INTERNSHIP REPORT

Sergio Pertierre do Monte

SPID – Robotic Engineering

Akademia Górniczo-Hutnicza (Krakow, Poland)

Application of Interval Methods for the Solving Parametric Linear Programming Problems

2016

ABSTRACT

Considering the fact that interval arithmetic is well applied in robotics as in other several fields of science, learning it in more details has a great importance in a scientific career. Therefore, this internship was intended to improve this knowledge using interval analysis through two different and complementary activities: programming real applications to this theory in robotics using Python and using this theory to program iterative solutions to parametric interval linear systems. This first activity was gave by Professor Luc Jaulin of ENSTA-Bretagne in Brest, France.

However, the second activity, which carries a lot of theory, was the main part of the internship and it was gave by the Professor Iwona Skalna of Faculty of Management at AGH in Krakow, Poland. She was the tutor and she leads the work realized there. The activity realized there has to do with her particular interest, which is programming iterative methods to find better solutions to LIP systems called *p*-solutions (this corresponds to a linear interval form).

ACKNOWLEDGEMENTS

The internship opportunity I had in AGH (Akademia Górniczo-Hutnicza in Krakow) was a great chance for professional and personal development. I had assistance of excellent professors who helped me to work on this project about Interval Analysis and I am glad to have been part of it. I am also grateful for having a chance to been abroad and meet so many wonderful people and professionals who led me though this period.

I am using this opportunity to express my deepest gratitude and special thanks to Piotr Łebkowski, the dean of the Faculty of Management in AGH, for accepted me as intern in this great university.

I express my deepest thanks to Ms. Iwona Skalna, professor at the Faculty of Management in AGH, for her guidance, giving necessary advices, teaching me the issues for the work and arranging all facilities to make life easier. I choose this moment to acknowledge her priceless contribution.

I owe my sincere thanks to Mr. Luc Jaulin, professor at ENSTA-Bretagne, who in spite of being extraordinarily busy, helped me to find this internship, had availability to explain my doubts and taught me during the classes and in IAMOOC (MOOC on Interval Analysis with an application to parameter estimation and robot localization).

Finally, I would also like to thanks all my colleagues for their help in so many issues during my stay in Krakow.

CONTENTS

1. Introduction

Interval Analysis is a mathematical method developed as an approach to putting bounds on rounding errors and measurement errors (such as the position of a robot) in a mathematical computation and thus developing numerical methods to find the better results. Quite simply, it represents each value to a range of possibilities. Since that commonly it is not possible to have one perfect measure of any physical quantity, it is better to work with intervals.

The most common use is to monitor and handle rounding errors directly during the calculation and uncertainty in the knowledge of the exact values of the physical and technical parameters.

Due the possibility to apply this arithmetic in several fields of study, it is an important method to be studied and worked. Thus, this internship shows up as an opportunity to learn new applications. In addition, since it happens abroad, is also a way to improve the intercultural communication.

In addition, it is good to say that the mission of the AGH UST Faculty of Management is "to *support the development of modern management systems for organizations acting in the economy based on knowledge by means of creating and transferring advanced and practical knowledge in the field of economic and technical sciences*".

It is possible to separate the internship in two parts. The first one is about the application of this method in robotics using Python based on the course IAMOOC (better explained below) which consisted in solve some interval problems using the library PyIbex. The second part and the main part of the internship is implement different algorithms to figure out better solutions to parametric interval linear systems.

1.1. IAMOOC

IAMOOC is an online course given by Professor Luc Jaulin and Associate Professor Jordan Ninin, both at ENSTA-Bretagne, which shows a good introduction to Interval Analysis through several videos and solved exercises.

During the very beginning of the internship, it was necessary to redo all exercises on that course to recall some definitions of the interval analysis method, such as box and contractors.

Thus, Mr. Luc Jaulin gave some tasks based on IAMOOC courses to study more deeply the concept of intervals applied in robotics and to present them to Ms. Iwona Skalna, showing to her how this subject is worked at ENSTA-Bretagne, as an exchange of knowledge.

These exercises and their results are better explained in chapter 3.

