



HAL
open science

Validation of the odometric model for the purpose of the numerical analysis

Edern Ollivier

► **To cite this version:**

Edern Ollivier. Validation of the odometric model for the purpose of the numerical analysis: A pool of trajectories for the engineers of the automation of the fair and automated vehicle. 2020. hal-02977289v3

HAL Id: hal-02977289

<https://hal.science/hal-02977289v3>

Preprint submitted on 8 Apr 2021 (v3), last revised 28 May 2021 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Validation of the odometric model for the purpose of the numerical analysis

Edern Ollivier, *Member, IEEE*

edern.ollivier.fr@ieee.org

17 RUE DE L'AMIRAL HAMELIN at Paris 16 (75116)

Abstract—The odometric model is validated herein. We have been starting to simulate, integrate, explain and then to validate the odometric model. The model of validation is to be programmed in the Python language, with the help of a bridge for RTMaps 4.6.2. The matrix of command is to be added with the matrix of state to be also given to the interpreter, or maybe to the compiler, because it is used to be a difficulty for RTMaps 4.6.2 itself to multiply the matrixes one by one, please see at https://intempora.com/licenses/python_3.8.0.txt.

Index Terms—Odometry; Vehicle location and navigation systems; Mobile positioning systems; Trajectory; Discretization of a model; Control of a vehicle; Stability analysis.

I. INTRODUCTION

As one introduction, I could say that the odometric model is well known now and that everything is shaped for the validation to be executed under RTMaps 4.6.2.

This platform of validation has been developed by the CAOR with Mister Bruno Steux, Mister Olivier Meunier and Mister Nicolas du Lac, and then is intensively and fluently used by Renault SA for developing the laws of control of the components of their vehicle, and then used as a standard to validate the behaviour of such an autonomous vehicle.

II. VALIDATION

I want here to do the validation because the set of equations is now available as the document “Simulation, integration and explanation of the odometric model for the purpose of the numerical analysis”, [1], and ready to be programmed. There is a version of

RTMaps named 4.6.2 which is fully offered free of duty to me with the bridge for Python under the commercial licence ... So that my informatical support is based on the Intempora SA team of support of engineers at Issy-les-moulineaux, headed by Mister Nicolas du Lac, the CEO of Intempora SA. I should underline that RTMaps is a major key product coming from the School of the Mines of Paris previously headed by Mister Claude Lurgeau, retired from the laboratory of the CAOR, and actually headed by a famous professor and also sometimes researcher in the field of the probabilities.

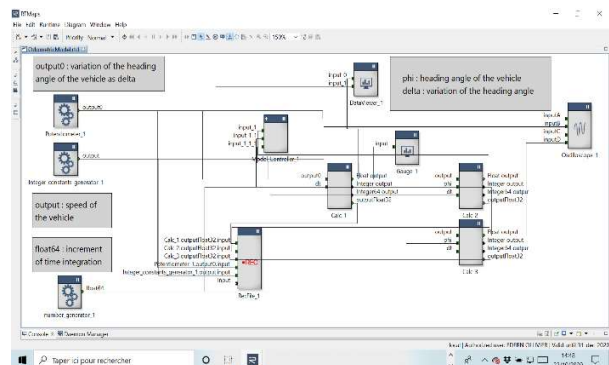


Figure 1: Diagram of validation under RTMaps 4.6.2

My mathematical background is coming from the course of mathematics of PSI*, the class to improve the knowledge in the industrial field of mechanics, automatics, mathematics, and also based on the book of Mister Jean François Ruaud, as a professor of mathematics at Lakanal in PSI* and then at Saint Louis in MP, who published “Mathématiques 2ème année” to compile all the course of MP, PC and PSI,

[2], with the work of colleagues at School of preparation of the high Schools of engineering such as the ENS Ulm/Cachan, Polytechnique, Centrale, the Mines and the ENSI.

III. EXAMPLE OF THE MULTIPLICATION OF TWO MATRIXES

Here is an example of the multiplication of two matrixes given free of charge to me from the support of Intempora SA.

#This is a template code. Please save it in a proper .py file.

```
import rtmmaps.types
import numpy as np
import rtmmaps.core as rt
import rtmmaps.reading_policy
from rtmmaps.base_component import
BaseComponent # base class
```

Python class that will be called from RTMaps.

```
class rtmmaps_python(BaseComponent):
```

```
def __init__(self):
```

```
BaseComponent.__init__(self) # call base class
constructor
```

```
def Dynamic(self):
```

```
self.add_input("in1", rtmmaps.types.MATRIX) #
```

```
define input
```

```
self.add_input("in2", rtmmaps.types.MATRIX) #
```

```
define input
```

```
self.add_output("out", rtmmaps.types.MATRIX)
```

```
# define output
```

Birth() will be called once at diagram execution startup

```
def Birth(self):
```

```
pass
```

