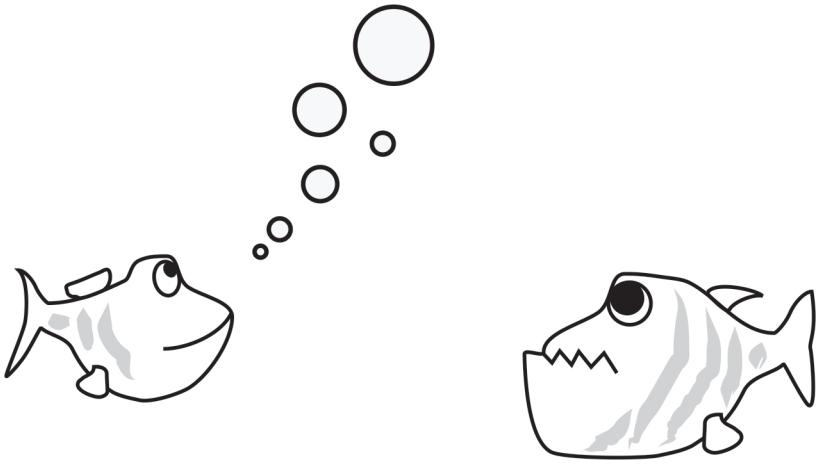
SWIM 2015

8^{th} Small Workshop on Interval Methods

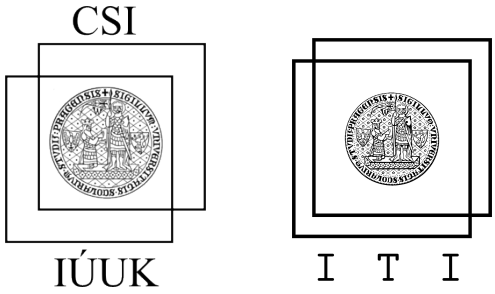


June 9 – 11, 2015, Prague

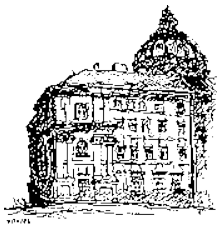
David Hartman, Milan Hladík, Jaroslav Horáček,
Luc Jaulin, Nacim Ramdani (eds.)

The 8^{th} Small Workshop on Interval Methods is organized and supported jointly by

- Computer Science Institute of Charles University,
- Centre of Excellence – Institute for Theoretical Computer Science, Charles University supported by GAČR as grant P202/12/G061 CE-ITI,
- Center for Discrete Mathematics - DIMATIA,



Most events are located in University historical building Malostranské namestí 25, Prague



SWIM 2015

Published by IUUK-ITI Series 2015-620
Centre of Excellence – Institute for Theoretical Computer Science,
Faculty of Mathematics and Physics, Charles University
Malostranské nám. 25, 118 00 Prague 1, Czech Republic

Published by MATFYZPRESS,
Publishing House of the Faculty of Mathematics and Physics, Charles University in Prague
Sokolovská 83, 186 75 Prague 8, Czech Republic
as the 487 th publication

First published, Prague 2015

ISBN 978-80-7378-293-1

Preface

This volume contains the Book of abstracts of The eighth Small Workshop on Interval Methods SWIM 2015. The tradition of SWIM workshops was set up in France in 2008, and since that time it is held annually by the effort of Luc Jaulin and Nacim Ramdani. The workshop joins people from different communities working with interval methods. Thus, it provides a unique opportunity to meet scientists from robotics, optimization, control, estimation, verification and other areas.

SWIM 2015 in Prague

The eighth issue of the Small Workshop on Interval Methods SWIM 2015 goes to Prague. Prague has a long term tradition in interval methods. Prof. Jiří Rohn, the father of interval analysis in the Czech Republic, has been working in this area since 1970's. He achieved great results in interval linear equation solving, regularity and eigenvalues of interval matrices and interval linear programming, among many others.

Interval community in Prague, however, is continuously expanding. Thus, we can informally speak of a 'Prague school of interval analysis'. One of the young excellent Prague intervallers is the invited plenary speaker Michael Černý, giving a lecture on statistics with interval data.

With 37 accepted abstracts and 43 participants, SWIM 2015 is one of the largest SWIM workshop and cannot be called 'small' any more.

Organizing SWIM 2015 in Prague

First, I wish to thank to all our sponsors since it would not be possible to organize the workshop without their support. In particular, the donations by CE-ITI (Center of Excellence - Institute for Theoretical Computer Science) and DIMATIA were fundamental for the success of SWIM.

My further thanks go to all my colleagues that help me a lot in preparations of the workshop. The experience of Luc Jaulin and Nacim Ramdani in organizing SWIM workshops guaranteed an easy to follow course. No doubt, there wouldn't be SWIM'15 without a great help of my co-organizers Jaroslav Horáček and David Hartman. Also the assistance of my students such as Jan Bok and Elif Garajová was very valuable. And most importantly, the success of each meeting depends on You, the participants.

Milan Hladík

Organizing Committe

Milan Hladík, Charles University in Prague

Luc Jaulin, ENSTA Bretagne

Nacim Ramdani, Université d Orléans

Local Organizing Committe

Jan Bok,

David Hartman,

Milan Hladík,

Jaroslav Horáček

Contents

Abstracts

Intervals of Sign Regular Matrices

[1]Mohammad Adm, [2]Jürgen Garloff, and [1]Jihad Titi

[1] Department of Mathematics and Statistics, University of Konstanz,
D-78464 Konstanz, Germany
mjamathe@yahoo.com, jihadtiti@yahoo.com
[2] Faculty of Computer Sciences, University of Applied Sciences/ HTWG Konstanz,
D-78462 Konstanz, Germany
garloff@htwg-konstanz.de

Keywords: Interval matrix, checkerboard ordering, totally nonnegative matrix, sign regular matrix.

We say that a class $\mathcal{C}$ of n-by-n matrices possesses the *interval property* if for any n-by-n interval matrix $[A] = [\underline{A}, \overline{A}] = ([\underline{a}_{ij}, \overline{a}_{ij}])_{i,j=1,\ldots,n}$ the membership $[A] \subseteq \mathcal{C}$ can be inferred from the membership to $\mathcal{C}$ of a specified set of its vertex matrices; here a *vertex matrix of* $[A]$ is a real matrix $B = (b_{ij})_{i,j=1,\ldots,n}$ with $b_{ij} \in \{\underline{a}_{ij}, \overline{a}_{ij}\}$ for all $i, j = i, \ldots, n$. Examples of such classes include the

- M-matrices or, more generally, inverse-nonnegative matrices [9], where only the bound matrices $\underline{A}$ and $\overline{A}$ are required to be in the class;
- inverse M-matrices [8], where all vertex matrices are needed;
- positive definite matrices [3], [12], where a subset of cardinality 2^{n-1} is required (here only symmetric matrices in $[A]$ are considered).

A class of matrices which in the nonsingular case are somewhat related to the inverse nonnegative matrices are the totally nonnegative matrices. A real matrix is called *totally nonnegative* if all its minors are nonnegative. Such matrices arise in a variety of ways in mathematics and its applications, e.g., in differential and integral equations,

numerical mathematics, combinatorics, statistics, and computer aided geometric design. For background information we refer to the recently published monographs [4], [11]. The second author posed in 1982 the conjecture that the set of the nonsingular totally nonnegative matrices possesses the interval property, where only two vertex matrices are involved [5], see also [4, Section 3.2] and [11, Section 3.2]. The two vertex matrices are the bound matrices with respect to the checkerboard ordering which is obtained from the usual entry-wise ordering in the set of the square matrices of fixed order by reversing the inequality sign for each entry in a checkerboard fashion. In our talk we apply the Cauchon algorithm [2] (also called deleting derivation algorithm [7] and Cauchon reduction algorithm [10]) to settle the conjecture. We also obtain the result that a fixed zero-nonzero pattern of the minors stays unchanged through an interval of nonsingular totally nonnegative matrices.

As a generalization of the totally nonnegative matrices we further consider *sign regular* matrices, i.e., matrices with the property that all their minors of fixed order have one specified sign or are allowed also to vanish. We identify some subclasses of the sign regular matrices which exhibit the interval property. The subclasses which require to check only two vertex matrices include the following sets (here it is understood that the two bound matrices have the same signature of their minors):

- the strictly sign regular matrices, i.e., the matrices with the property that all their minors of fixed order have one (strict) specified sign;
- the nonsingular almost strictly sign regular matrices, a class in between the nonsingular sign regular matrices and the strictly sign regular matrices;
- the tridiagonal nonsingular sign regular matrices;
- the nonsingular totally nonpositive matrices, i.e., the matrices with

the property that all their minors are nonpositive.

In some instances, the assumption of nonsingularity can be somewhat relaxed. These results lead us to the following new conjecture: Assume that the two bound matrices with respect to the checkerboard ordering are nonsingular and sign regular; then all matrices lying between the two bound matrices are nonsingular and sign regular, too. It was shown in [6] that the conclusion is true if we consider instead of the two bound matrices a set of vertex matrices with the cardinality of at most 2^{2n-1} (n being the order of the matrices).

References and Literature for Further Reading

[**1**] M. Adm and J. Garloff, Intervals of totally nonnegative matrices, *Linear Algebra Appl.*, 439:3796–3806, 2013.

[**2**] M. Adm and J. Garloff, Improved tests and characterizations of totally nonnegative matrices, *Electron. Linear Algebra*, 27:588–610, 2014.

[**3**] S. Bialas and J. Garloff, Intervals of P-matrices and related matrices, *Linear Algebra Appl.*, 58:33–41, 1984.

[**4**] S. M. Fallat and C. R. Johnson, *Totally Nonnegative Matrices*, Princeton Series in Applied Mathematics, Princeton University Press, Princeton and Oxford, 2011.

[**5**] J. Garloff, Criteria for sign regularity of sets of matrices, *Linear Algebra Appl.*, 44:153–160, 1982.

[**6**] J. Garloff, Vertex Implications for Totally Nonnegative Matrices, in *Total Positivity and its Applications*, M. Gasca and C.A. Micchelli, Eds., Kluwer Acad. Publ., Dordrecht, Boston, London, 103–107, 1996.

[**7**] K. R. Goodearl, S. Launois and T. H. Lenagan, Totally nonnegative cells and matrix Poisson varieties, *Adv. Math.*, 226:779–826, 2011.

[**8**] C. R. Johnson and R. S. Smith, Intervals of inverse M-matrices, *Reliab. Comput.*, 8:239–243, 2002.

[**9**] J. R. Kuttler, A fourth-order finite-difference approximation for the fixed membrane eigenproblem, *Math. Comp.*, 25:237–256, 1971.

[**10**] S. Launois and T. H. Lenagan, Efficient recognition of totally nonnegative matrix cells, *Found. Comput. Math.*, 14(2):371–387, 2014.

[**11**] A. Pinkus, *Totally Positive Matrices*, Cambridge Tracts in Mathematics 181, Cambridge Univ. Press, Cambridge, UK, 2010.

[**12**] J. Rohn, Positive definiteness and stability of interval matrices, *SIAM J. Matrix Anal. Appl.*, 15(1):175–184, 1994.

Robust indoor localization via interval analysis

Mohamed-Hédi Amri, Yasmina Becis, Didier Aubry, and Nacim Ramdani

Univ. Orléans, INSA-CVL, PRISME, EA 4229, F45072, Orléans, France
{mohamed-hedi.amri, yasmina.becis-aubry, didier.aubry, nacim.ramdani}@univ-orleans.fr

Keywords: set inversion, relaxed set intersection, multi-modal data fusion.

Indoor location sensing systems constitute a growing field of research involving both theoretical and applicative challenges. These systems are used in several applications of sensor networks such as tracking and monitoring. The design of an indoor positioning system depends on sensor's technology. The most common positioning systems are based on infrared (IR) technology. Such sensors are low-cost and non-wearable. These are binary sensors that can only detect the presence of a moving object in their visibility range. Pyroelectric infrared (PIR) sensors have been widely deployed in commercial applications, to detect human presence, to trigger alarms or to control lighting. PIR sensors networks are now employed in several advanced applications a.o. to achieve coverage, assist surveillance as well as perform tracking. However, indoor positioning systems using PIR sensors, may have some limitations. It is then desirable to combine them with other modalities to improve localisation accuracy. Dynamic triangulation using Received Signal Strength indicators (RSSI) is a good candidate.

Inspired by the results obtained by [1-4], we developed a method for tracking the location of residents in smart homes using only binary PIR sensors, and also by combining them with ultra wide-band based RSSI. We consider the unknown but bounded error framework and allow for possible sensor failure. Our set-membership estimation

algorithm is the classical interval-based predictor-correcteur algorithm based on the q-relaxed intersection, but with poorly known dynamical model. In fact, the actual mobility model for the inhabitant is unknown, thus we only consider a random walk with maximal velocity. The measurements data are gathered at discrete time instants. The method has been validated with actual data from a living-lab [5] and also as a tool for sensor fault detection and isolation [6].

Acknowledgement

This work is supported by the French FUI 14 e-monitorâge project.

References

[**1**] F. Mourad, H. Snoussi and C. Richard, Interval-based localization using RSSI comparison in MANETs, *IEEE Transactions on Aerospace and Electronic Systems* 2010.

[**2**] F. Mourad, H. Snoussi, M. Kieffer, and C. Richard, Robust interval-based localization algorithms for mobile sensor networks, *Int. Journal of Distributed Sensor Networks*, 7(1), 2012.

[**3**] V. Drevelle and P. Bonnifait, Robust positioning using relaxed constraint-propagation, In *Proc. of IEEE Conference on Intelligent Robots and Systems (IROS)*, 4843-4848, 2010.

[**4**] V. Drevelle and P. Bonnifait, A set-membership approach for high integrity height-aided satellite positioning, *GPS Solutions*, 15(4), 357-368, 2011.

[**5**] M.H. Amri, Y. Becis, D. Aurby, and N. Ramdani, Indoor Human/Robot Localization using Robust Multi-modal Data Fusion, in *IEEE Int. Conf. on Robotics and Automation*, 2015.

[**6**] M.H. Amri, D. Aurby, Y. Becis, and N. Ramdani, Robust Fault Detection and Isolation applied to Indoor Localization, in *Proceedings of the 9th IFAC SafeProcess*, 2015.

Comparison of Kalman versus Interval based loop detection problem

Clément Aubry

ENSTA-Bretagne, LAB-STICC / OSM,
2 rue françois verny, 29200 Brest Cedex
clement.aubry@gmail.com

Keywords: loop detection, Kalman, interval analysis

Introduction

Detecting loops in a mobile robot trajectory is a problem that can be resolved by two main approaches: using exteroceptive measurement and comparing the environment to a knowledge database or using proprioceptive measurement using a method developed in previous work that use interval analysis [1]. This talk proposes to compare Kalman versus the interval approach applied to loop detection with proprioceptive measurement only.

Kalman based method

A Kalman filter could be implemented without exteroceptive measurement for localization problem (dead reckoning). In this case, this Kalman predictor integrates proprioceptive data and is able to estimate the position of the robot with an error given by a covariance matrix.

When the evolution of a robot is described by classical state equation:

$$x_{k+1} = A_k.x_k + u_k, \tag{1}$$

where u_k represent inputs, A_k the state matrix and x_k the state of the robot, a Kalman predictor can be applied to determine it's uncertain

state $\hat{x}$:

$$\begin{cases} \hat{x}_{k+1} = A_k.\hat{x}_k + u_k \\ \Gamma_{k+1} = A_k.\Gamma_k.A_k^T + \Gamma_\alpha \end{cases}, \tag{2}$$

where Γ_{k+1} is the covariance matrix representing the uncertainty and Γ_α the covariance associated with a normally distributed noise.

In order to detect loops with a Kalman filter, we propose to compute distances between two estimated positions. Since [3]:

$$\hat{x}_k = P_k^0 \hat{x}_0 + \sum_{i=0}^{k-1} P_{k+1}^i u_i \tag{3}$$

$$\Gamma_k = P_k^0 \Gamma_0 \left(P_k^0\right)^T + \sum_{i=1}^{k} P_k^i \Gamma_\alpha \left(P_k^i\right)^T \tag{4}$$

where transition matrices P_k^i are defined by

$$\begin{aligned} P_k^i &= A_{k-1}A_{k-2}...A_i.I, \\ P_k^k &= I, \\ P_k^i &= P_k^l P_l^i, \\ P_k^i &= P_k^0 \left(P_i^0\right)^{-1}, \end{aligned}$$

we can derive an explicit form of those positions. In this context, loops are detected by pairs of timesteps k_1, k_2 which satisfies $\hat{x}_{k1} = \hat{x}_{k2}$ over position coordinates.

Main results

This talk will present results of the Kalman approach to resolve loop detection problem with proprioceptive measurement only and compare them with the interval based method. We will apply the Kalman method to an experiment done by the underwater minehunter Redermor from GESMA which have been already treated with the interval approach. Figure 1 represent results as *t-planes* given by both methods.

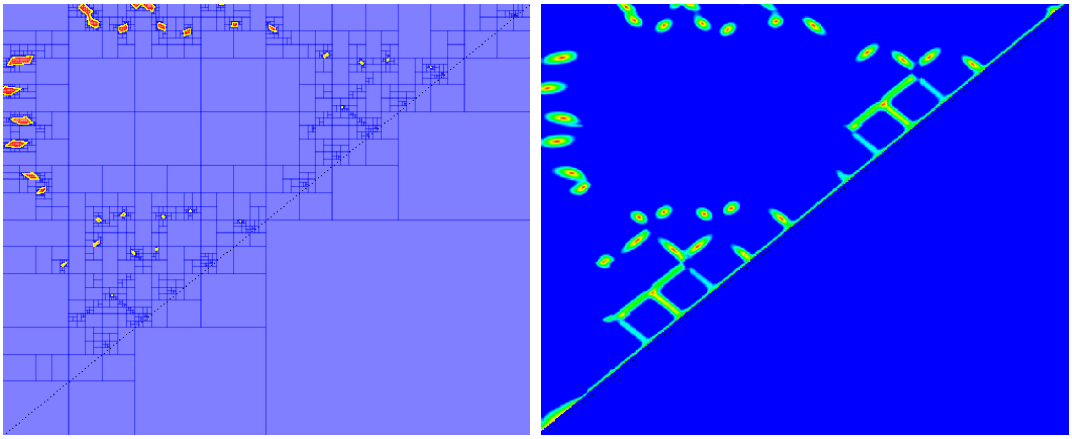


Figure 1: t-plane results of loop detection problem solved by: interval method(left), Kalman method(right).

References

[1] C.Aubry, R.Desmare and L.Jaulin, Loop detection of mobile robots using interval analysis, *Automatica* 49(2):463-470, 2013.

[2] R.E.Kalman, A new approach to linear filtering and prediction problems, *Journal of Fluids Engineering* 82(1):35–45, 1960.

[3] J.Nicola, L.Jaulin and S.Pennec, Toward the hybridization of probabilistic and set-membership methods for the localization of an underwater vehicle, *7th Small Workshop on Interval Methods*, Uppsala, Sweden, 2014.

Recent Results on Cooperative Interval Games

Jan Bok[1] and Milan Hladik[2]

[1] Computer Science Institute, Charles University in Prague, Malostranské náměstí 25, 11800, Prague, Czech Republic
bok@iuuk.mff.cuni.cz
[2] Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University in Prague, Malostranské náměstí 25, 11800, Prague, Czech Republic
hladik@kam.mff.cuni.cz

Keywords: interval game, cooperative games, core

Introduction

Cooperative interval games are a special model of cooperative transferable utility games in coalition form in which characteristic function maps to the set of closed real intervals instead of real numbers. This can be viewed as an approach to deal with inexact data since all numbers from interval are equally likely to occur and therefore, bounds of interval can be interpreted as the worst and the best possible outcome that can happen if some coalition cooperates.

There is an existing research on cooperative interval games, mainly done by Alparslan-Gök and Branzei. However, none of the existing papers focuses on selections, that is on possible outcomes of interval games (in other words, all possible classical cooperative games in which values of coalitions are contained in respective intervals of given cooperative interval game). There are some results on selections of interval games (for example analogous theorem to Bondareva-Shapley theorem), but that is almost all what is known about selections. Large portion of results is focused on examining interval valued characteristic function only, using a weakly better operator $\succ$. Interval (a, b) is weakly better than (c, d) if $a \geq c$ and $b \geq d$. Solutions (such as core

and the Shapley value) and classes obtained in this way have a big flaw that not all of their selection have the corresponding properties (for example, not all selections of a convex interval game is a classical convex game).

Basic properties

Definition 1. (Cooperative interval game) *A cooperative game is an ordered pair* (N, w)*, where* $N = \{1, 2, \ldots, n\}$ *is a set of players and* $w : 2^N \to \mathbb{IR}$ *is a characteristic function of the cooperative game. We further assume that* $w(\emptyset) = [0, 0]$*.*

The set of all interval cooperative games on player set N *is denoted by* IG^N*.*

Main results

We examine a problem of core coincidence. Core coincidence problem asks for characterization of games in which set of vectors generated by interval core coincides with the set of vectors of selection core. We found and proved such characterization. Furthermore, we introduced a concept of *strong core* - an universal stable payoff which is included in every selection and therefore we can depend on it, since it has to occur. Consequently, characterization of games with nonempty strong core is given.

We also introduced a new classes of interval games - selection monotonic, selection superadditive and selection convex games. These classes have an important and desirable property – all selections of selection monotonic games are monotonic classical game. Similarly for selection convex and selection superadditive games. We found and proved characterization theorems of all three classes. For convex selection games, characterization are inspired by Shapley's characterization of convex games. Furthermore, we show that classes based on selections and classes based on weakly better operator are incomparable for nontrivial player set (more than one player).

Selections of an interval game are useful, since they do not contain any additional uncertainty. On the top of that, selection-based classes, strong core and strong imputation have crucial property that although we deal with uncertain data, all possible outcomes preserve important properties. In case of selection classes it is preserving superadditivity, supermodularity etc. In case of strong core it is an invariant of having particular stable payoffs in each selection. Furthermore, concepts like selection core are important as well since if selection core is empty, no selection has stable payoff.

The importance of studying selection-based classes instead of the existing classes using weakly better operator can be further illustrated by the following two facts:

- Classes based on weakly better operator may contain games with selections that do not have any link with the properties of their border games and consequently no link with the name of the class. For example, superadditive interval games may contain a selection that is not superadditive.

- Selection-based classes are not contained in corresponding classes based on weakly better operator. Therefore, the results on existing classes are not directly extendable to selection-based classes.

Acknowledgement

The authors were supported by the Czech Science Foundation Grant P402/13-10660S. The work was supported by the grant SVV–2015–260223.

References

[1] S. Z. Alparslan Gök, Some results on cooperative interval games, *Optimization 63.1 (2014): 7-13.*

Some applications of interval computation in statistics

Michal Černý

University of Economics in Prague, Faculty of Computer Science and Statistics,
Winston Churchill Square 4, 13067 Prague, Czech Republic

cernym@vse.cz

Keywords: interval-valued data, interval linear regression, interval computation

Introduction

One of the main goals of interval analysis is to determine the range of a given continuous function over a given (multidimensional) interval. This talk is devoted to some particular functions important in statistics.

One-dimensional data

First we consider the case of one-dimensional data. We assume that there is a dataset $x = (x_1, \ldots, x_n)$ (a random sample from some distribution, say) and a continuous function (statistic) $S(x)$. The dataset x is unobservable; what is observable is a collection of intervals $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ such that we are guaranteed that $x \in \boldsymbol{x}$ a.s. If we do not make any stronger assumptions on the distribution of $(x, \boldsymbol{x})$, then the maximum information we can infer about $S(x)$ from the observable data $\boldsymbol{x}$ is the pair of bounds $\underline{S} = \min\{S(\xi) : \xi \in \boldsymbol{x}\}$ and $\overline{S} = \max\{S(\xi) : \xi \in \boldsymbol{x}\}$.

Only a few statistics can be evaluated by the interval arithmetic, such as the sample mean or variance $n^{-1} \sum_i (x_i - \mu)^2$, when the true mean μ is known. More often, the arithmetical expressions suffer from the dependency problem.

One of the best understood statistics is the sample variance $\sigma^2 = (n-1)^{-1}\sum_{i=1}^n (x_i - n^{-1}\sum_{j=1}^n x_j)^2$. It is directly seen that $\underline{\sigma^2}$ can be computed in weakly polynomial time; however, there even exists a strongly polynomial method. On the other hand, computation of $\overline{\sigma^2}$ is NP-hard and inapproximable with an arbitrary absolute error. It is an open problem whether it is efficiently approximable with some "reasonable" relative error. A good news is that $\overline{\sigma^2}$ can be computed in pseudopolynomial time. Furthermore, many special cases solvable in polynomial time are known. We will study the algorithm by Ferson et al. [5], which works in time $O(2^{\omega_n} n^2)$, where ω_n is the size of the largest clique of the undirected graph $G(V,E)$ with $V = \{1,\ldots,n\}$ and $\{i,j\} \in E$ iff $[\boldsymbol{x}_i^C \pm n^{-1}\boldsymbol{x}_i^\Delta] \cap [\boldsymbol{x}_j^C \pm n^{-1}\boldsymbol{x}_j^\Delta] \neq \emptyset$. In general, ω_n can be large, but in many reasonable and natural stochastic setups it seems that $\omega_n = O(\log n)$ on average, making the algorithm practically very useful. Moreover, it seems that $\mathrm{var}(\omega_n) = O(1)$, showing that hard instances occur very rarely.