1.1.1. PYIBEX

PyIbex is a Python library, which allows work with IBEX in a Python environment.

IBEX is a C++ library used to process limited real numbers. It gives algorithms for manipulating non-linear limitations (such round off errors), which is used on interval and affine arithmetic.

PyIbex library has several necessary functions as the algorithm SIVIA (Set Inversion via Interval Analysis), which solves any set inversion problem.

1.1.2. VIBES

Vibes is another Python library used to drawn figures. It is very useful to visualize the results as the figures (taken of an IAMOOC exercise done during the internship) on the next page show:

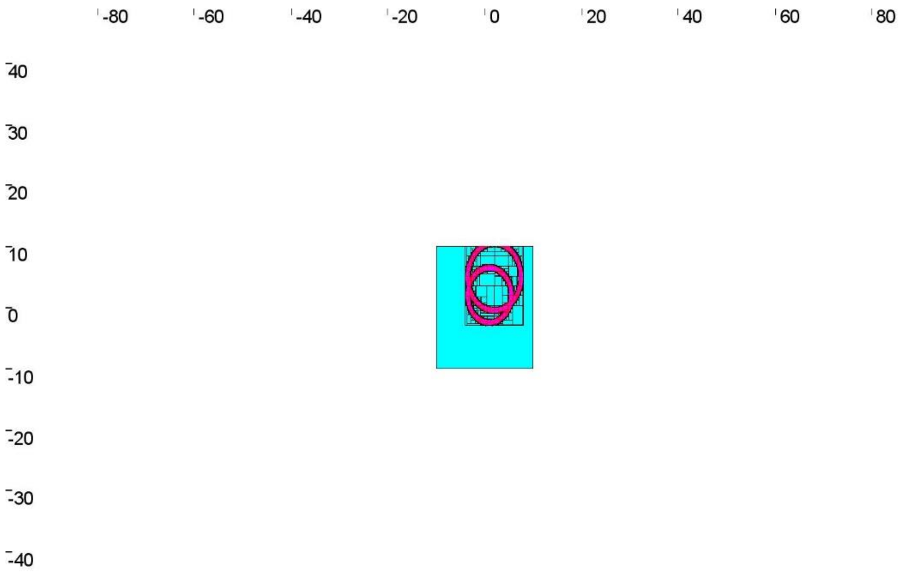


Figure 1: It shows two regions (rings) that could be a localization problem for example

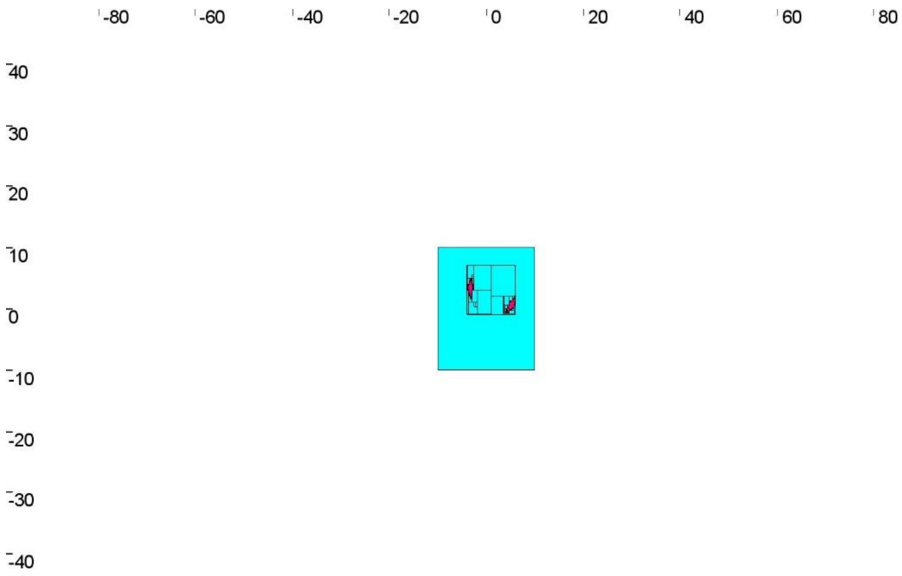


Figure 2: It shows the intersection of the previous rings on the Figure 1

1.2. Visual Studio

Visual Studio is an IDE made by Microsoft. It was used during most part of the internship to program mathematical computations with the intention to find better methods applying interval analysis.