Core() is called every time you have a new input

```
def Core(self):
```

```
ioeltin1 = self.inputs["in1"].ioelt
```

```
ioeltin2 = self.inputs["in2"].ioelt
```

```
matrix1 = ioeltin1.data
```

```
matrix2 = ioeltin2.data
```

```
ioelt_out = rtmmaps.types.Ioelt()
```

```
ioelt_out.data = rtmmaps.types.Matrix()
```

```
ioelt_out.ts = ioeltin1.ts
```

```
ioelt_out.data.matrix_data =
```

```
matrix1.matrix_data.dot(matrix2.matrix_data)
```

```
self.outputs["out"].write(ioelt_out) # and write
it to the output
```

Death() will be called once at diagram execution shutdown

```
def Death(self):
```

```
pass
```

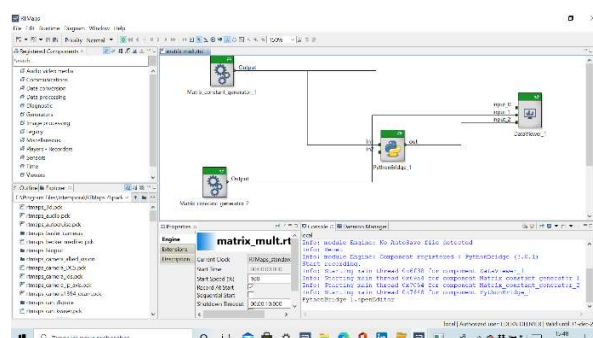


Figure 2: Multiplication of two matrixes

So that I can get the values in a table format.

Time of issue	1	2	3
07803029 (1433302963)	1	2	3
07803064 (143330764)	4	5	6
	7	8	9
	10	11	12
Time of issue	1	2	3
07803070 (143330830)	1	2	3
07803074 (143330834)	4	5	6
	7	8	9
	10	11	12
Time of issue	1	2	3
07803070 (143330830)	22	33	44
07803074 (143330834)	33	44	55
	44	55	66
	55	66	77

Figure 3: Table of values for the multiplication of two matrixes

I have to thank the support of Intempora SA for his kindness.

IV. INTEGRATION OF THE CONTROL OF THE LINEARIZED ODOMETRIC MODEL

I integrate the equations of the linearized odometric model in a script of Python, also interpreted by the motor of calculus of RTMaps 4.6.2, this is easy and absolutely free of charge.

Well I have done the scheme of validation under RTMaps 4.6.2, the issue is well known now, just that I cannot create the matrix with the given values of it, also in the diagram of validation, such as the matrix of state for the odometric model to be fully integrated, I should complain a lot about it, asking to the team of support of Intempora SA.

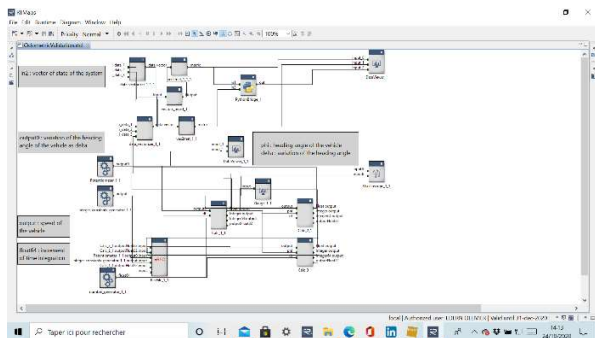


Figure 4: Diagram of validation without the matrix of state

Sure, it has to be developed, as far as I can understand and pick up the data_matrixer's component in RTMaps 4.6.2, I don't see it in the menu and I have to improve the panel of component.

Well I should do it in the Python's component, it will be a bit interesting to show the feed back of it. I think that it can be done also in the Python's component with the bridge.

V. STABILITY

The stability has been proven by the example as follow, it has been given by the solver of the MapleSim solver which is the next to the Maple suite of developments, which is one of the best known of his class of resolution, and finally it has been done with the odometric model detailed in the figures below.