We will deal with other statistics of one-dimensional data, such as higher moments or the coefficient of variation, from a similar perspective. We will also mention statistics important in testing hypotheses. We will also deal with simultaneous regions for dependent statistics, such as the joint region $\{(n^{-1}\sum_{i=1}^n \xi_i,\ (n-1)^{-1}\sum_{i=1}^n [\xi_i - n^{-1}\sum_{j=1}^n \xi_j]^2) \in \mathbb{R}^2 : \xi \in \boldsymbol{x}\}$ for sample mean and variance.

Linear regression

In the multivariate setup we discuss the linear regression model $y = X\beta + \varepsilon$, where the data (X, y) are unobservable and we can observe only intervals $\boldsymbol{X}, \boldsymbol{y}$ such that $X \in \boldsymbol{X}$ and $y \in \boldsymbol{y}$ a.s. Here, the most important statistics are estimators of the regression coefficients β and goodness-of-fit measures. We will study minimum-norm estimators based on L_p-norms and their associated loss functions, such as Ordinary Least Squares, Generalized Least Squares, Least Absolute Deviations and Chebyshev Approximation. We show that the orthant decomposition of the parameter space based on Oettli-Prager Theorem yields useful algorithms, which are exponential in the number of

regression parameters, but *not* in the number of observations.

We will also show how the orthant decomposition method applies to a form of the Errors-In-Variables model. In particular, we assume that the observations of both X and y are contaminated by random errors with a bounded support with a common radius. Then, the orthant decomposition method allows us to construct a consistent estimator of the regression parameters and the error radius.

***

The talk summarizes some well-known results, some new results as well as research challenges on the border between interval theory and statistics.

Acknowledgment. Many thanks to my colleagues and friends (alphabetically): Jaromír Antoch, Milan Hladík, Miroslav Rada and Ondřej Sokol.

This work was supported by the Czech Science Foundation under Grant P402/12/G097.

References

[1] M. Černý, J. Antoch and M. Hladík, On the possibilistic approach to linear regression models involving uncertain, indeterminate or interval data, *Information Sciences* 244: 26–47, 2013.

[2] M. Černý and M. Hladík, The complexity of computation and approximation of the t-ratio over one-dimensional interval data, *Computational Statistics & Data Analysis* 80: 26–43, 2014.

[3] E. Dantsin, V. Kreinovich, A. Wolpert and G. Xiang, Population variance under interval uncertainty: A new algorithm, *Reliable Computing* 12(4): 273–280, 2006.

[4] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré and M. Aviles, Computing variance for interval data is NP-hard, *ACM SIGACT News* 33(2): 108–118, 2002.

[5] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, Exact bounds on finite populations of interval data, *Reliable Computing* 11(3): 207–223, 2005.

[6] S. Ferson, V. Kreinovich, J. Hajagos, W. Oberkampf and L. Ginzburg, *Experimental Uncertainty Estimation and Statistics for Data Having Interval Uncertainty*, Sandia National Laboratories, Albuquerque, 2007.

[7] M. Hladík and M. Černý, Linear regression with interval data: Computational issues, Technical report II/2015, Department of Econometrics, University of Economics in Prague, `http://nb.vse.cz/~cernym/lri.pdf`.

[8] M. Hladík and M. Černý, Total least squares and Chebyshev norm, *Proceedings of International Conference on Computational Science (ICCS) 2015*, Reykjavik, to appear in *Procedia Computer Science*, Preprint: `http://nb.vse.cz/~cernym/tls.pdf`.

[9] J. Horowitz, C. Manski, C. Ponomareva and J. Stoye, Computation of bounds on population parameters when the data are incomplete, *Reliable Computing* 9(6): 419–440, 2003.

[10] H. T. Nguyen, V. Kreinovich, B. Wu and G. Xiang, *Computing Statistics under Interval and Fuzzy Uncertainty*, Studies in Computational Intelligence 393, Springer, 2012.

[11] S. Schön and H. Kutterer, Using zonotopes for overestimation-free interval least-squares: Some geodetic applications, *Reliable Computing* 11: 137–155, 2005.

[12] E. Tamer, Partial identification in econometrics, *Annual Review of Economics* 2: 167–195, 2010.

[13] G. Xiang, M. Ceberio and V. Kreinovich, Computing population variance and entropy under interval uncertainty: Linear-time algorithms, *Reliable computing* 13(6): 467–488, 2007.

Guaranteed Coverage Assessment of a Robotic Survey with Uncertain Trajectory

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. Survey missions consists in gathering information (image, relief...) about every point of an area, using embedded sensors like cameras, lidars or sonars. Other tasks such as lawn-mowing or cleaning also involve covering an area with an effector (e.g a blade or a vacuum cleaning system).

In practice, the robot trajectory is known with an uncertainty, which propagates as an uncertainty on the area that has been actually covered during the mission. Assessing the acquired of the survey mission is an important task [1], to ensure there will be no gap when merging the acquired data.

A interval analysis based set-membership approach to computing the explored area with uncertain trajectory will be presented. Single- and multi-robot applications will be demonstrated.

Problem statement

Let us consider a mobile robot equipped with actuators and sensors. The robot is classically represented by the following state equations

$$\begin{cases} \dot{\mathbf{x}}(t) &= \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \\ \mathbf{y}(t) &= \mathbf{g}\left(\mathbf{x}(t)\right), \end{cases} \tag{1}$$

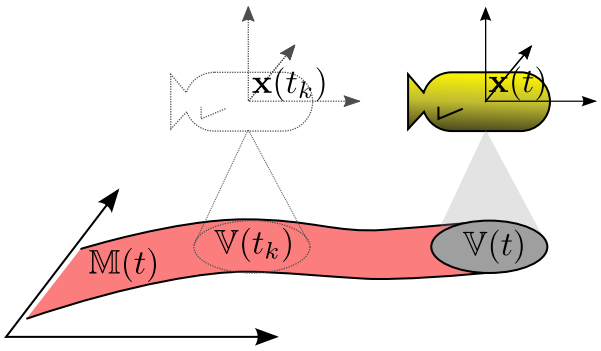
where $\mathbf{x}$ denotes the robot's state vector (e.g. position, velocity...), $\mathbf{u}$ is the input vector and $\mathbf{y}$ is the observation vector. The robot's evolution is modeled by the function $\mathbf{f}$ and $\mathbf{g}$ is the observation function.

The robot uses a sensor for searching or mapping, that can cover a given area. Let $\mathbb{V}$ be the set-valued function that returns the area which is in the robot's field of view at each time, i.e the *visible area* $\mathbb{V}(t)$ at time t. It is defined by the *visibility function* v, such that $v\left(\mathbf{z}, \mathbf{x}(t)\right)$ is negative iff the point $\mathbf{z}$ is in the range of the sensor for the given the robot state $\mathbf{x}(t)$:

$$\mathbb{V}(t) = \left\{\mathbf{z} \in \mathbb{R}^2 : v\left(\mathbf{z}, \mathbf{x}(t)\right) \leq 0\right\}. \tag{2}$$

The *mission's surveyed area* $\mathbb{M}$ is the area that has been covered by the robot's sensor at the end of the exploration mission, i.e from the mission start t_0 to the mission end t_f:

$$\mathbb{M} = \bigcup_{t \in [t_0, t_f]} \mathbb{V}(t). \tag{3}$$



Assuming bounded-error knowledge of $\mathbf{u}$ and $\mathbf{y}$, i.e $\mathbf{u}(t) \in [\mathbf{u}](t)$ and $\mathbf{y}(t) \in [\mathbf{y}](t)$, the problem we want to address consists in computing a bracketing of $\mathbb{M}$ in the form of a set-interval $[\underline{\mathbb{M}}, \overline{\mathbb{M}}]$. The set $\underline{\mathbb{M}}$ is guaranteed to have been covered during the mission, while the complement of $\overline{\mathbb{M}}$ has guaranteedly not been covered.

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 surveyed area* $\mathbb{M}^{\forall} = \left\{\mathbf{z} \in \mathbb{R}^2 \mid \forall \mathbf{x} \in \mathcal{T},\ \exists t,\ v\left(\mathbf{z}, \mathbf{x}(t)\right) \leq 0\right\}$, and the *possibly surveyed area* $\mathbb{M}^{\exists} = \left\{\mathbf{z} \in \mathbb{R}^2 \mid \exists \mathbf{x} \in \mathcal{T},\ \exists t,\ v\left(\mathbf{z}, \mathbf{x}(t)\right) \leq 0\right\}$. We thus have $\underline{\mathbb{M}} \subseteq \mathbb{M}^{\forall} \subseteq \mathbb{M} \subseteq \mathbb{M}^{\exists} \subseteq \overline{\mathbb{M}}$.

Approach

Simple approach: union of visible area intervals

A first approach to compute a $[\underline{\mathbb{M}}_{\cup}, \overline{\mathbb{M}}_{\cup}]$ set-interval has been presented in [2]. It consists in first contracting the tube $[\mathbf{x}](t)$ with the constraints of Eq. 1. This is done using the contractor programming approach. Then, the surveyed area interval is obtained as the union of visible area intervals: $[\underline{\mathbb{M}}_{\cup}, \overline{\mathbb{M}}_{\cup}] = \bigcup_{t \in [t_0, t_f]} [\underline{\mathbb{V}}(t), \overline{\mathbb{V}}(t)]$. Symbolic interval arithmetic [3] is used to derive lower and upper bounds of $v(\mathbf{z}, [\mathbf{x}](t))$. A set-inversion method is then employed for surveyed area computation.

Taking robot evolution into account

While being very fast, the previous approach provides very pessimistic bounds for $\mathbb{M}$: Using a tube $[\mathbf{x}](t)$ to represent the set of admissible trajectories $\mathcal{T}$ discards temporal dependancies.

This pessimism is clearly visible when the robot position uncertainty is larger than its sensor field of view. Indeed, in this case, by considering independently the visible areas at each time, it is often not possible to guarantee a non-empty lower-bound for $\mathbb{M}$. However, by considering the admissible trajectories, an non-empty set $\underline{\mathbb{M}}$ can

be guaranteed to have been surveyed, thanks to inter-temporal dependency of the robot positions (i.e robot evolution model).

We propose an improved method which yields a tighter set-interval for $\mathbb{M}$. It consists in partitioning and contracting the tube $[\mathbf{x}]$ at given times. This thinner representation of $\mathcal{T}$ yields a smaller set-interval for the mission surveyed area, at the expense of a longer computation time.

Results and comparison of the two methods will be presented on a single robot and a multi-robot test-case.

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] L. Jaulin and G. Chabert, Resolution of nonlinear interval problems using symbolic interval arithmetic. *Eng. Appl. of Artif. Intel.*, 23(6):1035–1049, 2010.

Verification of zeros in underdetermined systems.

Peter Franek[1] and Marek Krčál[2]

[1] Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodarenskou vezi 271/2, 182 07 Prague, Czech Republic
peter.franek@gmail.com

[2] IST Austria, Am Campus 1, 3400 Klosterneuburg, Austria
marek.krcal@ist.ac.at

Keywords: Verification, Zero set, Robustness, Computational topology

Introduction

Let $B \subseteq \mathbb{R}^m$ be a box (product of closed intervals). While the nonexistence of a zero of a continuous functions $f\colon B \to \mathbb{R}^n$ can often be verified by interval arithmetic, existence verification requires additional ingredients. In case of a continuous function $f\colon [-1,1]^n \to \mathbb{R}^n$, such ingredients include Brouwer's fixed point theorem, Miranda's theorem, or, more generally, topological degree.

For underdetermined systems—that is, $\dim B > n$—the problem of zero verification may be surprisingly complex, with connections to the field of computational topology.

The section method and its incompleteness

One way to reduce underdetermined systems to "square" systems is to fix some coordinates and verify the existence of a zero of $f(\cdot, y_0)$ where y_0 represents some fixed $\dim B - n$ coordinates. If the Jacobian f' has maximal rank n in every $x \in f^{-1}(0)$, then the zero set has dimension $\dim B - n$ and—possibly after a rotation—we can eventually find y_0 so that $f(\cdot, y_0)$ has a zero and verify it. We call this "section method" due to the fact that we analyze the restriction of f to a section $B' \times \{y_0\}$.

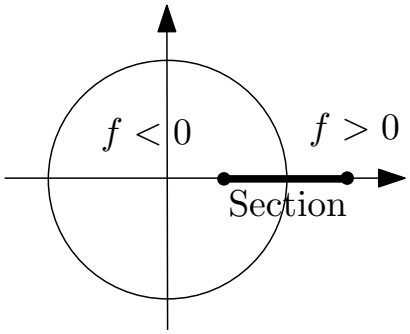


Figure 1: Zero detection for $f(x, y) = x^2 + y^2 - 1$.

There is a potential problem with this approach: if 0 is a singular value of f, then the zero set may have lower dimension and resist detection by the section method. This is illustrated by the quadratic function $f\colon [-1,1]^4 \to \mathbb{R}^3$ defined component-wise by $(x_1x_3 + x_2x_4, x_2x_3 - x_1x_4, x_1^2 + x_2^2 - x_3^2 - x_4^2)$ which has a single singular zero at the origin (and also every "close enough" perturbation of f has a zero). The restriction of f to the unit sphere $S^3 = \{x : |x| = 1\}$ is the Hopf map $S^3 \to S^2$. For any rotation R of $\mathbb{R}^4$, once we use the 3-dimensional section $[-1,1]^3 \times \{0\}$ containing the origin, the "section method" for zero verification of $(f \circ R)|_{[-1,1]^3\times\{0\}}$ fails. Arbitrarily small perturbation of $f \circ R$ have no zero in $[-1,1]^3 \times \{0\}$, thus the zero cannot be verified by common methods such as Brouwer or Miranda's theorem (as these methods are stable with respect to small perturbations). However, the zero of f cannot be removed by perturbations: in particular, any continuous g such that $\|g - f\| \leq 1$ has a nonempty zero set. Therefore, it is natural to expect that more sophisticated verification methods should detect the zero of f.

The key is to analyze the function on a subdomain A where it is further away from zero, that is, $A := \{x\colon |f(x)| \geq r\}$ for some $r > 0$. If $f|_A : A \to \mathbb{R}^n \setminus \{0\}$ *cannot* be extended to a map $B \to \mathbb{R}^n \setminus \{0\}$ then f has a zero in B that is r-robust in the sense explained below. We summarize the up-to-date results about decidability of the extension problem and its implications for the zero verification problem.

Zero verification with incomplete information

The above mentioned method using non-extendability can not only verify a zero of f, but also a zero of any continuous r-perturbation g of f, that is, $g\colon B \to \mathbb{R}^n$ s.t. $\|g - f\|_\infty \leq r$. The non-extendability criterion is complete in the sense that once it fails to verify a zero, there has to be a continuous r-perturbation g of f without a zero.

Thus the method is useful in situations where we don't know the exact values of the function f and only deal with its approximation. For example, the function may be inferred from measurements or come from a scientific model that only approximates the reality. An important instance is the case when the function f is given just by its values on the vertices of a cubical grid and a Lipschitz constant, that is, as a higher dimensional bitmap.

References

[1] Franek P, Krčál M, *Robust satisfiability of systems of equations*, Proceedings of ACM-SIAM Symposium of Discrete algorithms, Portland 2014, pp 193–203. Extended version to appear in Journal of the ACM, preprint arXiv:1402.0858

Solving and visualizing nonlinear constraint satisfaction problems

Elif Garajová and Martin Mečiar

Charles University, Faculty of Mathematics and Physics,
Malostranské nám. 25, 11800, Prague, Czech Republic
elif.garajova@gmail.com, martinmeciar@centrum.sk

Keywords: nonlinear constraints, interval analysis, interval data visualization

Introduction

The concept of constraint satisfaction can be used to formalize many practical problems involving a set of variables with given properties. Our talk will focus on continuous constraint satisfaction problems with interval domains, where the solution set is described by a system of nonlinear inequalities. We would like to introduce an interval solver for nonlinear constraints [1] with a visualization software, which will be included in the interval toolbox Lime.

Interval solver for nonlinear constraints

Our solver is based on the branch-and-bound algorithm Sivia (Set Inversion via Interval Analysis, see [2]), which approximates the solution set $\mathbb{X}$ of a nonlinear problem using 3 sets $\mathcal{S}, \mathcal{N}, \mathcal{E}$ of non-overlapping interval boxes (also referred to as a paving) satisfying $\mathcal{S} \subseteq \mathbb{X} \subseteq (\mathcal{S} \cup \mathcal{E})$ and $\mathbb{X} \cap \mathcal{N} = \emptyset$. Interval contractors, such as the forward-backward contractor, are used to improve the efficiency of the basic algorithm. We have also implemented a simple unification procedure to decrease the number of boxes characterizing the solution set of a given problem.

An interesting application of the algorithm is the visualization of complex interval arithmetic. If we define (rectangular) complex intervals as two-dimensional interval boxes [3], we can describe an exact

product (or quotient) of two complex intervals using nonlinear constraints and variables with continuous domains.

Interval data visualization

The visualization software implements several methods for approximating the exact solution set of a nonlinear system using the results obtained by the solver. The approximation methods divide the set of undecided boxes in the resulting paving into the satisfying set and the non-satisfying set.

The visualization software communicates with the user through a simple and user-friendly graphical interface. Two or more pavings can be displayed at once for a visual comparison of the sets. It also offers a wide range of minor functions, which can be used to customize the visualization settings.

References

[**1**] E. Garajová, *Intervalový solver nelineárních podmínek [Interval solver for nonlinear constraints]*, Bachelor's thesis, 2014.

[**2**] L. Jaulin et al., *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer-Verlag, 2001.

[**3**] R. E. Boche, *Complex interval arithmetic with some applications*, Technical Report LMSC4-22-66-1, Lockheed Missiles and Space Co., Sunnyvale, CA, USA, 1966.

Permuted graph bases for verified computation of invariant subspaces

Federico Poloni and Tayyebe Haqiri

Dipartimento di Informatica, Università di Pisa,
Largo B. Pontecorvo 3, 56127 Pisa, Italy
fpoloni@di.unipi.it
Department of Mathematics, School of Mathematics and Computer Sciences,
Shahid Bahonar University of Kerman, P. O. Box 76169-14111, Kerman, Iran
Haqiri@math.uk.ac.ir

Keywords: Permuted graph bases, invariant subspaces, interval arithmetic, Krawczyk method, verified computation

The problem

A subspace $U \in \mathbb{C}^{n\times k}$ is called *invariant* under $H \in \mathbb{C}^{n\times n}$ if Hu is in U for all u in U [1]. The invariant subspace problem can be stated as finding $U \in \mathbb{C}^{n\times k}$ and $R \in \mathbb{C}^{k\times k}$ such that $HU = UR$, and the eigenvalues of R are a specified subset of those of H.

If one constrains U to be in the form $U = \begin{bmatrix} I_{k\times k} \\ X_{(n-k)\times k} \end{bmatrix}$, the problem of finding an invariant subspace can be recast as a (non-Hermitian) *algebraic Riccati equation* (NARE)

$$F(X) := Q + XA + \tilde{A}X - XGX = 0, \tag{1}$$

where $H = \begin{bmatrix} A_{k\times k} & -G_{k\times(n-k)} \\ -Q_{(n-k)\times k} & -\tilde{A}_{(n-k)\times(n-k)} \end{bmatrix}$ and $R = A - GX$.

Permuted graph bases

An idea reappered recently in the matrix equation community [3] is that by applying a suitable permutation of the entries one can get an equation in which the solution X has smaller entries.

Theorem 1 ([2]). *Let $U \in \mathbb{C}^{n\times k}$ have full column rank. Then, there exists a permutation matrix $P \in \mathbb{R}^{n\times n}$ so that the top $k\times k$ submatrix E of $P^TU = \begin{bmatrix} E \\ A \end{bmatrix}$ is nonsingular, and the matrix $Z = AE^{-1} \in \mathbb{R}^{(n-k)\times k}$ is such that $|Z_{ij}| \leq 1$ for all i, j.*

By constructing this permutation P, we can replace the original NARE (1) with the one associated with $\tilde{H} = PHP^T$, whose solution Z has smaller entries.

An efficient enclosure for the solutions to NAREs

From now on we focus on the NARE(1). We wish to use the following classical result to find an enclosure for the solution X.

Theorem 2 ([4] (modified Krawczyk method)). *Assume that $f : D \subset \mathbb{C}^n \to \mathbb{C}^n$ is continuous in D. Let $\tilde{x} \in D$ and $\mathbf{z} \in \mathbb{IC}^n$ be such that $\tilde{x} + \mathbf{z} \subseteq D$. Moreover, assume that $\mathcal{S} \subset \mathbb{C}^{n\times n}$ is a set of matrices containing all slopes $S(\tilde{x}, y)$ for $y \in \tilde{x} + \mathbf{z} := \mathbf{x}$. Finally, let $R \in \mathbb{C}^{n\times n}$. Denote by $\mathcal{K}_f(\tilde{x}, R, \mathbf{z}, \mathcal{S})$ the set*

$$\mathcal{K}_f(\tilde{x}, R, \mathbf{z}, \mathcal{S}) := \{-Rf(\tilde{x}) + (I - RS)z : S \in \mathcal{S}, z \in \mathbf{z}\}.$$

Then, if

$$\mathcal{K}_f(\tilde{x}, R, \mathbf{z}, \mathcal{S}) \subseteq \operatorname{int} \mathbf{z}, \tag{2}$$

the function f has a zero x^* in $\tilde{x} + \mathcal{K}_f(\tilde{x}, R, \mathbf{z}, \mathcal{S}) \subseteq \mathbf{x}$. Moreover, if $\mathcal{S}$ also contains all slope matrices $S(x, y)$ for $x, y \in \mathbf{x}$, then this zero is unique in $\mathbf{x}$.

The recent works [5, 6] have successfully applied the modified Krawczyk method to several matrix equations, adding some crucial issues:

1. Let

$$A - GX = V_1\Lambda_1W_1; \text{ with } V_1, W_1, \Lambda_1 \in \mathbb{C}^{n\times n}, \Lambda_1 = \operatorname{Diag}(\lambda_{11}, \ldots, \lambda_{n1}), V_1W_1 = I,$$

$$\tilde{A}^* - G^*X^* = V_2\Lambda_2W_2; \text{ with } V_2, W_2, \Lambda_2 \in \mathbb{C}^{n\times n}, \Lambda_2 = \operatorname{Diag}(\lambda_{12}, \ldots, \lambda_{n2}), V_2W_2 = I,$$

and set

$$R = (V_1^{-T} \otimes W_2^*).\Delta^{-1}.(V_1^T \otimes W_2^{-*}), \text{ where } \Delta = I \otimes \Lambda_2^* + \Lambda_1^T \otimes I.$$

This choice of R is so that its computation can be performed in $O(n^3)$, rather than the $O(n^5)$ obtained by vectorization without this improvement.

2. To reduce the problematic *wrapping effect* of interval arithmetic, use $\hat{f}$ as a linearly transformed function instead of f

$$\hat{f}(\hat{x}) = (V_1^T \otimes W_2^{-*}) f((V_1^{-T} \otimes W_2^*)\hat{x}),$$

where $(V_1^{-T} \otimes W_2^*)\hat{x} = x$.

We combine ideas from these two approaches to obtain an algorithm that can find enclosures for a larger class of problems in our experiments. A suitable modification of the ideas in Theorem 1 [3] can be used to work with structured invariant subspace problems and Hermitian algebraic Riccati equations (CAREs).

References

[1] I. Gohberg, P. Lancaster and L. Rodman, Invariant subspaces of matrices with applications, Classics in Applied Mathematics, SIAM. ISBN: 0-89871-608-X.

[2] D. E. Knuth, Semioptimal bases for linear dependencies, *Linear and Multilinear Algebra* 17 (1985), no. 1, 1–4.

[3] V. Mehrmann and F. Poloni, Doubling algorithms with permuted Lagrangian graph bases, *SIAM J. Matrix Anal. Appl.* 33 (2012), no. 3, 780–805.

[4] S. M. Rump, *Kleine fehlerschranken bei matrixproblemen*, Ph.D. thesis, Fakultät für Mathematik, Universität Karlsruhe, 1980.

[5] A. Frommer and B. Hashemi, Verified computation of square roots of a matrix, *SIAM J. Matrix Anal. Appl.* 31 (2009), no. 3, 1279–1302.

[6] B. Hashemi and M. Dehghan, Efficient computation of enclosures for the exact solvents of a quadratic matrix equation, *Electron. J. Linear Algebra* 20 (2010), 519–536.

Yet another method for solving interval linear equations

Milan Hladík

Charles University, Faculty of Mathematics and Physics,
Department of Applied Mathematics,
Malostranské nám. 25, 11800, Prague, Czech Republic
hladik@kam.mff.cuni.cz

Keywords: interval linear equations, interval computations

Introduction

Solving interval linear equations is one of the basic tasks in interval computations. Let an interval system

$$Ax = b, \quad A \in \mathbf{A},\ b \in \mathbf{b},$$

be given, where $\mathbf{A}$ is an interval matrix and $\mathbf{b}$ an interval vector. The problem is to tightly enclose the solution set defined as

$$\{x \in \mathbb{R}^n \mid \exists A \in \mathbf{A} \exists b \in \mathbf{b} : Ax = b\}.$$

This is a difficult task in general since there is no polynomial-time enclosure method with any fixed accuracy unless $P = NP$. Even to check if the solution set is non-empty is an NP-hard problem.