Although using Python in IAMOOC exercises, all programs on this part of the internship were programmed in C++ using the libraries that Ms. Iwona Skalna had created. There are no graphical results as using Vibes but it is not important, because to achieve the main objective is just necessary to compare the numeric results.

The computations fulfilled into this part were based in papers of Professor Lubomir Kolev, where he teaches how to calculate these *p*-solutions. His activities are better explained on chapter 4.

2. What Is the Use of *p*-Solutions ant Its Relation with Interval Analysis Concept?

At first, the idea of the internship was to do parallel works with practical tasks to robotics given by Prof. Jaulin and the programs given by Prof. Skalna. This way it is possible to work theory and practice together. Nevertheless, due the time demanded to study and implement the *p*-solutions, it was not possible to continue doing the activities geared to robotics.

However, the course that the internship took was interesting. Focusing on the implementation of solutions to interval problems allowed deepening theory and learning new solving methods, which was the principal goal.

Ms. Skalna has been doing several researches in these methods. Also, she has been working in a book and in some papers which are important to the Faculty of Management as well. Thus, the study and implementation of these concepts are important as a serious research work.

As a result of this internship to the Faculty of Management, and principally Ms. Skalna, it is expected having let useful algorithms which show the utility of the two methods implemented. Also, the knowledge about the applications of interval arithmetic in the area of robotics (using the libraries PyIbex and Vibes) as real examples.

3. Using Interval Analysis to Solve Some Robotic Localization Problems

As mentioned before, this was the first part of the internship. It consisted in accomplish tasks given by Professor Luc Jaulin programming in Python. They were two interval arithmetic problems with application in robotics and the main idea was take advantage of the knowledge learned in class and doing the exercises of IAMOOC.

3.1. Task 1: Find a robot using coordinate matrices with errors

This problem was the localization of a robot using two matrixes of coordinates (D and α) with errors, it means, interval matrixes. There are four captors positioned symmetrically and each one of them knows a region where the robot is. These captors measure these regions with an interval of angles "α" and an interval of distances "d". Thus, the matrix D represents all distances d measured and the matrix A, all angles α.

To solve this problem it just was necessary find an intersection of all these measurements, which represents a small area than the union of all regions. However, in certain cases, one or more captors could be doing wrong measurements, what prevents the algorithm to find the intersection. Therefore, it is necessary to give some degrees of freedom that we call number of outliers. For example, if the number of outliers is equal to one, it means that the program will find all intersections possible ignoring one region at the time and after it will trace their union.

Fortunately, it was not the case, but even so, it was put the result of the same problem with different numbers of outliers to see what happens. It is possible to check later in this chapter these results.

3.1.1. Task Accomplishment

This first task took one week to be fulfilled. It was relatively easy, but in order to remember the theory to solve the problem it was necessary review the exercises on IAMOOC, then apply the same logic into the problem. Therefore, it was a good way to recall the knowledge learned during the semester.

3.1.2. Results

Thanks to Vibes, it is possible to visualize the four beacons in black and the regions that they can measure in red. Each figure shows the result with a certain number of outliers.