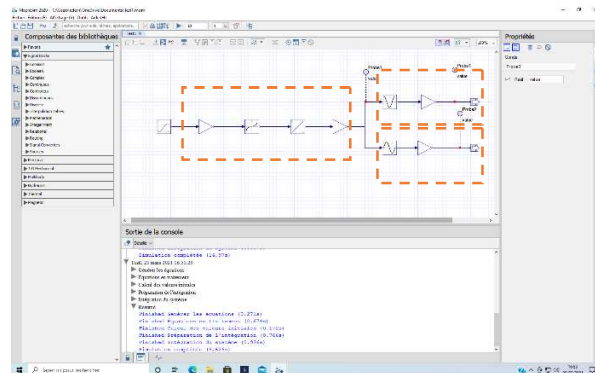


Figure 5: Model of the odometry with the Tan, Sin and Cos subsystems

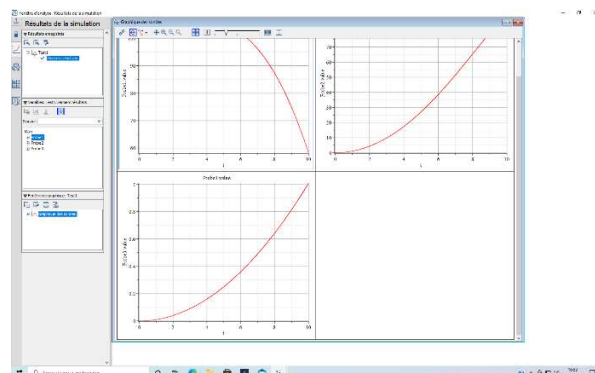


Figure 6: X, Y and Phi

The stability is ensure by the integrator in the given model of the odometry, this is a proven theorem from the course of automatic of the PSI* from the preparation of the high schools of engineering, and more it is to say that the margins of gains and phase are given by the figures below.

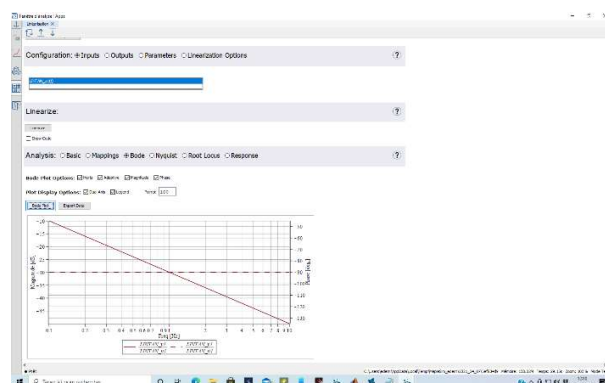


Figure 7: From the linearized Tan subsystem

Please see the gain here.

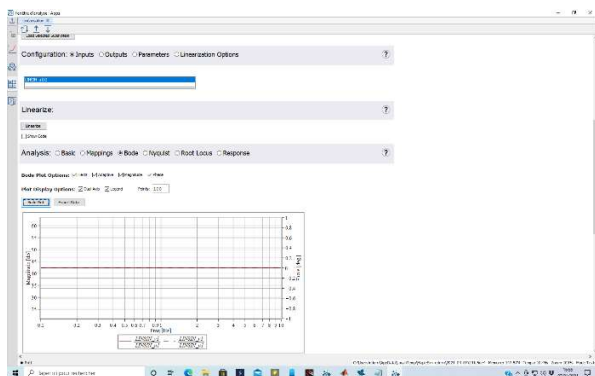


Figure 8: From the linearized Sin subsystem

Please see the phase here.

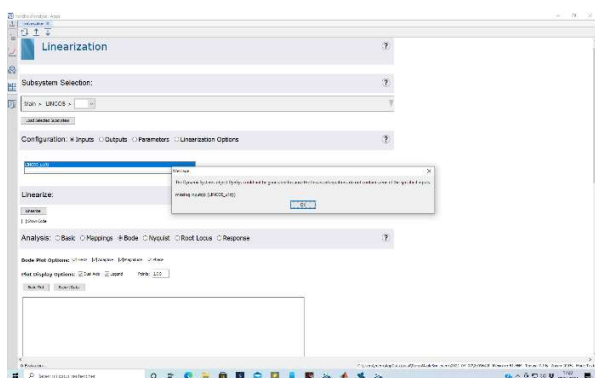


Figure 9: From the linearized Cos subsystem

All these figures are given for proving the stability of the odometric model, I can explain that the transfert function should not go under zero at the point of minus one hundred and eighty degree of phase in the Bode diagram of stability. The Cos subsystem is majored by one multiplied by the speed of the vehicle here.

VI. CONCLUSION

As one conclusion, I could say that the odometric model is well known now and that everything is done for the commands of a model of automation to be validated with the odometric model.

ACKNOWLEDGMENT

The computer where the simulation has been provided has got the RTMaps 4.6.2 suite and MapleSim 2020 for the simulations to be

executed.

I would like to thank the team of support of Intempora SA, leded by the CEO of Intempora SA.

I would also thank Mister Nicolas Cottureau for the support given with the Maplesoft products which are Maple and MapleSim.

REFERENCES

- [1] **Edern Ollivier**. Simulation, integration and explanation of the odometric model for the purpose of the numerical analysis: A pool of trajectories for the engineers of the automation of the fair and automated vehicle. 2019. (hal-02162501v8) ;
- [2] **Claude Deschamps, André Warufsel, JFR, FM, JCS, AM**, Mathématiques 2^{ème} année. Cours et exercices corrigés. 2^{ème} année MP, PC, PSI. Dunod.



Edern Ollivier was born in Paris XI, Paris, France in 1977. He received M.S. degrees in automation from the ENSMM, Besançon, in 2001 and the M.S. degrees in computer engineering from the FH of Karlsruhe, Karlsruhe, Germany, in 2001.

From 2001 to 2004, he was an Expert Engineer with the Inria of Rocquencourt. His research interests include the developpment of the Cybercars, SLAM, EKF and Front End Electronics.