There are, however, lots of methods known that work well in most of the cases. They differ in computational time and tightness of the resulting enclosures. Many methods use preconditioning, in particular preconditioning by (an approximation of) the midpoint inverse matrix, since a tight enclosure can be computed then. Thus, we will assume that a given interval linear system is preconditioned such that its midpoint matrix is the identity matrix.

Main results

We present a new operator for enclosing the solutions set; see [1]. It generalizes the classical interval Gauss–Seidel operator. Also, based on the new operator and properties of the well-known methods, we propose a new enclosing algorithm, We call it *the magnitude method* since it utilizes the easily computable magnitude (i.e., entrywise the largest absolute value) of the interval hull of the solution set.

The performance of the method depends on how several quantities are computed. If they are computed exactly, then the method yields the interval hull of the preconditioned solution set. On the other hand, even if they are approximated very roughly, the magnitude method performs always as well (w.r.t. tightness) as the Gauss–Seidel iteration method.

We illustrate by numerical examples that our approach overcomes the Gauss–Seidel iteration method with respect to both computational time and sharpness of enclosures. Compared to the INTLAB function `verifylss`, the magnitude method produces always tighter enclosures. Unless the input interval data are very narrow, it also overcomes `verifylss` with respect to computational time.

Acknowledgement

This work was supported by CE-ITI (GAP202/12/G061) of the Czech Science Foundation.

References

[1] M. HLADÍK, New operator and method for solving real preconditioned interval linear equations, *SIAM Journal on Numerical Analysis* 52(1):194–206, 2014.

Sub-Interval Perturbation Method for Standard Eigenvalue Problem

Nisha Rani Mahato[1] and Snehashish Chakraverty[1,*]

[1] Department of Mathematics,
National Institute of Technology Rourkela, Odisha, India

[*] sne_chak@yahoo.com

Keywords: interval analysis, sub-interval, interval perturbation, standard eigenvalue

Introduction

Solution of finite element analysis under dynamic condition led to generalized eigenvalue problem. Generally we have crisp values of material properties for structural dynamic problems. As a result of errors in measurements, observations, calculations or due to maintenance induced errors etc. we may have uncertain bounds which may be modeled through interval analysis. So, this paper deals with sub-interval perturbation procedure for computing upper and lower eigenvalue and eigenvector bounds of standard eigenvalue problem with interval parameters.

Few literatures for solving structural dynamics problem based on interval analysis in perturbation approach of structural dynamics are available. In this regard, Alefeld and Herzberger [1] and Moore et al. [2] presented a detailed discussion on interval computations. Qiu et al. [3] proposed an interval perturbation approximating formula for evaluating interval eigenvalues for structures. An inner approximation algorithm has been proposed by Hladik et al. [4] with perturbations belonging to some given interval. For structures with large interval parameters Qiu and Elishakoff [5] proposed a subinterval perturbation for estimating static displacement bound.

As such present section gives the introduction and the next section discusses sub-interval perturbation procedure. In third section, main

results based on the developed procedure is considered for a structural standard eigenvalue problem and lastly conclusion is included.

Sub-Interval Perturbation Procedure

Interval matrix of eigenvalue problem has been initially divided into small sub-intervals and then for each interval, unperturbed crisp eigenvalues λ_i^c and eigenvectors x_i^c are obtained from the corresponding center matrices K^c. Generally in structural dynamic problems the stiffness and mass matrices are symmetric in nature. So, by using orthogonality condition of eigenvectors of symmetric matrices and neglecting higher order perturbations, structural system has been perturbed through λ_i^c and x_i^c to obtain eigenvalues $\delta\lambda_i$ and eigenvectors δx_i.

Sub-Interval

Let $A^I = [\underline{a},\ \overline{a}]$ be an interval, then its subintervals may be obtained by dividing the interval into m equal parts with width $(\underline{a}-\overline{a})/m$. In case of an interval matrix K^I of order n, the subinterval matrices are obtained with respect to each element having matrix width $(\underline{K}-\overline{K})/m$. So, the subinterval matrices may be obtained as $K^I = [\underline{K},\ \overline{K}] = \bigcup_{k=1}^{m} K_t^I$ where $K_t^I = [\underline{K}+(t-1)(\underline{K}-\overline{K})/m,\ \underline{K}+t(\underline{K}-\overline{K})/m],\ t = 1,\ 2, ...,\ m$.

Perturbation

Let us consider a standard interval eigenvalue problem

$$K^I x_i^I = \lambda_i^I x_i^I,\ i = 1,\ 2, ...,\ n \tag{1}$$

In term of interval center and radius, equation (1) may be written as

$$(K^c + \delta K)(x_i^c + \delta x_i) = (\lambda_i^c + \delta\lambda_i)(x_i^c + \delta x_i) \tag{2}$$

Then the required first order perturbation of eigenvalues for $\delta\lambda_i = [-\Delta\lambda_i,\ \Delta\lambda_i]$ where $\Delta\lambda_i = (x_i^c)^T \Delta K x_i^c$ may be given by

$$\underline{\lambda}_i = \lambda_i^c - (x_i^c)^T \Delta K x_i^c \tag{3a}$$

$$\overline{\lambda}_i = \lambda_i^c + (x_i^c)^T \Delta K x_i^c \tag{3b}$$

and the first order perturbation of eigenvectors may be written as

$$\underline{x}_i = x_i^c + \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{(x_j^c)^T \Delta K x_i^c}{\lambda_i^c - \lambda_j^c} x_j^c \tag{4a}$$

$$\overline{x}_i = x_i^c - \sum_{\substack{j=1 \\ j\neq i}}^{N} \frac{(x_j^c)^T \Delta K x_i^c}{\lambda_i^c - \lambda_j^c} x_j^c \tag{4b}$$

This perturbation procedure is then implemented over each subinterval K_t^I to obtain interval bounds of eigenvalues and eigenvectors.

Main results

We consider here a four degree of freedom spring mass system (Qiu et al. [3]) with mass matrix as crisp identity matrix and interval stiffness matrix $K^I = K^c + \delta K$ such that $\delta K = [-\Delta K, \Delta K]$ with

$$K^c = \begin{bmatrix} 3000 & -2000 & 0 & 0 \\ -2000 & 5000 & -3000 & 0 \\ 0 & -3000 & 7000 & -4000 \\ 0 & 0 & -4000 & 9000 \end{bmatrix} \text{ and } \Delta K = \begin{bmatrix} 25 & 15 & 0 & 0 \\ 15 & 35 & 20 & 0 \\ 0 & 20 & 45 & 25 \\ 0 & 0 & 25 & 55 \end{bmatrix}.$$

Accordingly inner and outer approximations of eigenvalue bounds may be computed for $m = 1$ and sufficiently large m respectively.

- *Inner approximation*: λ_i^I for global (without sub-intervals) stiffness matrix.
- *Outer approximation*: $\lambda_i^I = [min\underline{\lambda}_{it},\ max\overline{\lambda}_{it}]$ where $t = 1,\ 2, ...,\ m$ and m being sufficiently large.

Conclusion

This investigation presents sub-interval perturbation procedure for obtaining inner and outer approximation of eigenvalue bounds for stan-

dard interval eigenvalue problems. Accordingly corresponding perturbed eigenvectors are also computed. The perturbation of subintervals may not give exact bounds as higher order perturbations are neglected but provides a tighter first order inner approximation interval bounds with a small deflection from its center. The proposed procedure may also be applied to other practical eigenvalue problems involving interval material properties .

Acknowledgement

The authors are thankful to the Science and Engineering Research Board, Department of Science and Technology, New Delhi, India for the funding to carry out the present research.

References

[**1**] G. Alefeld and J. Herzberger, *Introduction to Interval Computations*, Academic Press, London, 1983.

[**2**] R. E. Moore, R. B. Kearfott and M. J. Cloud, *Introduction to Interval Analysis*, SIAM Publications, Philadelphia, PA, 2009.

[**3**] Z. Qiu, S. Chen and I. Elishakoff, Bounds of eigenvalues for structures with an interval description of uncertain-but-nonrandom parameters, *Chaos, Solitions and Fractals* 7:425–434, 1996.

[**4**] M. Hladik, D. Daney, E. Tsigaridas, Characterizing and approximating eigenvalue sets of symmetric interval matrices, *Computers and Mathematics with Applications* 62:3152–3163, 2011.

[**5**] Z. Qiu and I. Elishakoff, Antioptimization of structures with large uncertain-but-nonrandom parameters via interval analysis, *Computer Methods in Applied Mechanics and Engineering* 152:361–372, 1998.

Validated Explicit and Implicit Runge-Kutta Methods[1]

Alexandre Chapoutot†, Julien Alexandre dit Sandretto† and Olivier Mullier†

† U2IS, ENSTA ParisTech
828, boulevard des maréchaux, 91000, Palaiseau, France
{chapoutot,alexandre,mullier}@ensta.fr

Keywords: ordinary differential equations, guaranteed numerical integration, affine arithmetic.

Introduction

The guaranteed solution of initial value problem of ordinary differential equations is well studied from interval analysis community. In the most of the cases Taylor models are used in this context, see [1] and the references therein. In contrast, in numerical analysis community other numerical integration methods, e.g., Runge-Kutta methods, are used. Indeed, these methods have very good stability properties [2] and they can be applied on a wide variety of problems.

We propose a new method to validate the solution of initial value problem of ordinary differential equations based on Runge-Kutta methods. The strength of our contribution is to adapt any explicit and implicit Runge-Kutta methods to make them guaranteed. We experimentally verify our approach against Vericomp benchmark[2] and the results are reported in [3]. We hence extend our previous work [5] on explicit Runge-Kutta methods.

[1]This research benefited from the support of the "Chair Complex Systems Engineering - Ecole Polytechnique, THALES, DGA, FX, DASSAULT AVIATION, DCNS Research, ENSTA ParisTech, Télécom ParisTech, Fondation ParisTech and FDO ENSTA"

[2]http://vericomp.inf.uni-due.de, we consider results dated back to October 2014.

Main idea

We want to solve

$$\dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t)) \quad \text{with} \quad \mathbf{x}(0) = \mathbf{x}_0 \ . \tag{1}$$

We denote by $\mathbf{x}(t; \mathbf{x}_0)$ the solution of Equation (1) at a time t associated to the initial value $\mathbf{x}_0$. Applying a s-stage Runge-Kutta method on Equation (1), we have the following recurrence relation

$$k_i = f\left(t_n + c_i h_n, \mathbf{x}_n + h \sum_{j=1}^{s} a_{ij} k_j\right), \qquad \mathbf{x}_{n+1} = \mathbf{x}_n + h \sum_{i=1}^{s} b_i k_i \ .$$

The coefficients c_i, b_i and a_{ij} with $i, j = 1, 2, \ldots, s$ are associated to a given Runge-Kutta methods, see [2] for more details. That is for each time instant t_n, we have $\mathbf{x}_n \approx \mathbf{x}(t_n; \mathbf{x}_{n-1})$.

The challenge to make Runge-Kutta guaranteed is to compute a safe bound of the local truncation error (LTE for short) at each time t_n, that is $\mathbf{x}(t_n; \mathbf{x}_{n-1}) - \mathbf{x}_n$ must be bounded. An elegant solution to compute the formula of the LTE is based on the order condition of Runge-Kutta methods. A Runge-Kutta method has order p, i.e., $\mathbf{x}(t; \mathbf{x}_{n-1}) - \mathbf{x}_n \leq C \cdot \mathcal{O}(h^{p+1})$, with C a constant independent of f, if and only if the Taylor expansion of the true solution and that of the numerical solution have the same coefficients for the $p + 1$ first terms. In consequence, the formula of the LTE of a Runge-Kutta methods is given by the difference of the remainders of the two Taylor expansions.

Main results

John Butcher in [4] defines a generic method to compute the Taylor expansion of the true and a numerical solutions of Equation (1). It is based on the Fréchêt derivatives F of the function $\mathbf{x}(t)$. The great idea of John Butcher is to connect these Fréchêt derivatives of a given order m to a combinatorial problem to enumerate the number of trees τ with q nodes.

In summary, in [4] we have

$$\mathbf{x}^{(q)}(t) = \sum_{r(\tau)=q} \alpha(\tau) F(\tau), \quad \text{and} \quad \mathbf{x}_n^{(q)} = \sum_{r(\tau)=q} \alpha(\tau)\gamma(\tau)\psi(\tau) F(\tau) \ .$$

with $\mathbf{x}^{(q)}(t)$ the q-th time derivative of the true solution and $\mathbf{x}_n^{(q)}$ the q-th time derivative of the numerical solution of Equation (1). Coefficients $\alpha(\tau)$ and $\gamma(\tau)$ are characteristics of trees τ, see [4] for more details. Note that the coefficient $\psi(\tau)$ is a function of the coefficients c_i, b_i and a_{ij}, $i = 1, 2, \cdots, s$.

Using the approach of John Butcher, we can validate any Runge-Kutta methods of order p using the following expression of the LTE

$$\mathrm{LTE}(t, \mathbf{x}(\xi)) = \frac{h^{p+1}}{(p+1)!} \sum_{r(\tau)=q} \alpha(\tau)[1 - \gamma(\tau)\psi(\tau)]F(\tau)\left(\mathbf{x}(\xi)\right) \quad \text{with} \quad \xi \in]t_n, t_{n+1}[\ .$$

Using a classical 2-step approach of guaranteed integration, see [1], from a given guaranteed initial value $[\mathbf{x}_n]$ at time instant t_n then

1. compute an enclosure $[\tilde{\mathbf{x}}]$ of $\mathbf{x}(t)$ on the time interval $[t_n, t_{n+1}]$;
2. compute a tight enclosure of the solution $[\mathbf{x}_{n+1}]$ at time t_{n+1} using Runge-Kutta method and the LTE formula with $[\tilde{\mathbf{x}}]$.

In summary, our approach is defined by

$$k_i(t, \mathbf{x}_n) = f\left(t_j + c_i(t - t_j), \mathbf{x}_n + (t - t_n)\sum_{j=1}^{s} a_{ij}k_j\right), \tag{2a}$$

$$\mathbf{x}_{n+1}(t, \xi) = \mathbf{x}_n + (t - t_n)\sum_{i=1}^{s} b_i k_i(t, \mathbf{x}_n) + \mathrm{LTE}(t, \mathbf{x}(\xi)) \ . \tag{2b}$$

Equation (2) can be used for computing a priori enclosure and tightening the solution. Note that in case of implicit Runge-Kutta methods, the equations k_i, for $i = 1, \ldots, s$, form a contracting system of equations. In consequence, we can easily build an interval contractor from the system of k_i and so we can solve it easily.

To illustrate our approach, we consider Vericomp Problem 61

$$\begin{cases} \dot{x}_1 = 1, & x_1(0) = 0 \\ \dot{x}_2 = x_3, & x_2(0) = 0 \\ \dot{x}_3 = \dfrac{1}{6}x_2^3 - x_2 + 2\sin(p \cdot x_1) \quad \text{with} \quad p \in [2.78, 2.79], & x_3(0) = 0 \ . \end{cases}$$

Using a validated version, with our approach, of Lobatto-3C implicit Runge-Kutta method of order 4, we obtain 10.597 as the maximal width of the solution enclosure at 10 seconds (tolerance 10^{-10} on LTE) while none of Riot, Valencia-IVP, nor VNODE-LP can produce a solution at 10 seconds.

Conclusion

We presented a new class of validated numerical integration method based on explicit and implicit Runge-Kutta methods. We have a generic formula to compute the LTE. We show that our approach has the ability to solve problems that state-of-the art methods cannot.

References

[1] N. S. Nedialkov, K. R. Jackson, and G. F. Corliss, Validated solutions of IVPs for ordinary differential equations, *Applied Mathematics and Computation*, 105(1):21 – 68, 1999.

[2] E. Hairer, S. Norsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer-Verlag, 1993.

[3] J. Alexandre dit Sandretto and A. Chapoutot, Validated Solution of IVP for ODE based on Explicit and Implicit Runge-Kutta Schemes, *Research report ENSTA ParisTech*, 2015.

[4] J. C. Butcher, Coefficients for the study of Runge-Kutta integration processes, *Journal of the Australian Mathematical Society*, 3:185–201, 1963.

[5] O. Bouissou, A. Chapoutot and A. Djoudi, Enclosing Temporal Evolution of Dynamical Systems Using Numerical Methods, *NASA Formal Methods*, LNCS volume 7871, Springer Verlag, 2013.

Distributed localization and control of a group of underwater robots using contractor programming

L. Jaulin, S. Rohou, J. Nicola,

M. Saad, F. Le Bars and B. Zerr,

ENSTA Bretagne, Lab-Sticc, rue François Verny 29200 Brest

{luc.jaulin}@ensta-bretagne.fr

Keywords: Interval analysis, distributed localization, group of robots, nonlinear control

Introduction

We consider the problem of localizing a group of underwater robots and to control the group in order to accomplish a survey. We assume here that

1. When a robot surfaces, it can use the GPS for its localization.
2. The robots can communicate with a very low symbol rate.
3. The robots can measure their distances with a given accuracy, but not the direction of arrival.
4. Some outliers on the distances could occur, but their numbers is limited.

5. The localization process should be fast.

6. The robots have to use their estimated location to control their trajectory

We propose here to use a contractor programming method [Cha09] to solve the problem.

Main approach

We assume that we have $\bar{\ell}$ robots and that the motion of the i^{th} robot is described by a state equation of the form $\mathbf{x}_{k+1}^j = \mathbf{f}(\mathbf{x}_k^j, \mathbf{u}_k^j)$ where $\mathbf{x}_k^j$ is the state vector of the robot at time k, and $\mathbf{u}_k^j$ is the input vector. The input vector corresponds to the proprioceptive sensors (speed, heading, actuators) and is assumed to be known with some accuracy. Moreover, the robots are able to collect some intrinsic measurements of the form $\mathbf{y}_k^{j,\ell} = \mathbf{g}(\mathbf{x}_k^j, \mathbf{x}_k^\ell)$. By *intrinsic*, we mean that the measurements are not related to the environment, but to the group itself. For instance, $\mathbf{g}(\mathbf{x}_k^j, \mathbf{x}_k^\ell)$ may correspond to the distance between to the robot j and the robot ℓ et time k. We also assume that each robot may also collect some *extrinsic* data which are related to the environment (distance to a landmark, for instance). This corresponds to an observation function of the form $\mathbf{z}_k^j = \mathbf{h}(\mathbf{x}_k^j)$. We assume that $\mathbf{u}_k^j, \mathbf{y}_k^{j,\ell}, \mathbf{z}_k^j$ all belong to some boxes $[\mathbf{u}_k^j], [\mathbf{y}_k^{j,\ell}], [\mathbf{z}_k^j]$. These boxes could be small for high quality sensors and could be equal to $\mathbb{R}^n$ when no information is available. For each robot $\mathcal{R}_j$, at time k, we define a distributed CSP (Constraint Satisfaction Problem) [Mou12] as follows.

Variables. The variables are all states on a time window of length $\bar{h}$, i.e, $\mathbf{x}_h^j$, $h \in k - \bar{h}, \ldots, k+1\}$.

.

Constraints. The constraints are the following

$$\left\{ \begin{array}{rcccc} \mathbf{x}_{h+1}^j & = & \mathbf{f}(\mathbf{x}_h^j, [\mathbf{u}_h^j]) & & (E_x^{h,j}) \\ [\mathbf{y}_h^{j,\ell}] & = & \mathbf{g}(\mathbf{x}_h^j, [\mathbf{x}_h^\ell]),\ \ell \neq j & & (E_y^{h,j,\ell}) \\ [\mathbf{z}_h^j] & = & \mathbf{h}(\mathbf{x}_h^j) & & (E_z^{h,j}) \end{array} \right.$$

where $h \in \{k-\bar{h},\ldots,k\}$.and $\ell \in \{1,\ldots,\bar{\ell}\}$. In these equation, when we write "$[\mathbf{y}_h^{j,\ell}] = \mathbf{g}(\mathbf{x}_h^j,[\mathbf{x}_h^\ell])$", we mean "$\exists \mathbf{y}_h^{j,\ell} \in [\mathbf{y}_h^{j,\ell}], \exists \mathbf{x}_h^\ell \in [\mathbf{x}_h^\ell]$, $\mathbf{y}_h^{j,\ell}\mathbf{g}(\mathbf{x}_h^j,[\mathbf{x}_h^\ell])$". For each constraint $E_x^{h,j}, E_y^{h,j,\ell}, E_z^{h,j}$, we associate a contractor $C_x^{h,j}, C_y^{h,j,\ell}, C_z^{h,j}$. A contractor programming approach [Cha09] can then be used in order to perform the localization. In our CSP, we have two types of domains: the *internal domains* of the CSP: $[\mathbf{x}_h^j], h \in \{k-\bar{h},\ldots,k+1\}$ and the *external domains* $[\mathbf{u}_h^j]$,$[\mathbf{y}_h^{j,\ell}]$,$[\mathbf{z}_h^j]$,$[\mathbf{x}_h^\ell]$,$\ell \neq j, h \in \{k-\bar{h},\ldots,k\}$. There is no need to contract these external domains, and the robot $\mathcal{R}_j$ will not try to contract them.

Communication. Each robot $\mathcal{R}_j$ contracts its own domain $[\mathbf{x}_k^j]$ and broadcasts this information through the network. Since under the water, the communication rate is very low, each robot $\mathcal{R}_j$ only broadcasts the box $[\mathbf{x}_k^j]$ at time k.

Initialization. At the initial time $k = 0$, all variables are initialized with $\mathbb{R}^n$. When we switch from k to $k+1$, we remove the variable $\mathbf{x}_{k-\bar{h}}^j$ from the CSP of each robot and we add the variable $\mathbf{x}_{k+2}^j$ with the domain $[\mathbf{x}_{k+2}^j]= \mathbb{R}^n$.

Range-only distributed localization and control

As an example, we will consider some underwater robots moving in the ocean [Dre13]. The intrinsic observations correspond to the distances between robots and the extrinsic observations correspond to the GPS available when a robot surfaces. We assume that at most q outliers in the intrinsic measurements could occur within a time window of length $\bar{h}$ [Leg10]. The following contractor, is implemented in each robot $\mathcal{R}_j$:

$$C^{k,j} = C_{x,z}^{k,j} \bigcap C_{x,y}^{k,j}$$

where

$$C_{x,z}^{k,j} = \bigcap_{h\in\{k-\bar{h},\ldots,k\}} \left(C_x^{h,j} \circ C_z^{h,j}\right)$$

and

$$C_{x,y}^{k,j} = \bigcap_{\substack{h \in \{k-\bar{h},\ldots,k\} \\ \ell \neq j}}^{\{q\}} \left(C_x^{h,j} \circ C_y^{h,j,\ell}\right).$$

We will show that $C^{k,j}$ will not remove the true position for the robot. For the control, a vector field approach will be considered. A test case will also be presented in order to illustrate the efficiency of the approach.

References

[**Mou12**] F. Mourad, H. Snoussi, M. Kieffer and C. Richard, Robust interval-based localization algorithms for mobile sensor networks, *International Journal of Distributed Sensor Networks,* 1-7, 2012.

[**Leg10**] J. Leger and M. Kieffer, Guaranteed robust distributed estimation in a network of sensors, in *Proc. ICASSP'10*, 2010.

[**Dre13**] V. Drevelle, L. Jaulin and B. Zerr, Guaranteed Characterization of the Explored Space of a Mobile Robot by using Subpavings, NOLCOS, 2013.

[**Cha09**] G. Chabert and L. Jaulin, Contractor programming. Artificial Intelligence, 2009.

A new class of iterative interval methods for solving linear parametric systems

Lubomir Kolev

Technical University of Sofia
Faculty of Automatica, 1000 Sofia, Bulgaria
lkolev@tu-sofia.bg

Keywords: linear interval parametric systems, parameterized solution, iterative methods

Introduction

The linear interval parametric (LIP) systems considered in the talk include

$$A(p) = A^{(0)} + \sum_{\mu=1}^{m} A^{(\mu)} p_\mu,\ b(p) = b^{(0)} + \sum_{\mu=1}^{m} b^{(\mu)} p_\mu,\ p_\mu = [-1, 1] \quad (1)$$

where $A^{(\mu)}$ $\mu = 0, 1, \ldots, m$ are $n \times n$ real matrices and $b^{(\mu)}$ $\mu = 0, 1, \ldots, m$ are real column vectors. As is well known, the following "interval solutions" to (1) are of interest: outer interval (OI) solution $\mathbf{x}$, interval hull (IH) solution $\mathbf{x}^*$ and inner estimation of the hull (IEH) solution ξ. It should be stressed that all known methods for determining OI or IEH solutions (and, hence, the IH solution) yield the solution sought in the form of an interval vector.

A new type of solution $\mathbf{x}(p)$ to the LIP system (1) (called parameterized or p-solution) has been recently introduced in [1]. It is of the following parametric form

$$\mathbf{x}(p) = c + Lp + \mathbf{s},\ p \in \mathbf{p}$$

where L is a real $n \times n$ matrix while where c and $\mathbf{s}$ are a real and interval symmetric vectors, respectively. The new solution $\mathbf{x}(p)$, $p \in \mathbf{p}$ has a

number of useful properties: using it one can determine comparatively narrower $\mathbf{x}$ and ξ as well as small intervals containing the lower and upper ends of each component of $\mathbf{x}^*$. Combined with a constraint satisfaction technique, it permits determination of $\mathbf{x}^*$ as well as the global solution of certain equality-constrained optimization problems [1]. An iterative method for determining $\mathbf{x}(p)$ was suggested in [1] which is obtained by modifying each step of a known iterative method (Reference[11] in [1]) for computing $\mathbf{x}$.