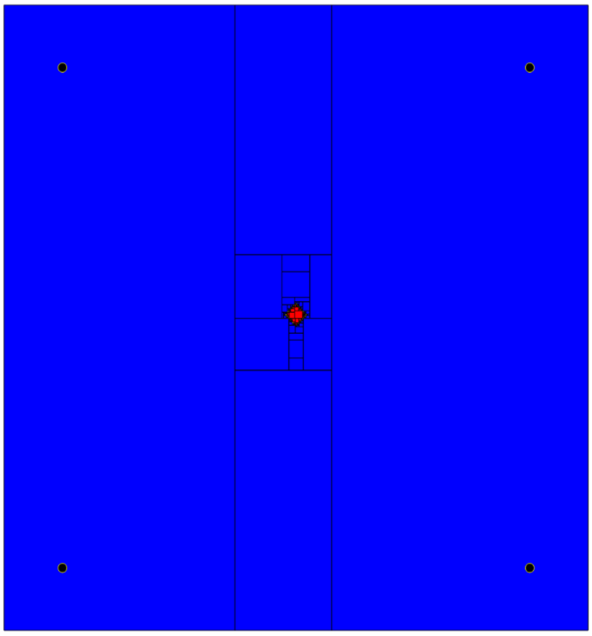


Figure 3: Result with zero outliers

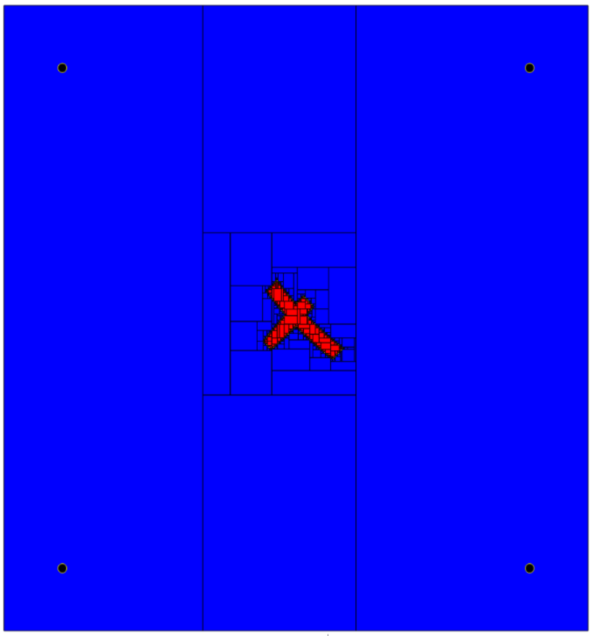


Figure 4: Result with one outlier

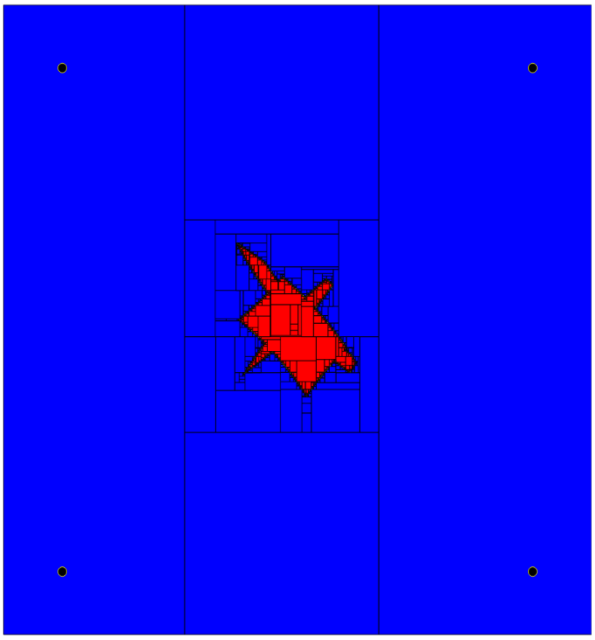


Figure 5: Result with two outliers

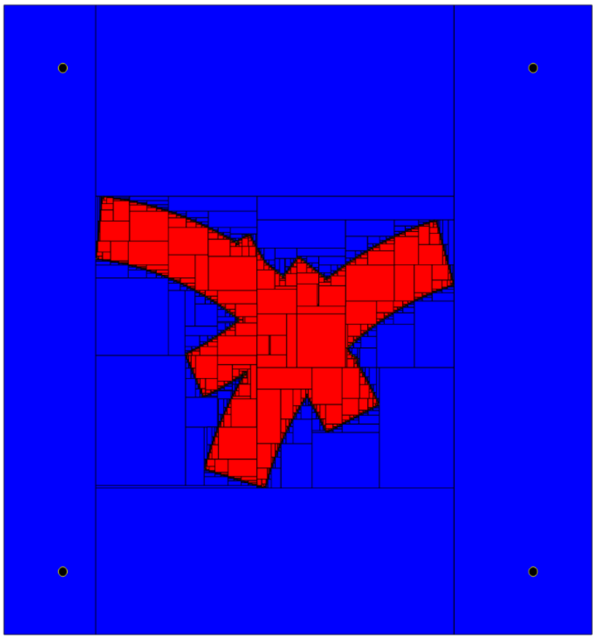


Figure 6: Result with three outliers

As shown in the figures, the more the number of outliers increases, the less accurate is the result. The figure 6 represent the union of all regions because when the number of outliers is equal to the number of regions minus one, it means that each region will be completely represented.

3.2. Task 2: Find a Mobile Robot

This problem looks like the previous one, but now it consists in doing the robot following a trajectory and it may use the estimated position for its control law. This task is similar also to the exercises in IAMOOC, which are useful to solve this problem.

3.2.1. Task Accomplishment

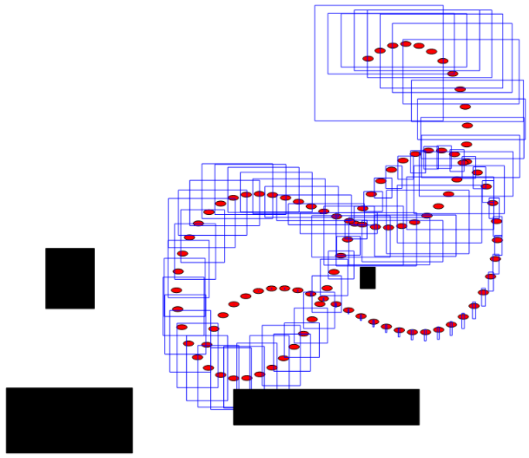
This exercise took two weeks because it was necessary understand better some concepts as doing the robot control law and how calculating the estimated position for each time, which is a common problem in mobile robotics.

Then, it was necessary more time to solve the problems on the algorithm to finally show and explain to Ms. Skalna how the program works and how the interval analysis is used.

3.2.2. Results

Again, using Vibes is possible to see the four beacons in black, the trajectory (the robot position along the time) in red and the estimated position of the robot in blue boxes.

To show better the dynamic of the problem, the program was configured to plot some positions over the time as the figures on next page represent:



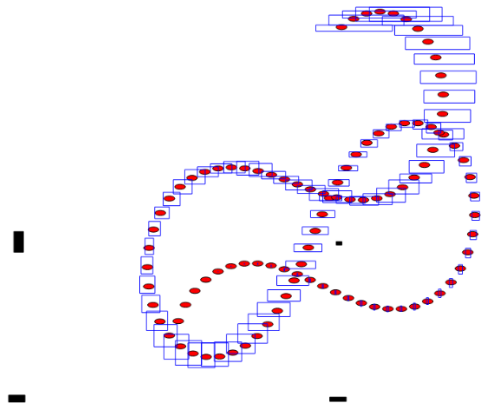


Figure 7: Dynamic of the robot localization on the first iteration *Figure 8: Dynamic of the robot localization on the last iteration*

This kind of computational problem is common in robotic mapping, called SLAM (Simultaneous Localization and Mapping).

When comparing both figures 7 and 8 it is possible to perceive the evolution of the estimation of the robot after some iterations (better accuracy of the interval box, which contains the estimated position). This was the expected result.

4. Applying Interval Methods to Solve Parametric Linear Programming Problems

Professor Iwona Skalna has a particular interest testing different models of affine arithmetic implementation to find the bests solutions. She is writing a book where she explains these methods and show the results. Therefore, in this part of the internship, she gave some papers written by Mr. Lubomir Kolev, professor at the Department of Theoretical Electrotechnics at Technical University of Sofia (Sofia, Bulgaria). On these papers, he teaches his methods with examples and some applications. As the language used is a little complex to people who are not accustomed to work with interval analysis and large iterations, Ms. Skalna gave some explanations and examples to teach the basic model used by this professor.

Then, after teach the mathematical models applied into the programs, Ms. Skalna gave two main problems for the rest of the internship: a linear parametric problem and a quadratic parametric problem.