The objective of the present talk is to show that any known iterative method for determining $\mathbf{x}$ can be modified in a unified manner as to produce a corresponding method for determining $\mathbf{x}(p)$. Thus, a whole new class of iterative methods for solving (1) can be constructed.

Iterative scheme

The unified iterative scheme, applicable for any method belonging to the new class, will be illustrated using the fixed-point representation of (1). Hence, the iterative process is

$$x^{(k+1)}(p) = \left(I - A^{(0)} + \sum_{\mu} p_\mu A^{(\mu)} \right) x^{(k)}(p) + \sum_{\mu} b^{(\mu)} p_\mu, \; k \geq 0, \; x^{(0)} = x^0$$

where x^0 is the solution of (3) for $p = 0$. As in [1], it can be shown that each term $x^{(k)}(p)$ in (3) can be enclosed by the linear interval form

$$\mathbf{l}^{(k)}(p) = c^{(k)} + L^{(k)} p + \mathbf{s}^{(k)}, \; p \in \mathbf{p}.$$

It can be proved that if the sequence $\mathbf{l}^{(k)}(p)$, $k \geq 1$ is convergent to a limit $\mathbf{l}^{(\infty)}(p)$, then:

(i) the interval vector

$$\mathbf{x} = \mathbf{l}^{(\infty)}(\mathbf{p})$$

($\mathbf{l}^{(\infty)}(\mathbf{p})$ is the range of $\mathbf{l}^{(\infty)}(p)$) is an OI solution to (1);

(ii) the linear interval form

$$\mathbf{x}(p) = \mathbf{l}^{(\infty)}(p) = x^{(0)} + c^{(\infty)} + L^{(\infty)}p + \mathbf{s}^{(\infty)}$$

determines a p-solution to (1);

(iii) the matrix $A(p)$ is non-singular for each $p \in \mathbf{p}$.

The actual iterative method is implemented using Rump's epsilon-inflation technique so the numerical complexity of the method is polynomial.

Detailed analysis shows that each new iterative method based on the use of p-solutions is superior to the corresponding original method as regards conservatism of the results and applicability radius of the methods.

Also, this approach can be applied to implementing new hull consistency algorithms treating several equations simultaneously.

References

[1] L. Kolev, Parameterized solution of linear interval parametric systems, *Applied Mathematics and Computation* 246(11):229-246, 2014.

On feedback target control for uncertain discrete-time systems through polyhedral techniques

Elena K. Kostousova

N.N. Krasovskii Institute of Mathematics and Mechanics,
Ural Branch of the Russian Academy of Sciences,
16, S.Kovalevskaja street, Ekaterinburg, 620990, Russia

kek@imm.uran.ru

Keywords: discrete-time systems, uncertain systems, state constraints, control synthesis, parallelotopes, parallelepipeds, interval analysis

Introduction

Problems of feedback target control for linear and bilinear dynamical discrete-time systems under uncertainties and state constraints are considered. There are known approaches to solving problems of this kind, including ones for differential systems, based on construction of solvability tubes (Krasovskii's bridges). Since practical construction of such tubes may be cumbersome, different numerical methods were devised. Among them constructive computation schemes for linear systems based on the ellipsoidal techniques were proposed (see, for example, [1,2]) and then expanded to the polyhedral techniques [3,4]. Such methods are ideologically close to interval analysis. Their main advantage is that they allow to find solutions by rather simple means.

Main results

Here we continue the development of methods of control synthesis for discrete-time systems using polyhedral (parallelotope-valued) solvability tubes. The paper deals with two types of problems, where the

controls appear either additively or in the system matrix (i.e., in the coefficients of the system). Both problems are considered for systems with parallelotope-bounded additive uncertainty and with interval uncertainties in the coefficients. Moreover the systems are considered under constraints on the state, where the state constraints are described in terms of zones (i.e., intersections of strips). The techniques for calculation of the polyhedral solvability tubes by the recurrent relations are presented. Control strategies, which can be constructed on the base of the mentioned polyhedral tubes, are proposed. In contrast to [3,4], these control strategies can be calculated by explicit formulas. Results of computer simulations are presented.

Acknowledgement

The research is partially supported by the Program of Basic Research of the Ural Branch of RAS (Project 15-16-1-8) and by the Russian Foundation for Basic Research (Grant 15-01-02368).

References

[1] A.B. Kurzhanski and I. Vályi, *Ellipsoidal Calculus for Estimation and Control*, Birkhäuser, Boston, 1997.

[2] A.Yu. Vazhentsev, Internal ellipsoidal approximations for problems of the synthesis of a control with bounded coordinates, *Izv. Akad. Nauk Teor. Sist. Upr.* no.3: 70—77, 2000 [in Russian].

[3] E.K. Kostousova, Control synthesis via parallelotopes: optimization and parallel computations, *Optim. Methods Softw.* 14(4):267–310, 2001.

[4] E.K. Kostousova, On polyhedral estimates in problems of the synthesis of control strategies in linear multistep systems, *in: Algorithms and Software for Parallel Computations, Ross. Akad. Nauk Ural. Otdel., Inst. Mat. Mekh., Ekaterinburg,* vol.9: 84–105, 2006 [in Russian].

Presentation of a multithreaded interval solver for nonlinear systems

Bartłomiej Jacek Kubica

[1] Warsaw University of Technology, Institute of Control and Computation Engineering,
ul. Nowowiejska 15/19, 00-665, Warsaw, Poland
{bkubica}@elka.pw.edu.pl

Keywords: nonlinear systems, interval methods, heuristics, narrowing, subdivision

Introduction

The paper presents the interval branch-and-prune solver for nonlinear underdetermined and well-determined systems. The system has been developed by the author. Its various aspects have been presented in a series of publications, i.a., [3]–[7].

Narrowing operators

Several tools are incorporated in our solver to process the boxes in the branch-and-prune process. They include:

- two versions of the interval Newton operator (componentwise Newton and Gauss-Seidel; see [4]),
- enforcing two versions of consistency – box-consistency and bound-consistency; see [7],
- an initial exclusion phase, based on Sobol sequences; see [5], [7],
- a procedure based on quadratic approximation of one of the equations; see [6].

For each of the tools a proper heuristic has been developed to decide whether using it on a specific box is worthwhile.

It is the opinion of the author that developing such heuristics is crucial for efficiency – and hence applicability – of the interval approach.

Parallelization

Branch-and-prune methods can be parallelized in a relatively simple manner – different boxes can be processed by different threads. The presented solver is implemented using this approach – Intel Threading Building Blocks (TBB) have been used for parallelization.

Usually, operations on a single box are performed by a single thread, but there are exceptions to this rule. In particular, enforcing bound-consistency is an intensive operation and it can be worthwhile to parallelize its execution on a single box.

Box subdivision

Usually, bisection of the longest edge is used in interval algorithms. In [4] and [7] other heuristics have been developed to choose the coordinate for bisection.

During the presentation at SWIM 2015, new results are going to be presented – including heuristics to choose between bi- and multisection, recommended by some authors (e.g., [1], [2]).

References

[1] S. Berner, Parallel methods for verified global optimization practice and theory, *Journal of Global Optimization* 9(1):1-22, 1996.

[2] D. Ratz and T. Csendes, On the selection of subdivision directions in interval branch-and-bound methods for global optimization, *Journal of Global Optimization* 7:183–207, 1995.

[**3**] B.J. Kubica, Interval methods for solving underdetermined nonlinear equations systems, *Reliable Computing* 15:207–217, 2011.

[**4**] B.J. Kubica, Tuning the multithreaded interval method for solving underdetermined systems of nonlinear equations, *Lecture Notes in Computer Science* 7204:467–476, 2012.

[**5**] B.J. Kubica, Excluding regions using Sobol sequences in an interval branch-and-prune method for nonlinear systems, *Reliable Computing* 19(4):385–397, 2014.

[**6**] B.J. Kubica, Using quadratic approximations in an interval method for solving underdetermined and well-determined nonlinear systems, *Lecture Notes in Computer Science* 8385:623–633, 2014.

[**7**] B.J. Kubica, Presentation of a highly tuned multithreaded interval solver for underdetermined and well-determined nonlinear systems, *Numerical Algorithms*, accepted for publication, `http://dx.doi.org/10.1007/s11075-015-9980-y`.

One ill-posed estimation problem of experimental process parameters. Interval approach

Sergey I. Kumkov[1,2]

[1] Institute of Mathematics and Mechanics, Ural Branch of Russian Academy of Sciences, S. Kovalevskaya str., 16, 620990, Ekaterinburg, Russia
[2] Ural Federal University, Institute of Radioelectronics and Informational Technologies, Mira str., Ekaterinburg, Russia
kumkov@imm.uran.ru

Keywords: experimental process, measurements, bounded error, parameters, estimation, interval approach, approximate *a priori* data

Introduction and problem formulation

In a chemical experiment, the following dependency of a reagent activity is investigated:

$$P(T, a, b, c) = T^2\, a\, b/c,\ a > 0,\ b > 0, c > 0, \tag{1}$$

where T is the temperature, the main variable, C°; $P(\cdot)$ is the reagent activity, dimensionless; a, b, and c are unknown constant parameters with dimensions, respectively, in moles, 1/mole, and $(\text{C}^\circ)^2$.

After experiment, the sample (with length N) of the activity measurements P_n is given

$$\{T_n, P_n\},\ n = \overline{1, N}, \tag{2}$$

where the temperature values are known exactly, but the values P_n are noised with the bounded measuring errors

$$P_n = P_n^* + e_n,\ |e_n| \le e_{\max},\ n = \overline{2, N},\ n = 1,\ T_1 = 0,\ P_1 = 0, \tag{3}$$

where P_n^* is an unknown true value under measuring; e_n is the error in the nth measurement; $e_{\max}$ is the bound (by modulus) on the maximal error value; $P_1 = 0$ is the *conditional exact* zero initial measurement.

Specifics of the experiment is in the following: the probabilistic characteristics of the noise are *absolutely unknown*, the sample is fatally short (usually, $N \sim 6 - 7$ measurements are only provided), and only

rough approximate *a priori* intervals of the parameters in (1) can be given

$$\begin{array}{l} \boldsymbol{a}^{\rm ap} = [\underline{\boldsymbol{a}}^{\rm ap},\ \overline{\boldsymbol{a}}^{\rm ap}],\ \boldsymbol{b}^{\rm ap} = [\underline{\boldsymbol{b}}^{\rm ap},\ \overline{\boldsymbol{b}}^{\rm ap}],\ \boldsymbol{c}^{\rm ap} = [\underline{\boldsymbol{c}}^{\rm ap},\ \overline{\boldsymbol{c}}^{\rm ap}], \\ 0 < \underline{\boldsymbol{a}}^{\rm ap} < \overline{\boldsymbol{a}}^{\rm ap},\ 0 < \underline{\boldsymbol{b}}^{\rm ap} < \overline{\boldsymbol{b}}^{\rm ap},\ 0 < \underline{\boldsymbol{c}}^{\rm ap} < \overline{\boldsymbol{c}}^{\rm ap}. \end{array} \quad (4)$$

Problem formulation: *Using data* (1)–(4), *construct the informational set* (*set-membership*) *for admissible* (*consistent*) *values of parameters* (1).

The problem is ill-posed since of the "stuck"-character of parameters in (1) and since it is impossible to validate application of standard statistical approaches to estimating such a set. So, here, the approach on the basis of the interval analysis has been used. Theoretical aspects of the approach is highlighted in [1], special algorithms and software for solving practical problems of the mentioned type has been elaborated and described in [2].

Interval procedures for estimating the process parameters and the main results

The following procedure of the interval approach are implemented.

1) The standard [1, 2] uncertainty intervals $\boldsymbol{H}_n = [P_n - e_{\max}, P_n + e_{\max}]$ are constructed for each measurement P_n from (3) by the given bound $e_{\max}$ (Fig.1a); and, introducing the auxiliary "joined" parameter $g = ab/c$, its corresponding informational interval $\boldsymbol{g} = [\underline{\boldsymbol{g}},\ \overline{\boldsymbol{g}}]$ is calculated [2]. Additionally, it is worthy to calculate the *a priori* interval of the parameter g and compare it with the obtained interval $\boldsymbol{g}$ for analysis of *consistency* of the *a priori* data (4) on parameters a, b, and c with the given sample of measurements (3).

2) Having the interval equation $ad = \boldsymbol{g}$, where $d = b/c$, solve it w.r.t. the auxiliary parameter d as follows: $\boldsymbol{d} = \boldsymbol{g}/\boldsymbol{a}^{\rm ap}$. As a result in the plane $a \times d$, we obtain the informational set $I(a, d)$ with the curve (hyperbolic) lower $\underline{Fr}_d(a)$ and upper $\overline{Fr}_d(a)$ boundaries (Fig.1b) as a functions of the parameter a values from its *a priori* interval $\boldsymbol{a}^{\rm ap}$. In Figure 1b, a cross-section of the set $I(a, d)$ is shown for value $a = 1.89$ mole; it is the interval $\boldsymbol{d}(1.89)$. In Figure 1b (at the left) the *a priori* interval of the auxiliary parameter d is shown (the thick dash-dotted

vertical segment) calculated by the *a priori* intervals $\boldsymbol{b}^{\rm ap}$ and $\boldsymbol{c}^{\rm ap}$. Here, the thick vertical line in dashes marks the outer interval of $I(a,d)$ on d for the *a priori* interval $\boldsymbol{a}^{\rm ap}$. Comparison of these two intervals allows one to check out *consistency* of the *a priori* data (4) on parameters a, b, and c with the given sample of measurements (3).

3) For each value $a \in \boldsymbol{a}^{\rm ap}$ we have the interval $\boldsymbol{d}(a)$ (Fig. 1b). So, it becomes possible to construct the informational set $I_a(b,c)$ (Fig.1c) of admissible values for parameters b, c for each value of the parameter a. It is seen (Fig.1c) that the set $I_a(b,c)$ is composed of intersection of the rectangle $\boldsymbol{b}^{\rm ap} \times \boldsymbol{c}^{\rm ap}$ with the cone between the lower $\underline{c}(b, 1/\overline{\boldsymbol{d}}(a)$ and upper $\overline{c}(b, 1/\underline{\boldsymbol{d}}(a)$ rays for $a \in \boldsymbol{a}^{\rm ap}$ and $b \in \boldsymbol{b}^{\rm ap}$. In Figure 1c, the set $I_a(b,c)$ (shadowed five-apex polygon) is shown for value $a = 1.89$ mole and corresponding interval $\boldsymbol{d}(1.89)$ from Fig.1b.

Conclusions

In the considered ill-posed estimation problem with the "stuck" parameters and under absence of probabilistic characteristics of the measuring errors, the elaborated interval approach allows one to analyze consistency of the given sample of measurements itself, to analyze consistency of the given sample of measurements and the given *a priori* data, and to construct the informational set of admissible values of the "stuck" parameters.

Acknowledgment

The work was supported by the RFBR Grant proj. 15-01-07909.

References

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

[**2**] S.I. Kumkov and Yu. Mikushina, Interval Approach to Identification of Catalytic Process Parameters, *Reliable Computing*, 19: 197–214, 2013.

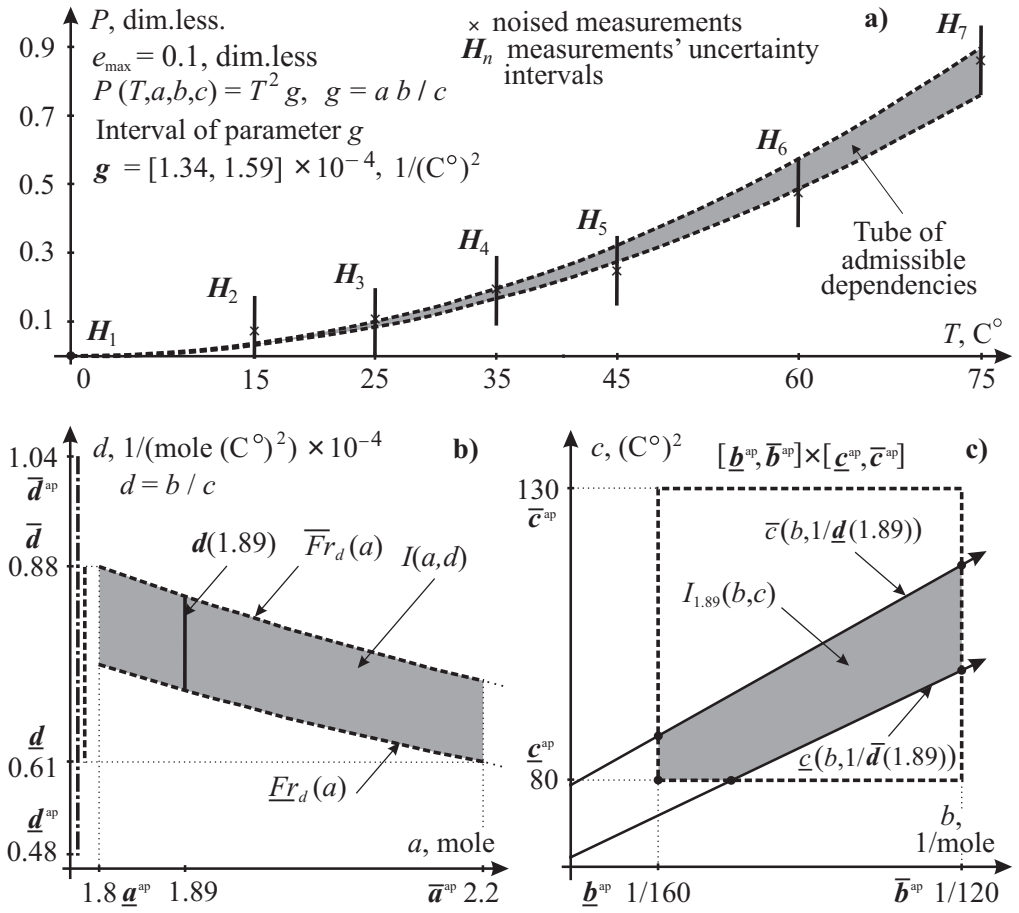


Figure 1: **a)** Input noised measurements (crosses); uncertainty intervals (thin vertical lines); tube of admissible dependencies (shadowed curve sector); the admissible interval of the parameter $\boldsymbol{g} = [1.34, 1.59] \times 10^{-4}$; the bound $e_{\max} = 0.1$ on the noise level; **b)** The set $I(a, d)$ of possible values for parameters a and d; the lower $\underline{Fr}_d(a)$ and upper $\overline{Fr}_d(a)$ boundaries; **c)** After taking into account *a priori* intervals $\boldsymbol{b}^{\rm ap}$ of parameter b and $\boldsymbol{c}^{\rm ap}$ of parameter c, the shadowed region is the informational set $I_{1.89}(a, d)$ of parameters b and c for the value of parameter $a = 1.89$ mole

EASIBEX-MATLAB: a simple tool to begin with interval contractors

Fabrice Le Bars[1], Jeremy Nicola[1,2] and Luc Jaulin[1]

[1] ENSTA Bretagne (ex ENSIETA), STIC/OSM, Lab-STICC/CID/IHSEV,
2 Rue F. Verny, 29806, Brest Cedex 9, France
{fabrice.le_bars, luc.jaulin}@ensta-bretagne.fr
[2] iXBlue,
Rue Rivoalon, Sainte Anne du Portzic, 29200, Brest, France
jeremy.nicola@gmail.com

Keywords: EASIBEX, MATLAB, interval, contractors

Introduction

EASIBEX-MATLAB is a simple tool to start using interval arithmetic and contractors. It uses IBEX (see [1]) as internal library. It is designed for people that do not feel comfortable with C++, Python, object-oriented programming, but want to quickly prototype and test programs using interval arithmetic and contractors, while manipulating easily the results thanks to MATLAB.

Description

The main goals of EASIBEX-MATLAB are the following:

- Start using interval arithmetic and contractors.
- Quickly prototype and test new algorithms.

The target users that would find EASIBEX-MATLAB useful are:

- Students.

- Scientists that do not know advanced computer science languages and paradigms such as C++, Python and object-oriented programming, but would like to use classical interval algorithms for their own problems, or to prototype new algorithms.

The base and philosophy of EASIBEX-MATLAB can be sum up as these ideas:

- It is designed to be a very simple MATLAB layer of IBEX to benefit from several advanced, efficient and already tested algorithms.
- The naming conventions and way of use is strongly based on the very simple interval library used in several existing samples (see [2], [3]).
- It can be also used with VIBes ([4]) to easily draw the results of interval computations.

Due to its design, EASIBEX-MATLAB has some inherent limitations:

- To keep it simple, not all the features provided by IBEX are available.
- Even if the computations are made by IBEX, no study has been made to check if the guarantee of the results (w.r.t. rounding, etc.) is lost when passing parameters and retrieving results through MATLAB functions and shared library calls.
- The function calls are simplified and different from IBEX to avoid the difficulties of object-oriented paradigm (IBEX uses notions such as inheritance, polymorphism, etc.).

If C++ is needed, EASIBEX-CPP provides a very simple way to start using IBEX and benefit from its features in C++, without extended knowledge of object-oriented programming. See [5] for several

guides and examples. Once you are comfortable with the notions of intervals, contractors as well as C++, you can be more efficient by using directly IBEX.

Quick tutorial

To start using EASIBEX-MATLAB, download and extract
https://github.com/ENSTABretagneRobotics/EASIBEX-MATLAB/archive/master.zip:

- In MATLAB, go to File\Set Path...\Add Folder... and add this folder as well as x86 folder if you use MATLAB 32 bit or x64 folder if you use MATLAB 64 bit.
- Run sivia_easibex.m to test. A red ring on a blue background and with yellow borders should appear.

To define an EASIBEX-MATLAB interval :

```
x=[-2,2]
```

`x(1,2)` would be 2. An empty interval would be:

```
x=[NaN,NaN]
```

and infinity:

```
x=[-Inf,Inf]
```

To define a box:

```
x=[[-2,2];[2,4];[-4,1]]
```

`x(2,:)` would be $[2, 4]$.

2 intervals can be added using:

```
Z=i_Add([0,2],[-1,2])
```

and 2 boxes:

```
Z=i_Add([[0,1];[0,10];[0,10]],[[-1,0];[2,5];[-1,0]])
```

To contract 3 intervals $Z = [-10, 1]$, $X = [0, 2]$, $Y = [-1, 2]$ knowing the constraint $Z = X + Y$:

```
[Z,X,Y]=i_Cadd([-10,1],[0,2],[-1,2])
```

To contract the vector $\mathbf{x} = [-10, 10] \times [0, 10] \times [-10, 0]$ w.r.t. the q-relaxed intersection (see e.g. [6]) of the 4 vectors $[-2, 2] \times [2, 4] \times [-4, 1]$, $[-1, 5] \times [-5, 8] \times [-7, 2]$, $[-1, 1] \times [0, 2] \times [1, 2]$, $[-2, 2] \times [2, 8] \times [-1, 2]$, with $q = 2$:

```
x = [[-10,10];[0,10];[-10,0]]
y_j = {[[-2,2];[2,4];[-4,1]];[[-1,5];[-5,8];[-7,2]];
[[-1,1];[0,2];[1,2]];[[-2,2];[2,8];[-1,2]]}
x = i_C_q_in(x, 2, y_j)
```

Quick reference

The current operations, functions and contractors available include $+$, $-$, $*$, $/$, intersection, union, $\sqrt{}$, exp, sin, arctan, min, abs, sign, determinant, scalar product, norm, distance, q-intersection, not inside, inside segment, inside circle, inside ring...

References

[1] G. Chabert, *IBEX, an Interval-Based EXplorer*, http://www.ibex-lib.org.

[2] http://www.ensta-bretagne.fr/jaulin/

[3] https://github.com/ENSTABretagneRobotics/interval/

[4] https://github.com/ENSTABretagneRobotics/VIBES/

[5] https://www.ensta-bretagne.fr/jaulin/easibex.html

[6] L. Jaulin (2009), *Robust set membership state estimation ; Application to Underwater Robotics*, Automatica 45.

A linear iterative interval method for computing the generalized inverse of an matrix

Taher Lotfi, Katayoun Mahdiani

Department of Applied Mathematics, Hamedan Branch,
Islamic Azad University, Hamedan, Iran
lotfitaher@yahoo.com, lotfi@iauh.ac.ir

Keywords: Interval matrix, Iterative method, Generalized Inverse, Linear Convergence,

Introduction

There are three principal situations in which it is required to obtain numerically a generalized inverse of a given matrix [1]:

(i) the case in which any $\{1\}$-inverse will suffice;

(ii) the cases in which any $\{1,3\}$-inverse (or sometimes any $\{1,4\}$-inverse) will do; and

(iii) the case in which a $\{2\}$-inverse having a specified range and null space is required.