Both iterative methods consist in take a problem in matrix format where to find the solution is needed do an iterative process over a *p* parameter. During the iteration on the linear case, when the equation arrives in a quadratic exponent (p^2), a linearization is applied to reduce this term. Similar in the quadratic case, but it happens when the equation is cubic.

To stop the both iterations, the stopping criteria is when the difference between the actual solution and the previous one is less than a default value.

4.1. Organization

In order to accomplish these tasks during the remaining time of the internship, a table of functions was created, as it is possible to see below.

It was better and more logical start with the linear method (since the quadratic case could see as an improved version of the linear case).

Linear case:

Activity	Expected deadline	Real end date of work
Study and theoretical work (method written on the paper)	10/07	13/07
Implementation of the algorithm (without the recursive part)	17/07	22/07
Full functional algorithm	31/07	28/07

As it is possible to see, it took more time to learn and write the solution of the exercise than it was expected. The implementation of the first part of the algorithm took more time as well. Nonetheless, once finished this part, doing the iterative part was simpler.

After finish the linear case, the results obtained were not satisfactory. However, no errors were found and the mathematical method seems to be correct, so it was just a limitation of the method.

Quadratic case:

Activity	Expected deadline	Real end date of work
Study and theoretical work (method written on the paper)	14/08	19/08
Implementation of the algorithm (without the recursive part)	28/08	02/09
Full functional algorithm	03/09	05/09

The time taken to search possible errors on linear case added with the fact that the quadratic case has a theory more complex, the programmed schedule could not be followed.

This delay did not influence too much the last activity, which was not accomplished. The problem was the result of the implementation of the first part of the algorithm that were not that better as expected. Nevertheless, some revisions were done looking for errors as well and finally an error was found. After repair the problem on the code, the program works much better as expected.

It is worth remembering that Ms. Skalna give her assistance throughout all internship, teaching the theory with notes, helping to find errors and correcting some of them.

4.2. Linear Parametric Solution

The method computing linear type *p*-solution to the parametric interval linear systems is relatively simple and commonly used to find solutions that do not need be very accurate.

This method takes initially a system with a matrix *A* and a vector *b* as showed below:

$$A(p)x = b(p)$$

Where *p* is an interval. Then, applying the concepts on the papers of Prof. Kolev, the objective is to find a linear solution to *x* in the form:

$$x(p) = c + Lp + s$$

Where *c* is the sum of all linear parts, *L* is the operator and *s* an interval.

4.2.1. Implementation

At first, the functions created by Ms. Skalna called “ktransform” and “gausse” on the program takes as parameters the matrix A and the vector b and apply linear transformation to change this system to obtain an ideal symmetric interval p = [-1,1]. This interval facilitates a lot the calculations and it is no needed to revert this decomposition after obtain the results.

Then, the function “gausse” returns the value of the first solution x(0) (called midpoint solution).

After, using the theory learned the iterative system “grows” until a quadratic equation. Then is necessary apply a linear approximation (hence the method name) to reduce it. Thus, put in a loop with the stopping criteria said earlier using the difference between the actual solution and the previous one.

At the end of each iteration, the program put all terms of the solution into a function called “kolevReduce” to reduce the parametric solution in an interval.

4.2.2. Results

The results were not satisfactory because the interval solution was always growing instead reducing. Then, after the revision of the algorithm, the conclusion was that the problem was not an error on the code or the concept, but a limitation of this method. Actually, it works well with an uncertainty equals zero, which leads to say that this method can be used with small uncertainties. Nonetheless, the result, even when its uncertainty is equal to zero is not very precise as it should be.

The figure below can show how the solution grows quickly (in just 10 iterations) instead of reducing, even with a little uncertainty as 0.1:

```
Invite de commandes
C:\Users\Sergio\Documents\ENSTA\UV4.7\Python\Internship\IS_IntervalPr
Running threads: 4
v
0 [-0.492063492063492, 1.06349206349206],
1 [-0.0835978835978836, 0.178835978835979],
2 [-2.95555555555556, -0.187301587301588]
c2
0 0.0586064324731505,
1 -0.0247997327674767,
2 -0.0551156912577817
=============
L2
<0.548008424720693, -0.0535158673755152, -0.180498161931023, >
<-0.0351742607554897, 0.0143115178750815, -0.0944705546145479, >
<-0.71648985481457, 0.00539220664582896, 0.674671158105781, >

=============
t3
0 6.50917081117004,
1 2.28157858607684,
2 9.01252685699309
=============
p-solutionIdeal
0 [-6.94687254700983, 7.6355139833847],
1 [-2.40271560447038, 2.44835423417353],
2 [-12.0356243392456, 8.78253581387292]
=======================================================


x2
0 [-1.77595238095238, 2.48738095238095],
1 [-5.18904761904762, 5.27845238095238],
2 [-6.12059523809524, 3.0302380952381]
=============
Niter: 10
0.008
p-solution
0 [-3176147655.4916, 3177561249.12551],
1 [-6357752004.58302, 6353735781.29249],
2 [-6251238875.33327, 6252583319.6287]
Uncertainty = 0.1
=======================================================
```

Figure 9: Comparison between the ideal linear p-solution and the liner p-solution with uncertainty = 0.1

The “p-solutionIdeal” represents the values that this method tends with uncertainty zero and the “p-solution” shows the result with uncertainty 0.1.

4.3. Quadratic solution

The quadratic *p*-solution to the parametric interval linear systems is quite similar as the previous case. However, due its complexity, it was required more time to understand the theory. This method is an improvement of the linear case, being more accurate once that the interval $[-1, 1]^2 = [0, 1]$, what reduces the interval of the solution.

Ms. Skalna initially wanted to start directly with this method, but after she thought that it would be easier and better to learn the theory starting with the previous one. Therefore, the developing of the quadratic case could see as the main objective of the internship.

The objective now it is to find a linear solution to *x* in the form:

$$x(p) = d + Cp + Qp^2 + s$$

Where *d* is the sum of all linear parts, *C* and *Q* are the operators and *s* an interval.

4.3.1. Implementation

All the process follow the same logic than the linear parametric solution. The only difference in the implementation is the linearization part. Nevertheless, even if the methods is called quadratic, here the approximation was also linear because it was applied a linear approximation in the cubic equation (bus just with the terms *p* and p^3).

The calculation of the matrix *Q* (which multiplies the vector p^2) was somewhat difficult and the calculation of the *p*-solution was even harder because it was needed calculate term by term from the result of matrix calculations. Then find a logic and put in four “for” loops.

In addition, as this method has one term more, it was necessary change the function “kolevReduce” to find the final solution in the form of an interval.

The problem found here was after finish the algorithm without the iteration part, because the results were not good as expected. Then, it took some time to find the error (on the linearization part) and repair it. However, after this, it was easier to proceed with the resolution.

4.3.2. Results

The results were very satisfactory here. Although the linear case was not that good, in the quadratic iteration is possible to observe better values of intervals as expected.

Ms. Skalna has an indispensable role in this part, solving the errors found on the algorithm, which, when compiled, presented strange values even before the iterative part. Then, thanks to her help, it was possible to reach the internship's objective within the time limit.

The figure on the next page presents the comparison between the ideal previous linear solution and the quadratic new one:

```
C:\Users\Sergio\Documents\ENSTA\UV4.7\Python\Internship\IS_Interval
Running threads: 4
v
0 [-0.492063492063492, 1.06349206349206],
1 [-0.0835978835978836, 0.178835978835979],
2 [-2.95555555555556, -0.187301587301588]
c2
0 0.0586064324731505,
1 -0.0247997327674767,
2 -0.0551156912577817
=============
L2
<0.548008424720693, -0.0535158673755152, -0.180498161931023, >
<-0.0351742607554897, 0.0143115178750815, -0.0944705546145479, >
<-0.71648985481457, 0.00539220664582896, 0.674671158105781, >

=============
t3
0 6.50917081117004,
1 2.28157858607684,
2 9.01252685699309
=============
p-solutionIdeal
0 [-6.94687254700983, 7.6355139833847],
1 [-2.40271560447038, 2.44835423417353],
2 [-12.0356243392456, 8.78253581387292]
=======================================================

Niter: 10
Iterations: 10
0.013
p-solution
0 [-3.90759907677803, 4.63098589329042],
1 [-1.31383574531315, 1.36270089017113],
2 [-8.03212622787874, 4.70682822932014]
Uncertainty = 0.1
=======================================================
```

Figure 10: Comparison between the ideal linear p-solution and the quadratic p-solution with uncertainty = 0.1

As it is presented, this solution is even lesser than the solution expected for the linear case proving that the quadratic form has a reliable accuracy.