The inverse desired in case (iii) is, in the majority of cases, the Moore–Penrose inverse, which is the unique $\{2\}$-inverse of the given matrix A having the same range and null space as A^*. The Drazin inverse can also be fitted into this pattern, being the unique $\{2\}$-inverse of A having the same range and null space as A^l, where l is any integer not less than the index of A. When $l = 1$, this is the group inverse. Generalized inverses are closely associated with linear equations, orthonormalization, least-squares solutions, singular values, and various matrix factorizations. In particular, the QR-factorization and the Singular

Value Decomposition (SVD) figure prominently in the computation of the Moore–Penrose inverse.

Zhan et al. [2] suggested an interval iterative method for computing Moore–Penrose inverse of the full row (or column) rank matrix. Motivated by this work, here, we will attempt to introduce a new linear iterative interval method for computing the {1}-inverse of a given arbitrary matrix A. It should be noted that the given matrix A is not itself an interval matrix, but the method is interval to enclosing its interval inverse as sharp as possible, like interval Newton's method [3] which can enclose any root of a given real nonlinear equation.

Main Results

In this section, we try to extend a new linear iterative of interval sequence, say $\{\mathbf{X}^{(k)}\}$, converging to {1}-inverse, denoted by $A^{\dagger}$, of a given matrix A. We recall that the order of convergence of such an iteration is defined by its corresponding *residuals* rate

$$\mathbf{R}^{(k)} = P_{R(A)} - A\mathbf{X}^{(k)}, \quad k = 0, 1, 2, \ldots, \tag{1}$$

where converges to $\mathbf{0}$ as $k \to \infty$, or $\mathbf{X}^{(k)} \to A^{\dagger}$. The iterative residual (1) is linear if there is a positive constant M such that

$$d(\mathbf{R}^{(k+1)}) \leq M\, d(\mathbf{R}^{(k)}), \quad k = 0, 1, 2, \ldots.$$

Suppose that $\rho(\mathbf{R}^{(0)}) = \rho(P_{R(A)} - A\mathbf{X}^{(0)}) < 1$. We now consider the following interval extension of iterative method for computing {1}-inverse

$$\mathbf{X}^{(k+1)} = \left(\mathbf{X}_c^{(k)} + \mathbf{X}_c^{(0)}(I - A\mathbf{X}^{(k)})\right) \cap \mathbf{X}^{(k)}, k = 0, 1, 2, \ldots, \tag{2}$$

where $A \in C^{m\times n}$, $\mathbf{X}_c^{(0)} \in R(A^*, A^*)$, i.e., $A^*BA^* \subset \mathbf{X}^{(0)}$ for some $B \in C^{m\times n}$, and $\mathbf{X}_c^{(k)} = \mathrm{mid}(\mathbf{X}^{(k)})$. Evidently, $\mathbf{X}^{(k+1)} \subset \mathbf{X}^{(k)}$, and because of inclusion monotonicity, $A^{\dagger} = \lim \mathbf{X}_c^{(k)} \in \mathbf{X}^{(k)}$ as $k \to \infty$.

Taking into account (2), we can write

$$\begin{aligned}\mathbf{X}^{(k+1)} &= \left(\mathbf{X}_c^{(k)} + \mathbf{X}_c^{(0)}(\mathbf{R}^{(k)})\right) \cap \mathbf{X}^{(k)} \\ &= \left(\mathbf{X}_c^{(k)} + \mathbf{X}_c^{(0)}(P_{R(A)} - A\mathbf{X}^{(k)})\right) \cap \mathbf{X}^{(k)}, \quad k = 0, 1, \ldots. \end{aligned} \tag{3}$$

Consequently,

$$\begin{aligned}\mathbf{R}^{(k+1)} &= \left(P_{R(A)} - \mathbf{X}^{(k+1)}\right) \cap \mathbf{R}^{(k)} \\ &= \left(P_{R(A)} - A\mathbf{X}_c^{(k)} - A\mathbf{X}_c^{(0)}\mathbf{R}^{(k)}\right) \cap \mathbf{R}^{(k)} \\ &= \left(\mathbf{R}_c^{(k)} - A\mathbf{X}_c^{(0)}\mathbf{R}^{(k)}\right) \cap \mathbf{R}^{(k)}, \quad k = 0, 1, \ldots. \end{aligned} \tag{4}$$

It follows from (4) that

$$d(\mathbf{R}^{(k+1)}) \leq M d(\mathbf{R}^{(k)}), \tag{5}$$

for some $M \geq 0$. Since we have assumed that $\rho(\mathbf{R}^{(0)}) < 1$, the sequence of residuals converges to the zero matrix as k approaches to infinity. Accordingly, the sequence (1) converges. Finally, we have $A\mathbf{R}^{(\infty)} = P_{R(A)}$, since $P_{R(A)} - d(\mathbf{R}^{(k)}) \to O$, as $k \to \infty$. In particular, we have proved $\mathbf{R}^{(\infty)}$ is a $\{1\}$-inverse of A.

To obtain a second order convergence of interval iterative method, it suffices to consider the following modification of method (2)

$$\mathbf{X}^{(k+1)} = \left(\mathbf{X}_c^{(k)} + \mathbf{X}_c^{(k)}(I - A\mathbf{X}^{(k)})\right) \cap \mathbf{X}^{(k)}, k = 0, 1, 2, \ldots, \tag{6}$$

Although this modification increases convergence rate from one to two, however, we need to do more computation per iterate.

To sum up, we have claimed that we have developed a new method for computing an interval sequence that includes generalized inverses of a given rectangular matrix. Our method seems to be linear, and developing it to higher convergence method is straightforward. We hope this study shed new light on this field of linear algebra which there is little research devoted to it. Based on our best knowledge,

there was only paper in this field and we have cited it here, [2]. We would be grateful if someone knows about this topic and inform us. We believe that much remains to be done in the area of interval methods for generalized inverses.

References

[1] A. B. Israel, T.N.E. Greville, (2003) Generalized Inverses: Theory and Applications, Springer-Verlag New York, Inc.

[2] X. Zhang, J. Cai, Y. Wei, Interval iterative methods for computing Moore–Penrose inverse, Applied Mathematics and Computation 183 (2006) 522–532.

[3] R.E. Moore, (1979) Methods and Applications of Interval Analysis, SIAM Studies in Applied Mathematics, Philadelphia.

Parameter identification with hybrid systems in a bounded-error framework

Moussa Maïga[1,2], Nacim Ramdani[1]
and Louise Travé-Massuyès[2]

[1] Univ. Orléans, INSA-CVL, PRISME, EA 4229, F45072, Orléans, France
{nacim.ramdani}@univ-orleans.fr
[2] CNRS, LAAS, University of Toulouse, 31031 Toulouse, France
mmaiga@laas.fr, louise@laas.fr

Keywords: hybrid systems, reachability, set inversion.

Hybrid systems exhibit continuous and discrete dynamics and are encountered in many complex and safety-critical systems. Robust parameter identification is an important step for monitoring, control and fault detection.

We consider the "unknown but bounded error" framework where all the uncertain quantities (measurement errors, modeling errors and uncertainty) are taken in a bounded set with known bounds. With fault detection in mind, we then introduce a set-membership method to address parameter identification in the latter framework for hybrid dynamical systems with nonlinear dynamics and nonlinear guards and invariants. One of the main advantages of the set-membership estimation approach is that it provides a guaranteed decision about fault occurrence, in contrast with the classical notion of risk, usually defined in terms of probability of occurrence and false detection. In other words, these methods allow us to avoid false positive (false alarm).

To develop the parameter identification method, we foster on our recently proposed algorithms for performing hybrid reachability [1-2-3], which combine interval Taylor methods for continuous reachability with techniques to solve event detection and localization in hybrid systems. We then embed the latter hybrid reachability algorithm within SIVIA algorithm to obtain an algorithm that can solve the set inver-

sion problem underlying parameter identification with hybrid dynamical systems [4].

An illustrative example will be given which shows that our method can naturally reconstruct both inner and outer approximations of the parameter solution set, for parameters acting either on the continuous dynamics (ODE) or on the event (guard conditions and invariants).

Acknowledgement

This work is supported by the French National Research Agency under contract ANR 2011 INS 006 "MAGIC-SPS" (projects.laas.fr/ANR-MAGIC-SPS).

References

[**1**] M. Maïga, N. Ramdani, and L. Travé-Massuyès, A fast method for solving guard set intersection in nonlinear hybrid reachability, In *Proc. of 52nd IEEE Conference on Decision and Control*, 508–513, 2013.

[**2**] M. Maïga, C. Combastel, N. Ramdani, and L. Travé-Massuyès, Nonlinear hybrid reachability using set integration and zonotope enclosures, In *Proceedings of the 13th European Control Conference, pp. 234-239*, 2014.

[**3**] M. Maïga, N. Ramdani, and L. Travé-Massuyès, A CSP versus a zonotope-based method for solving guard set intersection in nonlinear hybrid reachability, *Mathematics in Computer Science*, 8(3-4), 2014.

[**4**] M. Maïga, N. Ramdani, and L. Travé-Massuyès, Robust fault detection in hybrid systems using set-membership parameter estimation, In *Proc. of 9th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, 2015.

Viability kernel computation based on interval methods

Dominique Monnet, Luc Jaulin, Jordan Ninin, Alexandre Chapoutot, Julien Alexandre-dit-Sandretto

[1] ENSTA Bretagne, Lab-STIC,
2 rue François Verny, 29806 Brest, France
dominique.monnet@ensta-bretagne.org

Keywords: Viability Kernel, Guaranteed Integration, Lyapunov Theory

Introduction

Since viability theory has been introduced by Jean-Pierre Aubin [1], almost exclusively discrete methods have been developed to approximate the viability kernel. We approached the computation of viability kernel with several methods based on interval analysis. Using guaranteed integration, we are able to compute a guaranteed kernel, which is not or hardly achievable with discrete methods.

Problem statement

Let us consider a dynamic system $\dot{\mathbf{x}} = f(\mathbf{x}, u)$, where x is the state vector, f the evolution function and u the control vector. We assume that $u \in \mathcal{U}$, where $\mathcal{U}$ is the set of possible control of the system. Let us define the flow map $\varphi(\mathbf{x_0}, u, t) = x(t)$, where $\mathbf{x}(t)$ is the solution to the evolution function with the initial condition $\mathbf{x}(0) = \mathbf{x_0}$. Given a subset $\mathcal{K}$ of the state space of the system, the viability kernel problem consists of characterizing the subset $\mathcal{C}$ of $\mathcal{K}$ such as for all state vector of $\mathcal{C}$ the system can stay in $\mathcal{K}$. Thus, we have

$$\mathcal{C} = \{\mathbf{x_0} \in K, \exists u \in \mathcal{U} | \forall t, \varphi(x_0, u, t) \in \mathcal{K}\} \tag{1}$$

Resolution Tools

We characterize the viability kernel using the following tools:

- $\mathcal{C}$ is approximated with an inner approximation $\mathcal{C}^-$ and an outer approximation $\mathcal{C}^+$, such as $\mathcal{C}^- \subset \mathcal{C} \subset \mathcal{C}^+$ [3].
- Lyapunov theory [2] is used to find ellipsoids of attraction of the system used to initialize $\mathcal{C}^-$.
- CSP programming is used to contract on the ellipsoids of attraction.
- Guaranteed integration of the evolution function is used to inflate $\mathcal{C}^-$ and reduce $\mathcal{C}^+$ such as the enclosure of $\mathcal{C}$ is thiner. If there exist a couple (u, t) such as $[\varphi]([\mathbf{x_0}], u, t) \subset \mathcal{C}^-$, with $[\mathbf{x_0}]$ a subset of $\mathcal{K} \backslash \mathcal{C}^-$, we add $[\mathbf{x_0}]$ to $\mathcal{C}^-$. If there exist t such as $[\varphi]([\mathbf{x_0}], \mathcal{U}, t) \cap \mathcal{K} = \emptyset$, which means no control exists such as the system can stay in $\mathcal{K}$ over the time, we remove $[\mathbf{x_0}]$ from $\mathcal{C}^+$.

Example

Let consider the "Car on the hill" two dimensional problem. The state equations are the following:

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = -9.81 sin(g(x_1)) - 0.7 x_2 + u \end{cases} \tag{2}$$

where $g(s) = (1.1 sin(1.2s) - 1.2 sin(1.1s))/2$ represents the shape of the hill. Using Lyapunov theory, we are able to find ellipsoids of attraction. Then we contract on these ellipsoid as shown on figure 1.

We use the green area to initialize $\mathcal{C}^-$, and we initialize $\mathcal{C}^+$ with $\mathcal{K}$ which corresponds the green and blue areas. Then, we integrate elements of the blue area over the time, trying to remove them from $\mathcal{C}^+$ or to add them to $\mathcal{C}^-$.

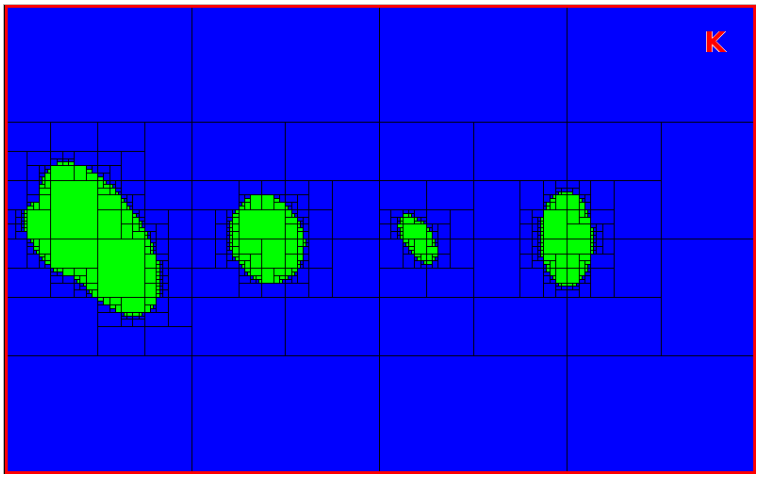


Figure 1: Results of contraction on the ellipsoids of stability. There is two ellipsoids overlapping on the left.

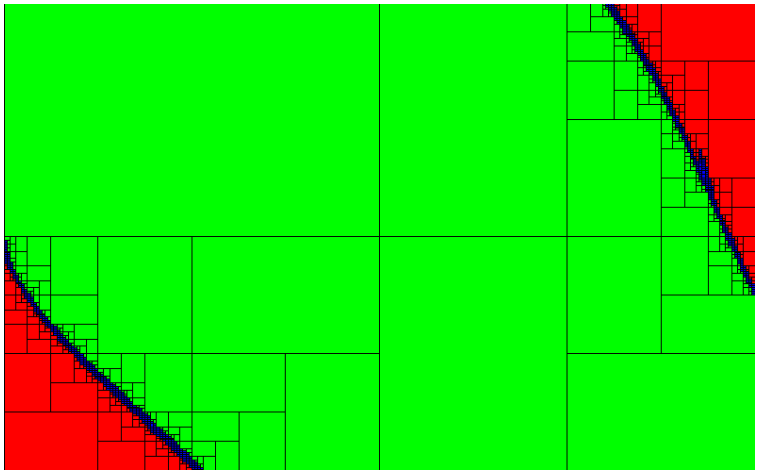


Figure 2: Computation of the viability kernel.

Figure 2 shows the result we obtain. $\mathcal{C}^-$ is shown in green, $\mathcal{C}^+$ corresponds to the green and blue areas. The red area contains the state vectors such that there is no feasible control such that the system always stays inside $\mathcal{K}$. The edge of the viability kernel is enclosed in the blue region, as $\mathcal{C}^- \subset \mathcal{C} \subset \mathcal{C}^+$. As the integration is guaranteed, the approximation of the viability kernel is also guaranteed.

References

[**1**] J.-P Aubin, *Viability theory*, Birkhäuser, Boston, 1991.

[**2**] N. Delanoue, L. Jaulin and B. Cottenceau, An algorithm for computing a neighborhood included in the attraction domain of an asymptotically stable point, *Communications in Nonlinear Science and Numerical Simulation* 21(1-3):181–189, 2015.

[**3**] M. Lhommeau, L Jaulin and L. Hardouin, Inner and outer approximation of capture basins using interval analysis, *ICINCO* 2007.

Mobile Robot Mapping using Interval Methods

Mohamed Mustafa[1], Eduard A. Codres[1], Alexandru Stancu[1]

[1] The University of Manchester, Control Systems Research Group
M13-9PL, Manchester, United Kingdom
{mohamed.mustafa, eduard.codres, alexandru.stancu}@manchester.ac.uk

Keywords: Interval methods, mobile robot, SLAM, nonlinear observation model

Introduction

For SLAM problem [3], building an accurate map leads to an accurate localization. We propose a guaranteed solution using interval methods for nonlinear observation model to work with holnomic robots with no rotation, where the map is proven to converge. Our approach does not require any assumptions with regard to the linearity of the observation model, nor its noise except that it needs to be bounded. We use interval methods to evaluate the domain of a function given the codomain and the function itself. This approach encapsulates all information in the current estimate, therefore, it is not necessary to keep track of all past observations. We will prove the convergence of the approach to the correct map as the robot moves in the environment over time, given that at each time step, at least one old landmark is observed, and the data association problem is assumed a solved problem.

Assumptions

In this work, we assume the following:

1. The robot is holonomic with no rotation and it is moving in a 2-D environment.

2. The robot is equipped with a LIDAR to measure the range $\rho_{m_i,t}$ and the bearing $\alpha_{m_i,t}$ of landmarks in the robot frame, where m_i is the position of the i^{th} landmark, $i = 1 : N$, and N is the total number of observed landmarks at time t.

3. Each measurements is uncertain with some known bounded noise in the range $[\omega_{\rho,m_i,t}]$ and in the bearing $[\omega_{\alpha,m_i,t}]$.

4. At least one old landmark is observed at each time step.

5. The robot can distinguish between different landmarks, i.e., data association problem is solved.

Concept

Let $d_{x,m_i,t}$ and $d_{y,m_i,t}$ be the distance between the robot and the i^{th} landmark in the x and the y directions, respectively. Then, the observation model of the sensor is in the form of:

$$z_{\rho,m_i,t} = \sqrt{d^2_{x,m_i,t} + d^2_{y,m_i,t}} \tag{1}$$

$$z_{\alpha,m_i,t} = \arctan 2(d_{y,m_i,t}, d_{x,m_i,t}) \tag{2}$$

The distance between landmark i and landmark j is defined as follows:

$$dl_{x,(i,j),t} = d_{x,m_i,t} - d_{x,m_j,t} \tag{3}$$

$$dl_{y,(i,j),t} = d_{y,m_i,t} - d_{y,m_j,t} \tag{4}$$

Since the measuremnts have some bounded uncertainty, both $z_{\rho,m_i,t}$ and $z_{\alpha,m_i,t}$ belong to *intervals*, such that $z_{\rho,m_i,t} \in [z_{\rho,m_i,t}]$ and $z_{\alpha,m_i,t} \in [z_{\alpha,m_i,t}]$. We define a constraint satisfaction problem (CPS) [1] using $dl_{x,(i,j),t}$, $dl_{y,(i,j),t}$ as *variables*, and Eq.(1-4) as *constraints*. We associate a contractor for each constraint such that: $\mathcal{C}_{z_{\rho,m_i,t}}^{d_{x,m_i,t},d_{y,m_i,t}}$ for Eq.(1), $\mathcal{C}_{z_{\alpha,m_i,t}}^{d_{x,m_i,t},d_{y,m_i,t}}$ for Eq.(2), $\mathcal{C}_{dl_{x,(i,j),t}}^{d_{x,m_i,t},d_{x,m_j,t}}$ for Eq.(3) and $\mathcal{C}_{dl_{y,(i,j),t}}^{d_{y,m_i,t},d_{y,m_j,t}}$ for Eq.(4), where $i, j = 1 : N$, and N is the number of observed

landmarks at time t. For example, if the robot observes two landmarks, then, we have 6 contractors associated with 6 constraints. Next, we use contractor programming [4] to solve for the distance between landmarks i and j, which is denoted by the subpaving $\mathbb{S}_{\mathbf{dl}_{(i,j)},t}$, where $\mathbf{dl} = [dl_x, dl_y]^T$. Since the landmarks are stationary in the environment, the distance estimate $\mathbb{S}_{\mathbf{dl},(i,j),t}$ must be consistent at any time t when landmarks i and j are observed, therefore, the following equation holds:

$$\cap_{t=1}^{T} \quad \mathbb{S}_{\mathbf{dl}_{(i,j)},t} \neq \phi \tag{5}$$

As the landmarks are observed frequently, we will show that the left-hand-side of Eq.(5) converges to the true distance $\mathbf{dl}_{(i,j)}$ between the i^{th} landmark and the j^{th} landmark as $t \to \infty$. If the position of one landmark $\mathbf{m}_i$ is known exactly (anchoring landmark), then, by using the distance $\mathbb{S}_{\mathbf{dl}_{(i,j)},t}$ we can estimate the position of the second landmark $\mathbf{m}_j$ as follows:

$$\mathbf{m}_j = \mathbf{m}_i - \mathbb{S}_{\mathbf{dl}_{(i,j)},t} \tag{6}$$

where $\mathbf{m} = [m_x, m_y]^T$, and the $''-''$ operator is overloaded for subtraction of set of vectors. Since $\mathbb{S}_{\mathbf{dl}_{(i,j)},t} \to \mathbf{dl}_{(i,j)}$, the map converges to the true map. If the anchoring landmark is not available, there will be some offset between the estimated map and the true map.

References

[1] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*, Sprinter, 2001, ISBN-13: 978-1852332198.

[2] L. Jaulin, A Nonlinear Set Membership Approach for the Localization and Map Building of Underwater Robots, *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 88-98, Feb. 2009.

[3] M. Montemerlo and S. Thrun and D. Koller and B. Wegbreit, FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

[**4**] L. Jaulin, Range-Only SLAM With Occupancy Maps: A Set-Membership Approach, *IEEE TRANSACTIONS ON ROBOTICS*, vol. 27, no. 5, pp. 1004–1020, 2011.

Gaussian Nonlinear set inversion

Jeremy Nicola[1,2] and Luc Jaulin[1]

[1] ENSTA Bretagne (ex ENSIETA), STIC/OSM, Lab-STICC/CID/IHSEV,
2 Rue F. Verny, 29806, Brest Cedex 9, France
{luc.jaulin}@ensta-bretagne.fr
[2] iXBlue,
Rue Rivoalon, Sainte Anne du Portzic, 29200, Brest, France
jeremy.nicola@ensta-bretagne.org

Keywords: interval, parameter estimation, state estimation, set-inversion, hybridization, contractors

Introduction

In this presentation, we treat the problem of estimating the parameters of a nonlinear model from experimental data in a reliable and precise manner. Using interval analysis, we are able to compute the set of all the parameters that are consistent with a given probability with the experimental data. Using statistical properties of the uncertainties associated with each measurement, we will show that a geometrical constraint can be extracted that enables us to drasticaly reduce size of the solution set.

Description

Let $\tilde{\mathbf{y}} \in \mathbb{R}^n$ be the vector of all the collected data, and $\mathbf{p} \in \mathbb{R}^m$ the parameters we want to estimate which parametrize a function $\mathbf{f}(\mathbf{p})$.

In the context of a bounded-error model, each measurement $\widetilde{y}_i$ is associated to an interval $[y_i]$ which is assumed to contain the true value y_i, and the vector of intervals $[\mathbf{y}]$ defines the set $\mathbb{Y}$ which is an n-dimensional axis-aligned box.

Then, the problem to be solved is characterizing the set $\mathbb{S}$ of all values of $\mathbf{p}$ consistent with the data:

$$\mathbb{S} = \{\mathbf{p} \in \mathbb{R}^m | \mathbf{f}(\mathbf{p}) \in \mathbb{Y}\} = \mathbf{f}^{-1}(\mathbb{Y})$$

which is a set-inversion problem and can be efficiently solved using interval analysis [1],[2].

These methods are reliable, in the sense that no approximation, no linearization is made on the model, and guarantees that not a single feasible solution will be lost.

However the bounded-error assumption, while compatible with an infinity of probability distribution, doesn't take into acount the statistical properties of the perturbation on each measurements y_i. In [3], a method is proposed that allows to compute the set of all the parameters that are consistent with a given probability with a set of measurements, while taking into account the statistical properties of the perturbation.

We will study the widespread case where each measurement y_i is subject to a normaly distributed perturbation w_i, that is: $\widetilde{y}_i = y_i + w_i$.

In this case, the set $\mathbb{Y}$ is not a box anymore, but an n-dimensional ellipsoid which expression is given by:

$$\mathbb{Y} = \{\mathbf{y} | (\tilde{\mathbf{y}} - \mathbf{y})\mathbf{Q}^{-1}(\tilde{\mathbf{y}} - \mathbf{y})^T \leq \alpha(\eta)\}$$

with $\alpha(\eta)$ a confidence threshold depending on a given probability η of $\widetilde{\mathbf{y}}$ being in $\mathbb{Y}$, and $\mathbf{Q}$ the covariance matrix of the random vector $\mathbf{w}$.

The set-inversion problem now consists of inverting an ellipsoid, whose volume, as will be shown, is much smaller than the volume of its n-dimensional box counterpart.

As will be presented, this additional constraint will greatly enhance the precision of the estimation, in the sense that the set $\mathbb{S}$ will be much smaller.

The improvements will be presented on some test-cases.

References

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

[**2**] L. Jaulin and E. Walter, Guaranteed Nonlinear Parameter Estimation via Interval Computations, *Interval Computations* 3:61–75, 1993.

[**3**] L. Jaulin, Probabilistic set-membership estimation, *Second Workshop on Principles and Methods of Statistical Inference with Interval Probability*, 2009

Global Optimization based on Contractor Programming

Jordan Ninin[1]

[1] ENSTA-Bretagne, Lab-STICC,
2 rue Francois Verny, 29806 Brest, France
jordan.ninin@ensta-bretagne.fr

Keywords: Global Optimization, Contractor Programming

Introduction

In this talk, we will present a general pattern based on contractor programming for designing a global optimization solver. This approach allows to solve problems with a wide variety of constraints. The complexity and the performance of the algorithm rely on the construction of contractors which characterize the feasible region. We illustrate the methodology on a H_∞ control synthesis under structural constraints.

General pattern to designing global optimization solvers

Contractor Programming is a methodology which allows to enclose each algorithm in a unify framework, in order to interact heterogeneous formulations or techniques. This approach is based on Interval Analysis.Using Contractor Programming, we will show a user-friendly way to solve problems with non-smooth functions, disjunctive constraints, non-mathematical constraints (such as "stay in an area defined by a polygon") and constraints with quantifiers (such as ForAll and Exists). This approach allows to design, in a single step, a model and a solver for a given problem.

Given a physical problem, the user can construct a contractor for the feasible region $\mathbb{X}$ of his problem. We denote this contractor $\mathcal{C}_{out}$. Moreover, using the counterparts of set-membership operators for contractors, we can construct in the same way a contractor for the negation of $\mathbb{X}$. This contractor is denoted by $\mathcal{C}_{in}$. The only required mathematical expression is the objective function, f_{cost}.

Given a box $[\mathbf{x}] \in \mathbb{R}^n$, $\mathcal{C}_{out}([\mathbf{x}])$ removes from $[\mathbf{x}]$ a part that does not contain a feasible solution. In the same way, $\mathcal{C}_{in}([\mathbf{x}])$ removes from $[\mathbf{x}]$ parts which are entire feasible; i.e. $([\mathbf{x}]/\mathcal{C}_{in}([\mathbf{x}])) \subseteq \mathbb{X}$. Thus, $([\mathbf{x}]/\mathcal{C}_{in}([\mathbf{x}]))$ is a feasible subset and we can perform a global optimization without constraint on it. If this step succeeds, this set can be discarded: indeed, if a new best current solution is found, we save it and it is proved that this set does not contain a better solution; else it is directly proved that no better solution can be found in this set $([\mathbf{x}]/\mathcal{C}_{in}([\mathbf{x}]))$.

The following algorithm describes a simple implementation pattern for a global optimization solver based on contractors. This algorithm is inspired from the *SIVIA* Algorithm (Set-Inversion Via Interval Analysis), which is used to compute the feasible set in a domain.

The inputs are an initial domain $[\mathbf{x}] \in \mathbb{IR}^n$, $\mathcal{C}_{out}$ a contractor for $\mathbb{X}$, $\mathcal{C}_{in}$ a contractor for $\overline{\mathbb{X}}$ and f_{cost} an objective function. The outputs are $\tilde{f}$, the global minimum value found and $\tilde{x}$, a global minimum. A boolean variable b is added for each element of $\mathcal{L}$ to indicate if the element is included in the feasible region.

H_∞ control synthesis under structural constraints

We will illustrate this new approach on a example on the control of a periodic second order system G with a PID controller K subject to two frequency constraints on the error e and on the command u of the closed loop system. The objective is to find $k = (k_p, k_i, k_d)$ minimizing the H_∞ norm of the controlled system.

$$G(s) = \frac{k\omega_n}{s^2 + 2\xi\omega_n s + \omega_n^2}, \quad K(s) = k_p + \frac{k_i}{s} + k_d s.$$

$(\tilde{x},\ \tilde{f})$ = **OptimCtc** ($[\mathbf{x}]$, $\mathcal{C}_{out}$, $\mathcal{C}_{in}$, f_{cost}):

$\tilde{f} := +\infty$, denotes the current upper bound for the global minimum;
$\mathcal{L} := \{([\mathbf{x}], false)\}$, initialization of the data structure of the stored elements;
Let $\mathcal{C}_f$ a contractor based on the constraint $\{x \ :\ f_{cost}(x) \leq \tilde{f}\}$;
Repeat until a stopping criterion is fulfilled:
  Extract from $\mathcal{L}$ an element $([\mathbf{y}], b)$,
  Bisect the considered box $[\mathbf{y}]$: $[\mathbf{y}_1], [\mathbf{y}_2]$,
  for $j = 1$ to 2 :
    if $(b =$ false$)$ *then*
      Contract $[\mathbf{y}_i]$ with $\mathcal{C}_{out} \cap \mathcal{C}_f$,
      $[\mathbf{y}_{tmp}] := [\mathbf{y}_i]$,
      Contract $[\mathbf{y}_i]$ with $\mathcal{C}_{in}$,
      Add ($[\mathbf{y}_i]$,false) in $\mathcal{L}$.
      $[\mathbf{y}_{tmp}] := [\mathbf{y}_{tmp}]/[\mathbf{y}_i]$,
    else
      Contract $[\mathbf{y}_i]$ with $\mathcal{C}_f$,
      $[\mathbf{y}_{tmp}] := [\mathbf{y}_i]$,
    Try to **find** the global optimum without constraint in $[\mathbf{y}_{tmp}]$,
    if the search succeeds in a limited time *then*
      Update $\tilde{f}$ and $\tilde{x}$.
    else
      Add $([\mathbf{y}_{tmp}], true)$ in $\mathcal{L}$.
end.

Table 1: General pattern for a global optimization algorithm based on contractor programming.

The feasible region $\mathbb{K}_{in}$ of our global optimization problem have the following form, with Re_1 and Im_1 the real and imaginary part of the transfer function C_1 corresponding to the first constraint on the error, and Re_2 and Im_2 of C_2 corresponding to the second constraint on the

command. The objective function consists to minimizing γ.

$$\begin{aligned}\mathbb{K}_1 &= \{(k,\gamma) \,:\, \|C_1\,(G \star K)\|_\infty \leq \gamma\} \\ &= \{(k,\gamma) \,:\, \forall\omega, \sqrt{Re_1^2(k,\omega) + Im_1^2(k,\omega)} \leq \gamma\},\end{aligned}$$

$$\begin{aligned}\mathbb{K}_2 &= \{(k,\gamma) \,:\, \|C_2\,(G \star K)\|_\infty \leq \gamma\} \\ &= \{(k,\gamma) \,:\, \forall\omega, \sqrt{Re_2^2(k,\omega) + Im_2^2(k,\omega)} \leq \gamma\},\end{aligned}$$

$$\mathbb{K}_{in} = \mathbb{K}_1 \cap \mathbb{K}_2.$$

This algorithm is implemented in the library IBEX which is free available. The goal of this library is to give all tools to the users for designing easily the best solver for its own problem.

References

[**1**] IBEX : *a C++ numerical library based on interval arithmetic and constraint programming.*
http ://www.ibex-lib.org

[**2**] P. Apkarian and D. Noll, Nonsmooth H_∞ Synthesis", *IEEE Transaction on Automatic Control*, 51(1):71–86, 2006.

[**3**] G. Chabert and L. Jaulin, Contractor Programming, *Artificial Interlligence*, 173:1079–1100, 2009.

[**4**] L. Jaulin and E. Walter, Set inversion via interval analysis for nonlinear bounded-error estimation, *Automatica* 29(4):1053–1064, 1993.

[**5**] J. Ninin, F. Messine and P. Hansen, A reliable affine relaxation method for global optimization, *Quarterly Journal of Operations Research*, to appear, 2015.

Verified Convex Hull for Inexact Data

Yuki Ohta[1] and Katsuhisa Ozaki[2]

[1] Shibaura Institute of Technology, Graduate School of Engineering and Science,
307 Fukasaku, Minuma-ku, Saitama-shi, Saitama, Japan
nb15102@shibaura-it.ac.jp
[2] Shibaura Institute of Technology, Department of Mathematical Sciences,
307 Fukasaku, Minuma-ku, Saitama-shi, Saitama, Japan
ozaki@sic.shibaura-it.ac.jp

Keywords: convex hull, floating-point arithmetic, interval data

Introduction

This talk is concerned with a convex hull in two-dimensions. The convex hull is one of well-known topics in computational geometry. If floating-point arithmetic [1] is used for convex hull algorithm, then an inexact result may be obtained due to accumulation of rounding errors. This problem is called robustness problem and is introduced in detail in [2]. We focus on the convex hull for uncertain data, namely, data is given by set of intervals. We developed a verified algorithm for the convex hull for interval data.

Proposed Method

Three points $A = (a_x, a_y)$, $B = (b_x, b_y)$ and $C = (c_x, c_y)$ are given, where $|\tilde{a}_x - a_x| \leq r_{ax}, \ldots, |\tilde{c}_y - c_y| \leq r_{cy}$. Let a computed result D be defined as

$$D := \mathrm{fl}((\tilde{a}_x - \tilde{c}_x)(\tilde{b}_y - \tilde{c}_y) - (\tilde{a}_y - \tilde{c}_y)(\tilde{b}_x - \tilde{c}_x)),$$

where $\mathrm{fl}(\cdot)$ means that all operations inside the parentheses are evaluated by floating-point arithmetic. We developed floating-point filters

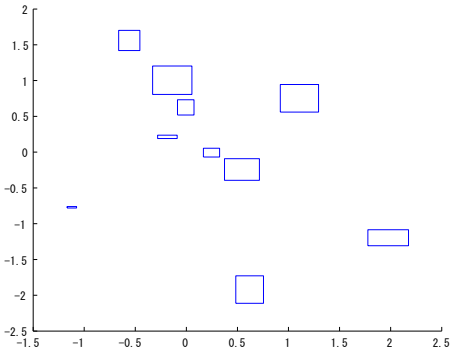


Figure 1: Input interval data

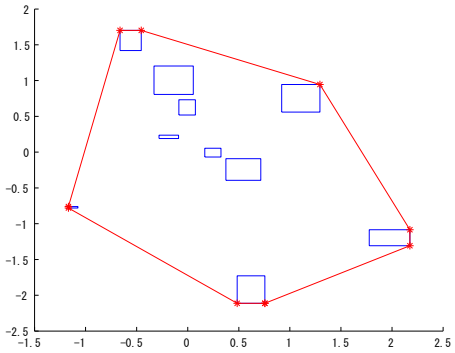


Figure 2: Outer convex hull

for a two-dimensional orientation problem with interval data. Our filters give a sufficient condition of the following

$$\mathrm{sign}(D) := \mathrm{sign}((a_x - c_x)(b_y - c_y) - (a_y - c_y)(b_x - c_x)), \quad \text{where } \forall a_x, \ldots, c_y.$$

Next, we developed an iterative convex hull algorithm based on the incremental algorithm for interval data. Our algorithm produces an outer convex hull [3] which encloses all intervals. Figure 1 shows input data and Figure 2 shows the outer convex hull for the input data. The details of the iterative algorithm with our floating-point filters and numerical results will be shown in the presentation.

References

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

[2] L. Kettner, K. Mehlhorn, S. Pion, S. Schirra, C. Yap, Classroom Examples of Robustness Problems in Geometric Computation, *Computational Geometry*, 40:61–78, 2007.

[3] T. Nagai, S. Yasutome and N. Tokura, Convex hull problem with imprecise input, *Discrete and Computational Geometry*, Springer, 1763:207–219, 2000.

Tight Enclosure of Matrix Multiplication with Level 3 BLAS

Katsuhisa Ozaki[1] and Takeshi Ogita[2]

[1] Shibaura Institute of Technology, Department of Mathematical Sciences, Fukasaku 307, 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

Keywords: Matrix Multiplication, Interval Arithmetic

Introduction

This talk is concerned with enclosure of a product of two matrices. Let $\mathbb{F}$ denote a set of floating-point numbers as defined in IEEE 754 [1]. For $A \in \mathbb{F}^{m\times n}$ and $B \in \mathbb{F}^{n\times p}$, the concern is to obtain an interval matrix $[C]$ which encloses AB, namely, $AB \in [C]$. Notations $\mathtt{fl}(\cdot)$, $\mathtt{fl}_{\triangle}(\cdot)$ and $\mathtt{fl}_{\triangledown}(\cdot)$ mean that all operations in the parenthesis are evaluated by floating-point arithmetic with rounding to nearest, rounding upward and rounding downward, respectively. A well-know method for this problem is to compute $[C] := [\mathtt{fl}_{\triangledown}(AB), \mathtt{fl}_{\triangle}(AB)]$, which involves two matrix products.

Proposed Method

Recently, we developed enclosure methods for AB via three or five floating-point matrix products, which often provide tighter results than the well-known method. First, A and B are split into an unevaluated sum of two matrices as follows

$$A = A^{(1)} + A^{(2)}, \quad B = B^{(1)} + B^{(2)}, \quad A^{(1)}B^{(1)} = \mathtt{fl}(A^{(1)}B^{(1)}) \qquad (1)$$

where $A^{(1)}, A^{(2)} \in \mathbb{F}^{m\times n}$, $B^{(1)}, B^{(2)} \in \mathbb{F}^{n\times p}$ and $\mathtt{fl}(A^{(1)}B^{(1)}) = A^{(1)}B^{(1)}$ means that no rounding error occurs in the evaluation of $A^{(1)}B^{(1)}$.

Then, AB is enclosed by

$$\begin{aligned} AB &= A^{(1)}B^{(1)} + A^{(1)}B^{(2)} + A^{(2)}B \\ &\in \mathtt{fl}(A^{(1)}B^{(1)}) + [\mathtt{fl}_{\triangledown}(A^{(1)}B^{(2)}), \mathtt{fl}_{\triangle}(A^{(1)}B^{(2)})] \\ &\quad + [\mathtt{fl}_{\triangledown}(A^{(2)}B), \mathtt{fl}_{\triangle}(A^{(2)}B)]. \end{aligned}$$

This method involves five matrix products.

Let a constant β be defined as

$$\beta := \lceil (\log_2 \alpha - \log_2 \mathtt{u})/2 \rceil \,. \tag{2}$$

We define two vectors σ and τ as follows:

$$\sigma_i := 2^{\beta} \cdot 2^{w_i}, \quad \tau_j := 2^{\beta} \cdot 2^{v_j},$$

where

$$w_i := \lceil \log_2 \max_{1 \le j \le n} |a_{ij}| \rceil \text{ for } \max_{1 \le j \le n} |a_{ij}| \neq 0 \;\;, \;\; w_i = 0 \text{ for } \max_{1 \le j \le n} |a_{ij}| = 0,$$
$$v_j := \lceil \log_2 \max_{1 \le i \le n} |b_{ij}| \rceil \text{ for } \max_{1 \le i \le n} |b_{ij}| \neq 0 \;\;, \;\; v_j = 0 \text{ for } \max_{1 \le i \le n} |b_{ij}| = 0.$$

If a suitable constant α in (2) can be set, then (1) is satisfied from the following results

$$a_{ij}^{(1)} := \mathtt{fl}((a_{ij} + \sigma_i) - \sigma_i), \qquad a_{ij}^{(2)} := \mathtt{fl}(a_{ij} - a_{ij}^{(1)}),$$
$$b_{ij}^{(1)} := \mathtt{fl}((b_{ij} + \tau_j) - \tau_j), \qquad b_{ij}^{(2)} := \mathtt{fl}(b_{ij} - b_{ij}^{(1)}).$$

In the corresponding original methods in [2] $\alpha := n$ yields (1). However, even if $\alpha < n$, we can prove (1) by a posteriori validation with diagonal scaling. The details of the method and numerical results will be shown in the presentation.

References

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

[**2**] K. Ozaki, T. Ogita and S. Oishi, Tight and efficient enclosure of matrix multiplication by using optimized BLAS, *Numerical Linear Algebra With Applications* 18(2):237–248, 2011.

Primitive Shape Characterization using Interval Methods

Salvador Pacheco and Alexandru Stancu

The University of Manchester,
School of Electrical and Electronic Engineering
Oxford Rd, Manchester, M13 9PL, UK.
{salvador.pachecogutierrez, alexandru.stancu}@manchester.ac.uk

Keywords: point cloud, parameter estimation, geometrical shapes.

Introduction

In the last years, the development of laser scanning technologies has lead to the increase of investigation in areas such as computer vision, 3D modeling, scene recognition, reverse engineering, etc.

This work is focused on the recognition of primitive shapes immerse in hazardous environments for decommissioning, more specifically in nuclear ambiances. This is, the characterization of basic geometrical shapes such as lines, circles, planes, spheres, and cylinders. The information is retrieved using the structured-light scanner Kinect, developed by Microsoft.

Approach

The detection of circles on images using interval methods has been investigated by Jaulin [1], as a problem of parameter estimation under error-bounded estimation basis. Therefore,

$$\mathbb{P} = \bigcap_{i \in \{1,\ldots,m\}} \underbrace{\{\mathbf{p} \in \mathbb{R}^{n_p}, \exists [\mathbf{y}] \in [\mathbf{y}]_i, \mathbf{f}(\mathbf{p}, \mathbf{y}) = 0\}}_{\mathbb{P}_i} \tag{1}$$

being $\mathbf{p}$ the parameter vector, $[\mathbf{y}]_i \subset \mathbb{R}^{n_y}$ is the ith measurement box and $\mathbf{f}$ is the model function. In this sense, the set $\mathbb{P}_i$ is the set of all

parameters vector consistent with the ith measurement box.

Jaulin defined the shape extraction as a set estimation problem [3]. The above can be expressed as follows,

$$\mathbf{f} : \begin{cases} \mathbb{R}^{n_p} \times \mathbb{R}^d \to \mathbb{R}^{n_f} \\ (\mathbf{p}, \mathbf{y}) \quad \to \mathbf{f}(\mathbf{p}, \mathbf{y}) \end{cases} \tag{2}$$

being $d \in \{2, 3\}$ the dimension of the analyzed shape. The vector $\mathbf{y} \in \mathbb{R}^d$ is the point cloud and $\mathbf{p}$ is the parameter vector that corresponds to the shape under analysis. Under this basis, the shape that corresponds to the vector $\mathbf{p}$ is defined as follows:

$$\mathcal{S}(\mathbf{p}) \stackrel{\text{def}}{=} \left\{ \mathbf{y} \in \mathbb{R}^d, \mathbf{f}(\mathbf{p}, \mathbf{y}) = 0 \right\} \tag{3}$$

Taking into consideration a set of boxes (measurements) in the primitive shape space dimension d, each of this boxes is assumed to touch the periphery of the considered shape. In this sense, the aim of this work is to extend the above concepts to a 3-dimensional space.

According to [3], the set of parameters for the description of each of the features of interest are the following:

Associated feature	**Parameters associated**		
	Location	Orientation	Size
Line (2D)	x_0, y_0	a, b	–
Circle (2D)	x_0, y_0	–	r
Plane	x_0, y_0, z_0	a, b, c	r
Sphere	x_0, y_0, z_0	–	r
Cylinder	x_0, y_0, z_0	a, b, c	r

These parameters are required to be accurate enough in order to have a good representation of the geometrical feature.

Results

Due to space restrictions, only the results of the plane and sphere are presented.

The shape of a sphere can be defined by the following expression:

$$\mathbf{f}(\mathbf{p}, \mathbf{y}) = (y_1 - x_0)^2 + (y_2 - y_0)^2 + (y_3 - z_0)^2 - r^2 \tag{4}$$

Using the proposed approach, the feature parameters are presented in Table 1.

Parameter	Value	Initial box	Parameters Detected
x_0	-10	$[-100, 100]$	$[-10.2597, -9.63753]$
y_0	-20	$[-100, 100]$	$[-20.3153, -19.657]$
z_0	30	$[-100, 100]$	$[29.6336, 30.3557]$
r	5	$[0, 100]$	$[4.77378, 5.28706]$

Table 1: Parameters of the sphere.

On the other hand, a plane can be defined by its general equation:

$$Ax + By + Cz + D = 0 \tag{5}$$

Fixing $B = -1$, the remaining feature parameters are presented in Table 2.

Parameter	Value	Initial box	Parameters Detected
A	2.94	$[-100, 100]$	$[2.71565, 3.20899]$
C	1.16	$[-100, 100]$	$[1.07501, 1.26956]$
D	-10.388	$[-100, 100]$	$[-11.423, -9.49593]$

Table 2: Parameters of the plane.

Moreover, a cylinder can be defined by a point lying on its central axis $\mathbf{X} = (x_0, y_0, z_0)$, the direction of such axis $\mathbf{n} = (n_x, n_y, n_z)$ and the radius r. Due to the nature of this representation, no closed form can be inferred. The procedure proposed to solve this problem lyes in transforming the 3-dimensional problem into a 2-dimensional one, by projecting all the points over a plane A in order to look for the plane

parameters on which the projection of the cylinder points is a circle. Further information regarding this approach will be discussed during the workshop.

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.

[**3**] European Committe for Standarization, Geometrical Product Specifications (GPS) – Acceptance and reverification tests for coordinate measuring machines (CMM), ISO 10360-6:2001 *British Standards*, May 2007.

Computing exact bounds for the solution set of parametric interval linear systems

Evgenija D. Popova

Bulgarian Academy of Sciences, Institut of Mathematics and Informatics
Acad. G. Bonchev str., bl. 8, 1113 Sofia, Bulgaria
epopova@bio.bas.bg

Keywords: interval linear system, dependent data, solution set, interval hull

Introduction

Consider linear algebraic systems involving linear dependencies between a number of interval parameters $p = (p_1, \ldots, p_k) \in (\mathbf{p}_1, \ldots, \mathbf{p}_k)$

$$\begin{aligned} & A(p)x \;=\; b(p) \\ & A(p) := A_0 + \sum_{i=1}^{k} p_i A_i, \qquad b(p) := b_0 + \sum_{i=1}^{k} p_i b_i, \end{aligned} \tag{1}$$

where $A_i \in \mathbb{R}^{n\times n}$, $b_i \in \mathbb{R}^n$, $i = 0, \ldots, k$. Performing worst-case analysis of uncertain systems one is interested in the parametric (united) solution set of the system (1)

$$\Sigma^p = \Sigma(A(p), b(p), \mathbf{p}) \;:=\; \{x \in \mathbb{R}^n \mid (\exists p \in \mathbf{p})(A(p)x = b(p))\}\,. \tag{2}$$

A parametric solution set (2), in general, has a complicated structure. It is nonconvex even in a single orthant. The boundary of Σ^p consists of parts of polynomials that may have arbitrary high degree. This causes difficulties in computing bounds of the solution set.

There are several methods for calculating lower and upper bounds for each component of the solution set, if the latter is bounded. Most of these methods require that the parametric matrix $A(p)$ be strongly regular on $\mathbf{p}$, which restricts the scope of their applicability. The obtained

bounds often overestimate the solution set considerably, especially for large parameter intervals.

We are interested in computing the exact interval hull $\square\Sigma^p$ of the solution set. For a bounded set Σ^p, $\square\Sigma^p := \bigcap\{u \in \mathbb{IR}^n \mid \Sigma^p \subseteq [u]\}$. This is an NP-hard problem even in the special case when Σ^p has linear boundary.

Proposed methodology

In [1] the boundary of a nonempty solution set Σ^p is described by parts of parametric hypersurfaces. The latter are defined by n parametric coordinate functions $x_i(q) := \{A^{-1}(q)b(q)\}_i$ depending on $n-1$ parameters $q \in \mathbb{R}^{n-1}$, $q \subseteq p$, $q \in \mathbf{q}$. Furthermore, following the same methodology, the projection of Σ^p on a 2-dimensional coordinate space can be represented by parts of parametric hypersurfaces depending on only one parameter.

Basing on the above representation of the boundary of a parametric solution set, computing the interval hull of $\{\Sigma^p\}_i$, $i = 1, \ldots, n$, is reduced to solving $\binom{k}{m}2^{k-m+1}$ constrained global optimization problems with corresponding objective functions $x_i(q)$ depending on m interval parameters, where m can be chosen to be any number $1 \leq m \leq n-1$. For $m = 1$, the corresponding constrained global optimization problems can be solved exactly in exact arithmetic and software tools which provide that are available.

We discuss various aspects (pros and cons) of the proposed methodology and its applicability along with some numerical examples.

References

[1] E. D. Popova and W. Krämer, Visualizing parametric solution sets, *BIT Numerical Mathematics* 48:95–115, 2008.

Computing Barriers of Ordinary Differential Equations

Stefan Ratschan

Institute of Computer Science, Czech Academy of Sciences,
stefan.ratschan@cs.cas.cz

Keywords: quantified constraints, safety verification

Consider the following problem: Given an ordinary differential equation, a set of initial states, and a set of states considered to be unsafe, compute a set that

- contains all initial states,
- cannot be left by any trajectory of the differential equation, and
- does not contain any unsafe state.

Such a set is often called a barrier. It certifies that the given ODE does not have a trajectory that starts in an initial state and ends in an unsafe state.

After using an inequality $p(x_1, \ldots, x_n) \leq 0$ for representing the barrier $\{x_1, \ldots, x_n \mid p(x_1, \ldots, x_n) \leq 0\}$, and assuming that that the set of initial states, and the set of unsafe states is also given as an inequality, the three conditions mentioned above can be formulated as universally quantified constraints [4]. Moreover, after introducing parameter values $a_1, \ldots, a_k$ into the inequality representing the barrier, arriving at $p(a_1, \ldots, a_k, x_1, \ldots, x_n) \leq 0$, the problem reduces to finding parameter values $a_1, \ldots, a_k$ such that three universally quantified constraints are fulfilled. Hence, we have a problem of solving quantified constraints [5, 3, 1] with quantifier prefix $\exists\forall$.