Normally, it is necessary to calculate a radius value for the uncertainties to know the limitations where this method works. However, there was not enough time remaining to do it.

5. Contribution to the student's professional project

Normally in classes, a library already done as PyIbex is used to develop and solve robotic localization problems. However, during this work realized in AGH, it was necessary to work directly with theory programming these *p*-solutions what requires more knowledge on the subject. In addition, the experience of heaving an excellent tutor as Ms. Skalna was indispensable to learn faster because she was regardful, helping whenever it was necessary and preparing notes to explain in the best way.

During the internship, it was very great and useful learning new subjects about programming, iteration methods and interval arithmetic, which certainly will have great uses for the next projects and works. Nevertheless, besides all the knowledge necessary to follow a scientific career as a professional project, there are some other important skills to develop.

As an experience abroad in a country with a culture so different compared to France, the intercultural exchange was intense making an improvement in the intercultural communication skills and in the formal English. Other strong points of a researcher.

Doing research tasks showed a little how is working in this career. Sometimes it is not satisfactory like spending time trying to find a good solution just to perceive that it was another wrong one. On the other hand, it serves as motivation to keep researching.

Then, one of the most important experiences in this internship was being with inspirational researchers, who were always available to teach something, and accompany closely their works (in particular Ms. Skalna who taught several things about the subject she was working).

Therefore, uniting all these points learned on these 10 weeks, it is possible to affirm that this experience was unique and extremely useful to create a great base to the future career.

6. Conclusion and Perspectives

The main part of the internship, the quadratic type *p*-solution to the parametric interval linear systems, presents the best solution. Its results were very satisfactory due the accuracy using small uncertainties. On the other hand, the results of the linear type were not good and after some revisions, it was assumed that the problem is on the limitation of the mathematical model.

As recommendation, it is necessary to calculate the valid radius of the accuracy of the quadratic method to know until where reliable value of uncertainty this method works well. Moreover, do a last review in the algorithm of the linear part just to confirm the problem in the model.

6.1. Perspectives

Some ideas to do in a possible future project, if someone is interested to do an internship in AGH also:

1. Use this quadratic method or one of the best methods found by Ms. Skalna and implement directly in ROS (Robot Operating System) to use in robot localization problems.

2. In the quadratic iterative method, as it was mentioned, it was applied a linearization with the terms p and p^3. Instead of that, it would be interesting to apply a quadratic approximation to the whole equation just to see if the results is even better.

3. Meliorate the second task (3.2) given by Mr. Jaulin, putting more robots and making them localize each other to be grouped.

7. References

[1] Lubomir Kolev. A Class of Iterative Methods for Determining p-Solutions of Linear Interval Parametric Systems.

[2] Lubomir Kolev. *SWIM'5: A New Class of Iterative Interval Methods for Solving Linear Parametric Systems*.

[3] Lubomir Kolev. *Parameterized Solution of Linear Interval Parametric Systems*.

[4] Luc Jaulin. *Mobile Robotics.*

[5] Iwona Skalna. *A method for Outer Interval Solution of Parameterized Systems of Linear Equations*.

[6] Ramon E. Moore, R. Baker Keartfott and Michael J. Cloud. *Introduction to Interval Analysis.*

[6] http://iamooc.ensta-bretagne.fr/

[7] http://enstabretagnerobotics.github.io/VIBES/

[8] https://www.visualstudio.com/fr/vs/

[9] http://www.agh.edu.pl/en/wydzial-zarzadzania/

[10] https://www.ensta-bretagne.fr/jaulin/robmooc.html