In our talk, we discuss a variant of the above problem where we want a quick but incomplete test for the existence of certain simple barriers which is useful as a part of more complicated safety verification

algorithms. We present an algorithm that uses interval techniques for handling the universal quantifier, but searches for the parameter values $a_1, \ldots, a_k$ using classical numerical techniques. We also show the results of first numerical experiments.

Acknowledgement

The research published in this paper was supported by GAČR grant 15-14484S and by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807).

References

[1] Olivier Bouissou, Alexandre Chapoutot, Adel Djaballah, and Michel Kieffer. Computation of parametric barrier functions for dynamical systems using interval analysis. In *IEEE CDC 2014*, pages 1–4, 2014.

[2] B. F. Caviness and J. R. Johnson, editors. *Quantifier Elimination and Cylindrical Algebraic Decomposition.* Springer, Wien, 1998.

[3] Stefan Ratschan. Efficient solving of quantified inequality constraints over the real numbers. *ACM Transactions on Computational Logic*, 7(4):723–748, 2006.

[4] Ankur Taly and Ashish Tiwari. Deductive verification of continuous dynamical systems. In Ravi Kannan and K Narayan Kumar, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2009)*, volume 4 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 383–394, Dagstuhl, Germany, 2009. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[5] A. Tarski. *A Decision Method for Elementary Algebra and Geometry.* Univ. of California Press, Berkeley, 1951. Also in [2].

The (near-)future IEEE 1788 standard for interval arithmetic

Nathalie Revol

INRIA, AriC team
LIP, ENS de Lyon
Université de Lyon
46 allée d'Italie, 69364 Lyon Cedex 07, France
Nathalie.Revol@ens-lyon.fr

Keywords: interval arithmetic, standardization, Moore's mathematical model, set-based mathematical model

Introduction

The standardization effort started in January 2008, at a meeting in Dagstuhl. The participants felt that interval arithmetic was mature enough to undergo the standardization process, and that this process was needed, for instance to make existing libraries interoperable. Since October 2008, where the project was launched, an intensive and sustained work has been done, through electronic means. All related work can be found on the Web page of the project [1]. The mailing list had more than 140 subscribers from 28 countries. More than 40 persons have been active in the discussions and votes. This led to a final draft which has been unanimously adopted by the working group in July 2014, and after one more year of editorial work and final ballot the standard should go through the final examination stage in June 2015.

The standard is structured into 4 levels, where level 1 is the mathematical level and level 4 is about encoding at the bit level. Level 2, *discretization*, is the central part of the standard, approximating the mathematical theory by an implementation-defined finite set of entities and operations. Level 3 is about representation issues.

Common intervals

The mathematical model, which serves as a basis for this standard, is called *common interval arithmetic*. It corresponds to the model proposed by Moore in [2]: only bounded and non-empty intervals are considered. We will detail the various operations available on these intervals: arithmetic operations, set operations, numeric and boolean functions of intervals, as well as operations on/with decorations.

Other flavors

Other mathematical models have been considered: set-based model, Kaucher/modal model in particular. These models are called *flavors*. The standard has been designed to allow a smooth integration of other models, by providing "hooks" to new models and by defining the process for submitting a new flavor.

The set-based model has been completely defined and it is, up to now, the only flavor defined in the standard. In the set-based model, the empty set and unbounded intervals are allowed. Its definition makes the second part of the draft.

We will also briefly describe some possible flavors, namely the Kaucher/modal flavor and an approach proposed by S. Rump to handle overflows and open or closed or half-open intervals.

Acknowledgement (non inclusive list)

Thank you to R. Baker Kearfott and John D. Pryce for the tremendous amount of work they put in this project.

References

[**1**] IEEE, *P1788/D9.7 2 – Draft Standard For Interval Arithmetic*, 2015.

[**2**] R.E. Moore, *Interval analysis*, Prentice-Hall, 1966.

Goal programming approach for solving interval MOLP problems

S. Rivaz[1], M. A. Yaghoobi[2] and M. Hladík[3]

[1] Department of Mathematics, Yasouj University, Yasouj, Iran
sanazrivaz@gmail.com
[2] Department of Applied Mathematics,
Shahid Bahonar University of Kerman, Kerman, Iran
yaghoobi@uk.ac.ir
3 Department of Applied Mathematics,
Charles University, Malostranské nám. 25, 11800 Prague, Czech Republic
hladik@kam.mff.cuni.cz

Keywords: Multiobjective linear programming, interval programming, goal programming

Introduction

Uncertainty of the model parameters are frequently involved in most of the decision-making problems due to inherent imprecise in nature of human judgments.
To deal with such imprecisions, various ways are existed. One of them is interval based approach in which the ambiguity of parameters are modeled by intervals. In this article, multiobjective linear programming problems with interval objective functions coefficients, or interval MOLP problems for short, are investigated. Such problems have gained many researchers' interests in the last three decades; an overview on interval MOLP was given by Oliveira et al. [4].
It is well known that goal programming method is one of the most popular and powerful methods in MOLP [2]. In the context of goal programming, Inuiguchi and Kume have derived four formulations for an interval MOLP problem with target intervals [1]. It should be noted that since the decision-maker is not always an expert, he/she cannot decide which formulation is the most appropriate one for his/her situation.

In this paper, we use a specified distance concept between intervals to introduce a unique model for an interval MOLP problem with target intervals, via goal programming approach. This model is derived from a simple mathematical procedure and because of its uniqueness, the decision-maker has no doubt about its appropriateness.

Problem statement and the proposed method

We consider the following interval MOLP problem with interval targets:
Optimize:

$$\sum_{j=1}^{n} c_{kj}x_j = t_k, \quad k = 1, \cdots, p, \tag{1}$$

$$\begin{aligned} s.t. \quad & Ax \leq b, \\ & c_{kj} \in C_{kj}, \quad k = 1, \cdots, p, \quad j = 1, \cdots, n, \\ & t_k \in T_k, \quad k = 1, \cdots, p, \\ & x \geq 0, \end{aligned}$$

where C_{kj} is the closed interval $[c_{kj}^l, c_{kj}^u]$ representing a region the coefficients c_{kj} possibly take. T_k is the closed interval $[t_k^l, t_k^u]$ representing a region the target value t_k possibly take. The constraints of the problem are as in conventional MOLP problem. our aim, for solving problem (1), is to bring each planned interval $[\sum_{j=1}^n c_{kj}^l x_j, \sum_{j=1}^n c_{kj}^u x_j]$ as close as possible to target interval $[t_k^l, t_k^u]$, $k = 1, \cdots, p$. It means that minimizing the total distance of planned intervals from target intervals is needed. In order to do that, we use the following distance concept between two intervals.

Definition 1. *([3]) Let $[a^l, a^u]$ and $[b^l, b^u]$ be two intervals. Then, the distance between them is defined as:*

$$d([a^l, a^u], [b^l, b^u]) = max\{|a^l - b^l|, |a^u - b^u|\}.$$

It is easy to check that the distance d, defined in above introduces a metric on the space of closed intervals in $\mathbb{R}$. Based on this concept, we have:

$$D_k = d([\sum_{j=1}^{n} c_{kj}^l x_j, \sum_{j=1}^{n} c_{kj}^u x_j], [t_k^l, t_k^u]) =$$
$$max(|\sum_{j=1}^{n} c_{kj}^l x_j - t_k^l|, |\sum_{j=1}^{n} c_{kj}^u x_j - t_k^u|), k = 1, \cdots, p,$$

as the distance between the kth planned interval and its target interval ($k = 1, \cdots, p$). Using the deviational variables $d_k^{l^-}$, $d_k^{l^+}$, $d_k^{u^-}$ and $d_k^{u^+}$ as:

$$\sum_{j=1}^{n} c_{kj}^l x_j + d_k^{l^-} - d_k^{l^+} = t_k^l; \quad \sum_{j=1}^{n} c_{kj}^u x_j + d_k^{u^-} - d_k^{u^+} = t_k^u;$$

D_k can be written as follows:

$$D_k = max(|d_k^{l^-} - d_k^{l^+}|, |d_k^{u^-} - d_k^{u^+}|), k = 1, \cdots, p.$$

For aggregating all distances, $D(x)$ is considered:

$$D(x) = \lambda \sum_{k=1}^{p} w_k D_k + (1 - \lambda) \bigvee_{k=1}^{p} D_k,$$

where $0 \leq \lambda \leq 1$, $w_k \geq 0$, $k = 1, \cdots, p$, and $\sum_{k=1}^{p} w_k = 1$. Now, our model, for solving an interval MOLP problem with target intervals, can be stated in the following way:

$$min \quad \lambda \sum_{k=1}^{p} w_k v_k + (1 - \lambda) u \tag{2}$$

$$
\begin{aligned}
s.t. \quad \sum_{j=1}^{n} c_{kj}^{l} x_j + d_k^{l^-} - d_k^{l^+} &= t_k^l, \quad k = 1, \cdots, p, \\
\sum_{j=1}^{n} c_{kj}^{u} x_j + d_k^{u^-} - d_k^{u^+} &= t_k^u, \quad k = 1, \cdots, p, \\
Ax &\leq b, \\
d_k^{l^-} + d_k^{l^+} &\leq v_k, \quad k = 1, \cdots, p, \\
d_k^{u^-} + d_k^{u^+} &\leq v_k, \quad k = 1, \cdots, p, \\
v_k &\leq u, \quad k = 1, \cdots, p, \\
x_j &\geq 0, \quad j = 1, \cdots, n, \\
v_k &\geq 0, \quad k = 1, \cdots, p, \\
d_k^{l^-},\ d_k^{l^+},\ d_k^{u^-},\ d_k^{u^+} &\geq 0, \quad k = 1, \cdots, p, \\
u &\geq 0.
\end{aligned}
$$

Lemma 1. *Let model (2) be feasible then it has an optimal solution in which the constraints* $d_k^{l^-}.d_k^{l^+} = 0$ *and* $d_k^{u^-}.d_k^{u^+} = 0$, $k = 1, \cdots, p$, *are satisfied.*

References

[1] M. Inuiguchi and Y. Kume, Goal programming problems with interval coefficients and target intervals, *European Journal of Operational Research* 52: 345-360, 1991.

[2] D.F. Jones and M. Tamiz, *Practical Goal Programming*, Springer, 2010.

[3] R.E. Moore, R.B. Kearfott and M.J. Cloud, *Introduction to Interval Analysis*, Society for Industrial and Applied Mathematics, Philadelphia, 2009.

[4] C. Oliveira and C.H. Antunes, Multiple objective linear programming models with interval coefficients - an illustrated overview, *European Journal of Operational Research* 181: 1434-1463, 2007.

Robot localization in an unknown but symmetric environment

Simon Rohou[1], Luc Jaulin[1], Lyudmila Mihaylova[2],
Fabrice Le Bars[1], Sandor M. Veres[2]

[1] ENSTA Bretagne, Lab-STICC, rue François Verny 29200 Brest, France
[2] University of Sheffield, Western Bank Sheffield S10 2TN, UK
simon.rohou@ensta-bretagne.org

Keywords: interval analysis, robotics, state estimation, inter-temporality, unknown environment, symmetry, tubes, t-plane

Introduction

We consider the dynamic localization of a robot moving inside an environment which is unknown. We assume the environment has some symmetry properties (*e.g.*, cylindrical, axial, etc.). This type of localization can be met in an underwater context where the celerity of the sound is unknown but depth-dependent. In such a context, a classical state estimation approach such as [Gni10] cannot be applied without some modifications based on the formalization presented in this paper.

We assume that the robot is described by:

$$\begin{cases} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ y(t) &= h \circ g(\mathbf{x}(t)) \end{cases}$$

Where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the input vector, $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is the *evolution* function, $g : \mathbb{R}^n \to \mathbb{R}$ the *observation* function and $y \in \mathbb{R}$ is a measurement which is assumed to be scalar. We call $h : \mathbb{R} \to \mathbb{R}$ the *distortion* function which pictures the uncertainties of the environment.

Here, h is considered unknown but strictly increasing and we will assume that $\mathbf{u}(t)$ and $y(t)$ are not known precisely. Instead, we have two tubes $[\mathbf{u}](t)$ and $[y](t)$ containing $\mathbf{u}(t)$ and $y(t)$.

The approach to be considered here extends the double weighing principle of Borda. More precisely, we will introduce some intertemporal relations in order to cancel the unknown effects of the environment. A contractor programming method [Cha09] will then be applied to enclose the trajectory of the robot.

Main approach

To solve this problem, we define the function:

$$\varphi(t_1, t_2) = h \circ g(\mathbf{x}(t_2)) - h \circ g(\mathbf{x}(t_1)).$$

Since h is injective, we have: $\varphi(t_1, t_2) = 0 \;\Rightarrow\; g(\mathbf{x}(t_1)) = g(\mathbf{x}(t_2))$. As a consequence if $\varphi(t_1, t_2) = 0$, the two states $\mathbf{x}(t_1)$ and $\mathbf{x}(t_2)$ are symmetrical with respect to the environment. A known relation between these two states (here $g(\mathbf{x}(t_1)) = g(\mathbf{x}(t_2))$) can be derived. This relation can then be used for state estimation. The main problem we now have is to find these t-pairs (t_1, t_2) [Aub13] such that $\varphi(t_1, t_2) = 0$.

Proposition 1. If we define:

$$[\varphi](t_1, t_2) = [y](t_2) - [y](t_1) = [y^-(t_2) - y^+(t_1), y^+(t_2) - y^-(t_1)]$$

then we have $\varphi(t_1, t_2) \in [\varphi](t_1, t_2)$.

Proof. Since for all t, $h \circ g(\mathbf{x}(t)) = y(t)$ and since $y(t) \in [y](t)$, we have $\varphi(t_1, t_2) \in [\varphi](t_1, t_2) = [y](t_2) - [y](t_1)$.

Definition. We define the *presymmetric* set $\mathbb{S}$ as:

$$\mathbb{S} = \{(t_1, t_2) \mid g(\mathbf{x}(t_2)) - g(\mathbf{x}(t_1)) \leq 0\}$$

Since h is increasing, we have:

$$\mathbb{S} = \{(t_1, t_2) \mid \varphi(t_1, t_2) \leq 0\} = \varphi^{-1}(\mathbb{R}^-)$$

Now, φ is not known exactly, *i.e.*, we only have $\varphi \in [\varphi] = [\varphi^-, \varphi^+]$. As a consequence, we have:

$$\underbrace{(\varphi^+)^{-1}(\mathbb{R}^-)}_{\mathbb{S}^-} \subset \mathbb{S} \subset \underbrace{(\varphi^-)^{-1}(\mathbb{R}^-)}_{\mathbb{S}^+}$$

We can reformulate the problem into the following *Constraint Network*:

$$\begin{cases} \textbf{Variables: } \mathbf{x}, \mathbb{S} \\ \textbf{Constraints:} \\ \quad \text{(a) } \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \quad \text{(b) } \mathbb{S} = \{(t_1, t_2) \mid g(\mathbf{x}(t_2)) - g(\mathbf{x}(t_1)) \leq 0\} \\ \quad \text{(c) } (\varphi^+)^{-1}(\mathbb{R}^-) \subset \mathbb{S} \subset (\varphi^-)^{-1}(\mathbb{R}^-) \\ \qquad \text{where } \varphi^-(t_1, t_2) = y^-(t_2) - y^+(t_1) \text{ and } \varphi^+(t_1, t_2) = y^+(t_2) - y^-(t_1) \\ \textbf{Domains: } [\mathbf{x}], [\mathbb{S}] \\ \textbf{Initialization}: \ \mathbf{x} \in [-\infty, +\infty]^2, \ [\mathbb{S}] = [\emptyset, [0, t_{max}]^2]. \end{cases}$$

In this network, the unknow function h does not appear anymore. This is due to the fact that inter-temporal constraints compensate the unknown influence of h.

Test case

We consider a boat moving along a line. A static beacon stands on the seabed just below the origin of the frame. We do know neither the depth of the beacon nor the celerity of the sound. We assume that the environment is symmetric with respect to any horizontal translation. The robot is able to measure signals time of flight between its position and the beacon. The state vector of the system is $\mathbf{x} = \{s, \dot{s}\}^\intercal$ where s is the position of the robot. At first, the robot is completely lost ($s \in [-\infty, +\infty]$). Whith relevant inter-temporal measurements, the robot is able to contract $[\mathbf{x}]$ in *forward-backward*. This will be done computing an approximation of the set $\mathbb{S}$ as illustrated by Fig 1.

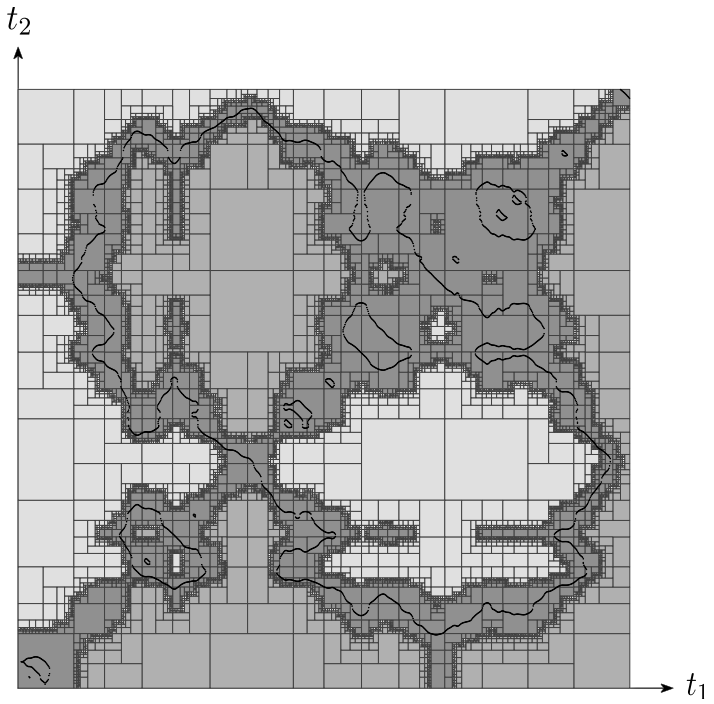


Figure 1: a t-plane obtained with a SIVIA algorithm. Middle-gray boxes are inside the *presymmetric* set $\mathbb{S}$ whereas light-gray ones are outside. We can guarantee true symmetries (pictured with black lines) belong to dark-gray areas.

Acknowledgement

Thanks the *Direction Générale de l'Armement* (FR) for its support.

References

[**Cha09**] G. CHABERT AND L. JAULIN, Contractor programming, *Artificial Intelligence*, 2009.

[**Aub13**] C. AUBRY, R. DESMARE AND L. JAULIN, Loop detection of mobile robots using interval analysis, *Automatica*, 2013.

[**Gni10**] A. GNING AND L. MIHAYLOVA, Mixture of uniform probability density functions for non linear state estimation using interval analysis, *Fusion*, 2010.

Validated Simulation of Differential Algebraic Equations*

Julien Alexandre dit Sandretto, Alexandre Chapoutot

U2IS, ENSTA ParisTech
828, boulevard des maréchaux, 91000, Palaiseau, France
{alexandre,chapoutot}@ensta.fr

Keywords: differential algebraic equations, guaranteed integration.

Introduction

Our recent results on validated simulation of ordinary differential equations (ODE) with implicit Runge-Kutta schemes [1] lead us to go up in complexity of kind of differential equations. Indeed, we are able to simulate ODE with interval parameters which is one of the requirement for our solver of differential algebraic equation (DAE). We currently focus on the DAE in Hessenberg index 1 form, that is

$$\dot{\mathbf{y}} = f(t, \mathbf{x}, \mathbf{y}), \tag{1a}$$
$$0 = g(t, \mathbf{x}, \mathbf{y}) \ . \tag{1b}$$

In Equation (1) $\mathbf{y}$ is the state variable and $\mathbf{x}$ is the algebraic variable (without an expression for its derivative) and $\dot{\mathbf{y}}$ stands for the time derivative of $\mathbf{y}$. This kind of *DAE* is common and used by a majority of simulation tools as Simulink and Modelica-like software.

A simulation procedure for ODE consists in two phases repeated at each simulation step k, starting from a given guaranteed initial value $[\mathbf{y}_k]$ at time instant t_k then compute an enclosure $[\mathbf{y}_{k+1}] \ni \mathbf{y}(t_{k+1})$ in function of $[\mathbf{y}_k] \ni \mathbf{y}(t_k)$ with the step-size $h_k = t_{k+1} - t_k$.

*This research benefited from the support of the "Chair Complex Systems Engineering - Ecole Polytechnique, THALES, DGA, FX, DASSAULT AVIATION, DCNS Research, ENSTA ParisTech, Télécom ParisTech, Fondation ParisTech and FDO ENSTA"

The major issue in the validated integration of DAE is the consistency of the initial values [4]. The additional constraints, Equation (1b), to be satisfied by the differential and the algebraic values are generally obtained by the Pantelides algorithm [3]. One attempt was made in order to solve DAE with guarantee by using an approximation of the solution and by adding a "post" consistency verification [2]. This approach produced mitigated results, and we propose a new approach in the following.

Main idea

Our method is based on the ability of the interval representation to enclose a set of solutions and use two contractors to reduce this enclosure around the solution.

An enclosure

Firstly, we do need the guaranteed enclosures of the solution of the differential equations (Equation (1a)), denoted by $[\tilde{\mathbf{y}}_k]$, and of the solution of the algebraic constraints (Equation (1b)), denoted by $[\tilde{\mathbf{x}}_k]$, at each step k of integration process. These enclosures are obtained with a novel operator mixing a classical Picard-Lindelöf operator with $[\tilde{\mathbf{x}}_k]$ as a $\forall$-parameter and a parametric Krawczyk operator with $[\tilde{\mathbf{y}}_k]$ as a $\forall$-parameter. The goal is then to find a post fixpoint simultaneously satisfying $[\tilde{\mathbf{y}}_k]$ and $[\tilde{\mathbf{x}}_k]$. This operator prove the existence and the unicity of the solution for the dynamical part for all values of the algebraic variable and by the way the fulfillment of the constraints, whatever the state variable.

Two contractors

After obtaining these enclosures, we have to reduce $[\tilde{\mathbf{y}}_k]$ around the solution $\mathbf{y}(t_{k+1})$. It is done with the help of our powerful validated implicit Runge-Kutta schemes [1]. Essentially, we used a validated Radau quadrature IIA, known for its efficiency and stability on DAE.

This scheme is able to manage with an interval parameters, such as $[\tilde{\mathbf{x}}_k]$. After that, the second contractor is used to reduce $[\tilde{\mathbf{x}}_k]$ around the solution $\mathbf{x}(t_{k+1})$. We combine for this purpose the Krawczyk used in the first step and a forward/backward contractor.

Main results

We solve the pendulum problem in index 1 form whose dynamics is given by

$$f: \quad \begin{cases} \dot{p} = u \\ \dot{q} = v \\ m\dot{u} = -p\lambda \\ m\dot{v} = -q\lambda - g \end{cases}$$

associated with constraint

$$g: \quad 0 = m(u^2 + v^2) - gq - \ell^2\lambda \ .$$

In functions f and g, m is the mass of the pendulum, ℓ is the length of the rod, u and v are Cartesian coordinates of the mass while p and q stand for the angular speed, g is the gravity force and λ stands for the Lagrange multiplier.

The simulation time is set to 1.6 seconds and with the initial conditions given by $p(0) = 1, q(0) = 0, u(0) = 0, v(0) = 0$, and $\lambda(0) \in [-0.01, 0.01]$. The consistency is verified with Krawczyk which gives $\lambda(0) \in [-0, 0]$. The trajectory computes by our method is given in Figure 1. This simulation takes about ten minutes with a maximal diameter of 0.02 for the final solution.

Conclusion

We presented in this abstract the first solid approach for the validated integration of the DAE under the Hessenberg-index 1 form. The first results are already interesting even if many issues have been opened to obtain an efficient tool.

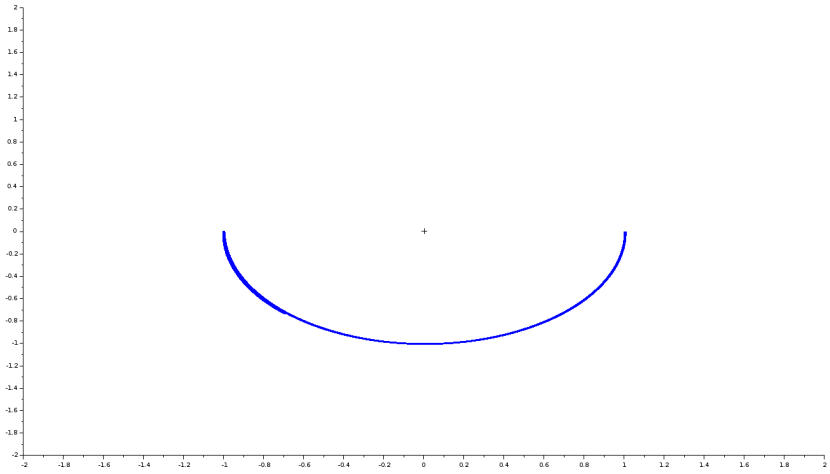


Figure 1: Trajectory of the pendulum.

References

[1] Alexandre dit Sandretto, J. and Chapoutot, A., Validated Solution of Initial Value Problem for Ordinary Differential Equations based on Explicit and Implicit Runge-Kutta Schemes, *Research Report, ENSTA ParisTech*, hal-01107685, 2015.

[2] Rauh, Andreas, Michael Brill, and Clemens Günther, A novel interval arithmetic approach for solving differential-algebraic equations with ValEncIA-IVP, *International Journal of Applied Mathematics and Computer Science*, 19.3: 381-397, 2009.

[3] Pantelides, Constantinos C, The consistent initialization of differential-algebraic systems, *SIAM Journal on Scientific and Statistical Computing*, 9.2: 213-231, 1988.

[4] Vieira, R. C., and E. C. Biscaia Jr, An overview of initialization approaches for differential algebraic equations, *Latin American Applied Research*, 30.4: 303-313, 2000.

Cooperative Localization And Formation Maintaining Using Range-only Measurements Without Communications

Mohamed Saad Ibn Seddik, Luc Jaulin and
Jonathan Grimsdale

Abstract

This paper is about using range-only measurements between multiple Autonomous Underwater Vehicles (AUV) to maintain a predefined formation. The AUVs have no a priori knowledge of each other's path or decisions. All vehicles must maintain the same speed during the mission and are not allowed to stop. Each vehicle must then adjust its position using the available range-only data. We provide a guaranteed state estimation of the targeted vehicle using interval analysis and a set-membership approach. Simulation results are described and discussed.

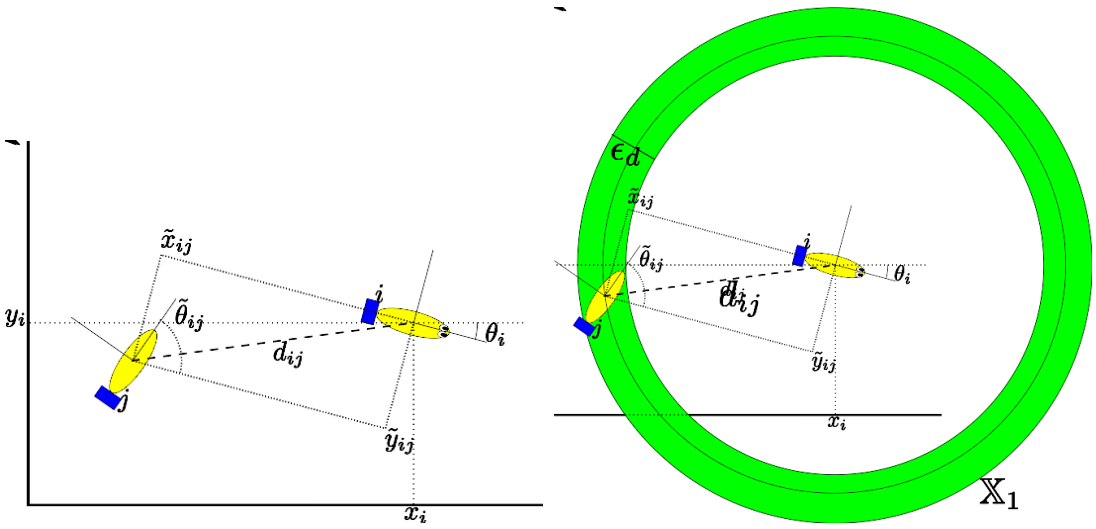
Introduction And Background

Envision a scenario where a swarm of Autonomous Underwater Vehicles (AUV) is exploring an area. Due to the large number of vehicles and because of the communication medium (underwater acoustics), the vehicles can hardly exchange their position with their neighbors to maintain a given formation. In this paper, we propose to use range-only measurements to keep the formation. Using synchronized clocks, all vehicles emit a unique ping at a given time. The one-way acoustic time of flight is used to determine the range between vehicles. One-Way time of flight has already been explored using stationary objects

[3] and a similar study has been done with communicating moving AUVs [1, 7].

As all the vehicles are similar and evolve at the same depth, the state of each vehicle i will be represented with its position in a 2-D plane and its heading, $\mathbf{x}_i = [x_i, y_i, \theta_i]^T$. For the mission purposes, the vehicles must keep a given speed v and can only control their rotation speed $u_i \in U$, where U is the set of all possible rotation speeds. For the sake of simplicity, let the robot motion be described by the following state equations

$$\begin{cases} \dot{x}_i &= v_i \cdot \cos\theta_i \\ \dot{y}_i &= v_i \cdot \sin\theta_i \\ \dot{\theta}_i &= u_i \end{cases} \tag{1}$$



(a) Representation of two vehicles of the swarm.

(b) With range-only information at the first step, $\mathbb{X}_1$ is a cylinder (projected here on the 2-D space).

Figure 1: Relative Localization

Due to the lack of information about the vehicles' absolute position, we propose to study the relative position and therefore a relative localization. Let f be the evolution function of the AUV j's relative position $\tilde{\mathbf{x}}_{ij}$ in the vehicle i frame, figure 1a. The ranging information

between the two vehicles will be represented as

$$\mathbf{y}_{ij} = d_{ij} + \epsilon_d = \sqrt{\tilde{x}_{ij}^2 + \tilde{y}_{ij}^2} + \epsilon_d = g(\tilde{\mathbf{x}}_{ij}) \tag{2}$$

The system can then be represented in the frequently used from,

$$\Sigma : \begin{cases} \dot{\tilde{\mathbf{x}}}_{ij} & = f(\tilde{\mathbf{x}}_{ij}, u_j, u_i) \\ \mathbf{y}_{ij} & = g(\tilde{\mathbf{x}}_{ij}) \end{cases} \tag{3}$$

To solve the system Σ of equation (3), we propose to use set-membership techniques and interval analysis [5].

Set-Membership Estimator

Measurements of distance come at discrete instants. Let us consider the system in equation (3) at a discrete time domain using Euler's discretization:

$$\Sigma : \begin{cases} \tilde{\mathbf{x}}_{ij}^k & = \tilde{f}(\tilde{\mathbf{x}}_{ij}^{k-1}, u_j^{k-1}, u_i^{k-1}) \\ \mathbf{y}_{ij}^k & = g(\tilde{\mathbf{x}}_{ij}^k) \end{cases}, k = 0, \ldots, t_{final} \tag{4}$$

where $\tilde{\mathbf{x}}_{ij}^k \in \mathbb{IR}^3$ is the state vector at the discrete time k , $\mathbf{y}_{ij}^k \in \mathbb{IR}$ is the output and $\tilde{f}$ is the Euler integral of f.

Let $\mathbb{X}_k$ be the associated domain set of the variable $\tilde{\mathbf{x}}^{k1}$ at the instant k. As no prior information is available on $\tilde{\mathbf{x}}^0, \ldots, \tilde{\mathbf{x}}^{t_{final}}$, thus $\mathbb{X}_0, \ldots, \mathbb{X}_{t_{final}}$ are taken as $\mathbb{IR}^3$. The measurement $\mathbf{y}^k$, with the noise value ϵ_d, is used to form the measurement set $\mathbb{Y}_k$. u_i is known as the robot's own input, and we define $\mathbb{U}_k$ is the set of all possible u_j^k. The equation (4) then becomes

$$\begin{cases} \mathbb{X}_k & = \tilde{f}(\mathbb{X}_{k-1}, \mathbb{U}_{k-1}, u_i^{k-1}) \\ \mathbb{Y}_k & = g(\mathbb{X}_k) \end{cases} \tag{5}$$

[1]For simplification purposes, the indexes ij will be omitted.

At every instant k, the state set $\mathbb{X}_k$ is defined with equation (5), thus it can be computed with

$$\mathbb{X}_k = \tilde{f}(\mathbb{X}_{k-1}, \mathbb{U}_{k-1}, u_i^{k-1}) \cap g^{-1}(\mathbb{Y}_k),\ k = 1, \ldots, t_{final} \tag{6}$$

Let's take as an example the first two set $\mathbb{X}_0$ and $\mathbb{X}_1$. As no prior information is available, $\mathbb{X}_0 = \mathbb{X}_1 = \mathbb{IR}^3$. Then applying equation (6) to $\mathbb{X}_1$ will contract the space to a cylinder in $\mathbb{IR}^3$, see figure 1b.

Reversibly, $\mathbb{X}_{k-1}$ can be computed from the recently computed $\mathbb{X}_k$ using the inverse of $\tilde{f}$,

$$\mathbb{X}_{k-1} = \tilde{f}^{-1}(\mathbb{X}_k, \mathbb{U}_{k-1}, u_i^{k-1}),\ k = 1, \ldots, t_{final} \tag{7}$$

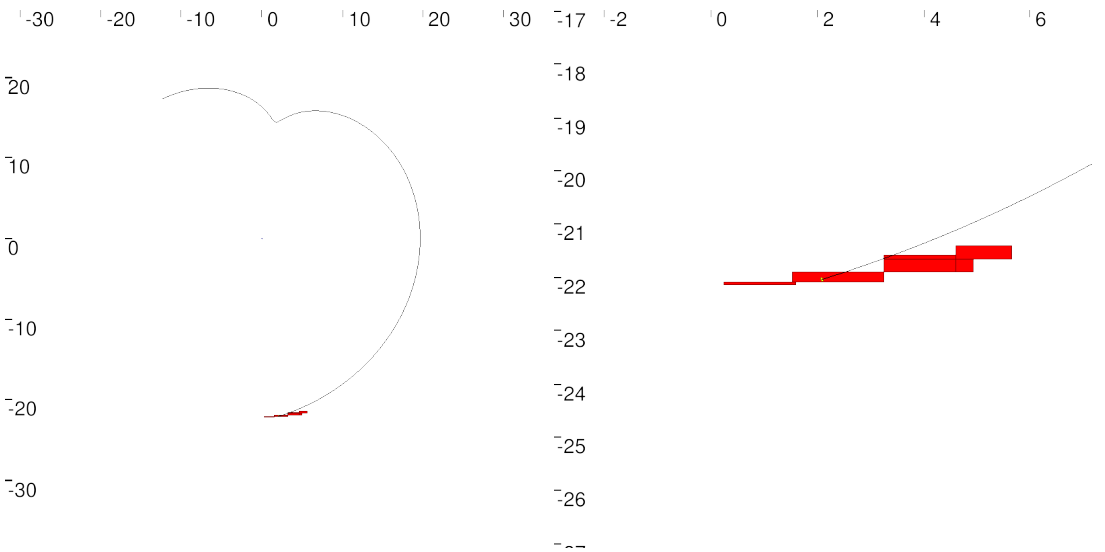
Applying both equation (6) and equation (7) to every set $\mathbb{X}_k$ is similar to applying a non-causal state estimator [4].

Preliminary Results

A scenario of two vehicles is simulated, where, first, a vehicle i loiters around a position and the vehicle j moves in a straight line.

Applying the algorithm on the data set provided by the simulation, figure 2, shows that it provides a relatively accurate position estimation of the targeted vehicle. In this figure, the red boxes represent a projection of the set $\mathbb{X}_k$ of the (xOy) plane. One will notice that the set $\mathbb{X}_k$ is made of multiple boxes, this is due to the bisection algorithm [5, 6].

As the vehicle i has a relatively accurate position of the position of the vehicle j, it can now maintain the formation by going to a desired position knowing the set $\mathbb{X}_k$ where the other vehicle is. A strategy based behavior [2] can be implemented to keep the formation as this algorithm is symmetrical. For example, when j realizes that it is too far from i and knows that the latter will not be able to maintain the formation because of the speed limitation, it can loiter waiting for i to catch up.



(a) The black line represents the trajectory of j in i's frame.

(b) Zooming around the real position (represented with the yellow ellipse) shows that it is closely surrounded by the computed set $\mathbb{X}_{t_{final}}$.

Figure 2: Set-Membership Inversion results.

Conclusion And Future Work

In this paper, a relative positioning based only on range information has been developed. Even though the system is non-linear and non observable, we showed that the set-membership approach can produce a relatively accurate relative position estimation based only on ranging measurements. The resulting position estimation can be enough to maintain a given formation between the vehicles. Future work will consist of simulating a swarm of AUVs all running the same algorithm. A real-life experiment will also conducted to provide realistic data to be studied as the modeling used for simulation is not as accurate.

References

[1] P Baccou, B Jouvencel, V Creuze, and C Rabaud. Cooperative positioning and navigation for multiple auv operations. In

OCEANS, 2001. MTS/IEEE Conference and Exhibition, volume 3, pages 1816–1821. IEEE, 2001.

[2] Michael R Benjamin, Joseph A Curcio, John J Leonard, and Paul M Newman. Navigation of unmanned marine vehicles in accordance with the rules of the road. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3581–3587. IEEE, 2006.

[3] Ryan M Eustice, Louis L Whitcomb, Hanumant Singh, and Matthew Grund. Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4257–4264. IEEE, 2007.

[4] L. Jaulin, M. Kieffer, I. Braems, and E. Walter. Guaranteed nonlinear estimation using constraint propagation on sets. *International Journal of Control*, 74(18):1772–1782, 2001.

[5] L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993.

[6] Mohamed Saad Ibn Seddik, Luc Jaulin, and Jonathan Grimsdale. Phase based localization for underwater vehicles using interval analysis. *Mathematics in Computer Science*, 8(3-4):495–502, 2014.

[7] Xun S Zhou and Stergios I Roumeliotis. Robot-to-robot relative pose estimation from range measurements. *Robotics, IEEE Transactions on*, 24(6):1379–1393, 2008.

Matrix Methods for the Bernstein Form and their Application in Global Optimization

[1] Jihad Titi and [2]Jürgen Garloff

[1]Jihad Titi,
Department of Mathematics and Statistics, University of Konstanz,
D-78464 Konstanz, Germany
jihadtiti@yahoo.com

[2]Jürgen Garloff,
Faculty of Computer Sciences, University of Applied Sciences/ HTWG Konstanz,
D-78462 Konstanz, Germany
garloff@htwg-konstanz.de

Keywords: Bernstein polynomials, Bernstein coefficients, range enclosure, subdivision, convex optimization, optimality conditions.

Introduction

Solving optimization problems is of paramount importance in many real-life and scientific problems; polynomial global optimization problems form a significant part of them. One approach for their solution is based on the expansion of a polynomial into Bernstein polynomials, the so-called *Bernstein form*, see [1-5], [8]. This approach has the advantage that it does not require function evaluations which might be costly if the degree of the polynomial is high.
Shorthand notation for multi-indices is used; a multi-index $(i_1, \ldots, i_n)$ is abbreviated as i, where n is the number of variables. Comparison between and arithmetic operations with multi-indices are defined entry-wise. For $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$, its monomials are defined as $x^i := \prod_{j=1}^n x_j^{i_j}$, and the abbreviations $\sum_{i=0}^k := \sum_{i_1=0}^{k_1} \cdots \sum_{i_n=0}^{k_n}$ and $\binom{k}{i} := \prod_{\alpha=1}^n \binom{k_\alpha}{i_\alpha}$ are used.

We will consider the unit box $\mathbf{u} := [0,1]^n$, since any compact nonempty box $\mathbf{x}$ of $\mathbb{R}^n$ can be mapped affinely upon $\mathbf{u}$. Let p be an n-variate polynomial of degree l which can be represented in the power form as $p(x) = \sum_{i=0}^{l} a_i x^i$. We expand p into Bernstein polynomials over $\mathbf{u}$ as

$$p(x) = \sum_{i=0}^{k} b_i^{(k)} B_i^{(k)}(x), \quad k \geq l, \tag{1}$$

where $B_i^{(k)}$ is the *i-th Bernstein polynomial of degree* k, $k \geq l$, defined as

$$B_i^{(k)}(x) = \binom{k}{i} x^i (1-x)^{k-i}. \tag{2}$$

The coefficients of this expansion are called the *Bernstein coefficients* of p over $\mathbf{u}$ and are given by

$$b_i^{(k)} = \sum_{j=0}^{i} \frac{\binom{i}{j}}{\binom{k}{j}} a_j, \quad 0 \leq i \leq k. \tag{3}$$

The Bernstein coefficients can be organized in a multi-dimensional array $B(\mathbf{u}) = (b_i^{(k)})_{0 \leq i \leq k}$, the so-called *Bernstein patch*.

The Bernstein coefficients provide lower and upper bounds for the range of $p(x)$ over $\mathbf{u}$,

$$\min b_i^{(k)} \leq p(x) \leq \max b_i^{(k)}, \quad \text{for all } x \in \mathbf{u}. \tag{4}$$

Equality holds in the left or right inequality in (4) if and only if the minimum or the maximum, respectively, is attained at a vertex of $\mathbf{u}$, i.e., if $i_j \in \{0, k_j\}$, $j = 1, \ldots, n$.

We can improve the enclosure for the range of p given by (4) by elevating the degree k of the Bernstein expansion or by subdividing $\mathbf{u}$. The subdivision is more efficient than the degree elevation.

From the Bernstein coefficients $b_i^{(k)}$ of p over $\mathbf{u}$, we can compute by the de Casteljau algorithm the Bernstein coefficients over sub-boxes $\mathbf{u}_1$ and $\mathbf{u}_2$ resulting from subdividing $\mathbf{u}$ in the s-th direction, i.e.,

$$\begin{aligned} \mathbf{u}_1 &:= [0,1] \times \ldots \times [0,\lambda] \times \ldots \times [0,1], \\ \mathbf{u}_2 &:= [0,1] \times \ldots \times [\lambda,1] \times \ldots \times [0,1], \end{aligned} \tag{5}$$

for some $\lambda \in (0,1)$.

Bounding the range of a function over a box is an important task in global optimization when a branch and bound approach is applied. In the case that the optimization problem is convex we have the advantage that each local minimum is also a global one. Therefore, it is useful to know when a function is convex over a box. A well-known criterion for convexity is that the Hessian matrix is positive definite.

Main results

In our talk we present the following results:

- We propose a new method for the computation of the Bernstein coefficients of multivariate Bernstein polynomials which involves matrix operations such as multiplication and transposition and which is more efficient than the matrix method presented in [6].

- We present a new method for the calculation of the Bernstein coefficients over a sub-box by premultiplying the matrix representing the Bernstein patch by matrices which depend on the intersection point λ.

- As an application to global optimization, we propose a test for the convexity of a polynomial p. This check depends on the *interval Hessian matrix* that is obtained by the entry-wise application of the range enclosure property (4). Following [7], we test the positive semidefiniteness of this interval matrix which leads to the test for convexity of p.

References

[1] G.T. Cargo and O. Shisha, The Bernstein Form of a Polynomial, *J. Res. Nat. Bur. Standards* 70(B):79–81, 1966.

[2] J. Garloff, Convergent Bounds for the Range of Multivariate Polynomials, *Interval Mathematics 1985*, K. Nickel, Ed., *Lecture Notes in Computer Science*, vol. 212, Springer, Berlin, Heidelberg, New York, 37–56, 1986.

[3] J. Garloff, The Bernstein Algorithm, *Interval Comput.* 2:154–168, 1993.

[4] P.S.V Nataraj and M. Arounassalame, A New Subdivision Algorithm for the Bernstein Polynomial Approach to Global Optimization, *Int. J. Automat. Comput.* 4(4):342–352, 2007.

[5] S. Ray and P.S.V. Nataraj, An Efficient Algorithm for Range Computation of Polynomials Using the Bernstein Form, *J. Global Optim.* 45:403–426, 2009.

[6] S. Ray and P.S.V. Nataraj, A Matrix Method for Efficient Computation of Bernstein Coefficients, *Reliab. Comput.* 17(1):40–71, 2012.

[7] J. Rohn, Positive Definiteness and Stability of Interval Matrices, *SIAM J. Matrix Anal. Appl.* 15(1):175–184, 1994.

[8] M. Zettler and J. Garloff, Robustness Analysis of Polynomials with Polynomial Parameter Dependency Using Bernstein Expansion, *IEEE Trans. Automat. Control* 43(3):425–431, 1998.

BIFURCATION AND CONTINUATION OF HALO ORBITS - RIGOROUS NUMERICAL APPROACH

Irmina Walawska[1], Daniel Wilczak[1]

Faculty of Mathematics and Computer Science
Jagiellonian University
Łojasiewicza 6, 30-348 Kraków, Poland

Keywords: Halo orbits, three body problem, bifurcations, rigorous numerical analysis

Introduction

The Restricted Three Body Problem is a model equation for motion of a massless particle in the gravitational force of two large primaries. It attracts attention of many researches also because of its applicability to space missions, for instance the Genesis mission.

Main results

The system is given by

$$\begin{cases} \ddot{x} - 2\dot{y} = \frac{\partial \Omega(x,y,z)}{\partial x} \\ \ddot{y} + 2\dot{x} = \frac{\partial \Omega(x,y,z)}{\partial y} \\ \ddot{z} = \frac{\partial \Omega(x,y,z)}{\partial z} \end{cases} \tag{1}$$

where

$$\begin{aligned} \Omega(x,y,z) &= \frac{1}{2}(x^2+y^2) + \frac{1-\mu}{d_1} + \frac{\mu}{d_2} \\ d_1 &= \sqrt{(x+\mu)^2+y^2+z^2}, \\ d_2 &= \sqrt{(x-1+\mu)^2+y^2+z^2}. \end{aligned}$$

Our study is devoted to the analysis of an extended neighborhood for the collinear equilibrium points of The Restricted Three Body Problem. It was observed by Robert Farquhar that there is a family of symmetric, periodic orbits, parameterized by the amplitude 'z'. These orbits are called Halo orbits. Although they were found numerically, they have never been proven. We propose an algorithm for rigorous validation that the family of Halo orbits bifurcates from the family of well known planar Lyapunov orbits. We also give an algorithm for rigorous continuation of the family of Halo orbits. The method utilizes rigorous computation of higher order derivatives of well chosen Poincare map with symmetry properties of the system. As an application we give a computer assisted proof that the Halo orbits bifurcate from the family of Lyapunov orbits for wide range of the parameters μ that stand for the relative mass ratio of the two main bodies. For μ corresponding to the Sun-Jupiter system we give a proof of the existence of a wide continuous branch of Halo orbits that undergo period doubling bifurcation for some large amplitude 'z'. The computer assisted proof uses rigorous ODE solvers and algorithms for computation of Poincare maps from the CAPD library.

References

[1] Franco Bernelli Zazzera, Francesco Topputo, Mauro Massari, *Assessment of Mission Design Including Utilization of Libration Points and Weak Stability Boundaries*,

[2] Howell, K. C., Three-Dimensional Periodic Halo Orbits, *Celestial Mechanics*, Vol. 32, Issue 1, p. 53, 1984.

[3] I.Walawska, D. Wilczak, Validated continuation and bifurcations of Halo orbits, Preprint

[4] CAPD-Computer Assisted Proofs in Dynamics group, a C++ package for rigorous numerics, http://capd.wsb-nlu.edu.pl

Rigorous computation of Poincaré maps

Tomasz Kapela, Daniel Wilczak and Piotr Zgliczyński

Faculty of Mathematics and Computer Science,
Łojasiewicza 6, 30-348, Kraków, Poland
{Tomasz.Kapela,Daniel.Wilczak,Piotr.Zgliczynski}@ii.uj.edu.pl

Keywords: Poincaré section, Poincaré map, rigorous ODE solver

Introduction

There are many powerful topological and smooth tools for studying dynamics of maps. These are, in particular, the Interval Newton Operator, Brouwer degree, covering relations, cone conditions and many other.

In continuous time dynamical systems there are some tools that can be used for validation of some properties without integration of the system - for instance isolating blocks, Conley index and isolating segments. It turns out, however, that using the concept of Poincaré map combined with efficient ODE/PDE solver one can apply the well known tools available for maps for studying continuous-time dynamics. The applicability of this concept is proved by many examples, just to mention few of them:

- chaos in the Lorenz, Rössler systems and Planar Circular Restricted Three Body problem,
- the existence of choreographic solutions to the n-body problem,
- heteroclinic and homoclinic dynamics,
- various local bifurcations (like period doubling, homoclninic tangencies) and global bifurcations (cocoon bifurcations, Shilnikov homoclinic bifurcations),

- the existence and uniform hyperbolicity of attractors,
- periodic solutions to delay and partial differential equations.

All the above results have been obtained by the authors by means of Poincaré map techniques and ODE solvers. Therefore it is very important to have a good numerical algorithms for rigorous computation of Poincaré maps.

The talk will be based on our over 20 years of experience in designing and implementation of rigorous ODE solvers and interval tools for Poincaré maps. The software is available as a part of the CAPD library.

Main results

It turns out that obtained enclosures for Poincaré map can be significantly reduced by proper choice of Poincaré section. Even in the case when the Poincaré section cannot be changed (for instance due to some engineering reasons or presence of symmetries which should be preserved) we can still manipulate coordinate system in which we represent the arguments and values of Poincaré maps.

In the talk we will show that the optimal choice of Poincaré section nearby periodic orbit is related to left eigenvectors of the derivative of Poincaré map at a periodic point.

We will also give an algorithm for efficient computation of Poincaré map in a given coordinate system that takes into account internal representation of solutions in an ODE solver.

These heuristics are confirmed by tests we performed.

References

[**1**] CAPD - `http://capd.ii.uj.edu.pl` - the CAPD library, a C++ toolbox for rigorous dynamics.

[**2**] T. Kapela, D. Wilczak and P. Zgliczyński, Rigorous computation of Poincaré maps, *in preparation*.

Author Index