Projet de Fin d'Études (PFE)

Ingénierie système : robotique et systèmes embarqués

2014/2015

Réf : DIASI / 15-351

Local Dynamic Path Planning for an Autonomous Forklift in Human Environment

Unclassified Report

Can be made public on the internet

Author:

José Magno Mendes Filho — Promotion 2014

Supervisor - ENSTA:
David Filliat

Supervisor - CEA:
Éric Lucet

Internship from 05 Mars 2015 to 28 August 2015

CEA LIST Digiteo Moulon
Bât. 660 91191 GIF-SUR-YVETTE Cedex, France

Acknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Résumé

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

Part I

Internship Description

Chapter 1

Work Descritpion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Part II

Internship Contribution

Chapter 2

Global Near-optimal Solution for Path Planning

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2.1 Description of the Problem

Chapter 3

Algorithmic Approach

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3.1 Optimizers

There is a variety of numerical optimization packages implemented in many different programming languages available for solving optimization problems [10]. Each of them may have their own way of defining the optimization problem and may or may not support specific kinds of constraints (equations, inequations or boundaries).

For the initial implementation written in python two packages stood out as good, easy-to-use options for solving the constrained optimization problem that models the planning motion task.

Scipy is a vast open-source scientific package based on python that happens to have a minimization module. Within this module many minimization methods can be found. For this specific optimization problem, only the method SLSPQ was appropriate. It was the only one to handle constrained minimization where the constraints could be equations as well as inequations.

pyOpt is a much smaller ecosystem than Scipy that is specialized in optimization. It gathers many different numerical optimization algorithms some of them free and some licensed. Again, among all of them there were only a few suitable for this problem which were also free: SLSQP (same as the one implemented within Sicpy), PSQP and ALGENCAN.

SLSQP and PSQP are both SQP (for sequential quadratic programming) methods. A SQP method attempts to solve a nonlinearly constrained optimization problem where the object function and the constraints are twice continuously differentiable. It does so by modeling the object function ($\min f(x)$) at the current iterate x_k by a quadratic programming subproblem and using the minimizer of this subproblem to define a new iterate x_{k+1} [9].

The ALGENCAN method

describe algecan

$$\min_{(t_{final}, C_0, \ldots, C_{d+n_{knot}-2})} J = (t_{final} - t_{initial})^2 \tag{3.1.1}$$

under the following constraints $\forall k \in \{0, \ldots, N_s - 1\}$:

$$\begin{cases} \varphi_1(z(t_{initial}), \ldots, z^{(l-1)}(t_{initial})) & = & q_{initial} \\ \varphi_1(z(t_{final}), \ldots, z^{(l-1)}(t_{final})) & = & q_{final} \\ \varphi_2(z(t_{initial}), \ldots, z^{(l)}(t_{initial})) & = & u_{initial} \\ \varphi_2(z(t_{final}), \ldots, z^{(l)}(t_{final})) & = & u_{final} \\ \varphi_2(z(t_k), \ldots, z^{(l)}(t_k)) & \in & \mathcal{U} \\ d_{O_m}(t_k) & \geq & \rho + r_m, \quad \forall O_m \in \mathcal{Q}_{occupied} \end{cases} \tag{3.1.2}$$

— Problem with discretization

Try adding CONSTRAINTS related to max acceleration (DONE)
For that we have to increase the maximum derivative order of the flat output needed so we calculate $[\dot{v}\ \dot{\omega}]$ building a φ_3 function
Also, the constraints to be added:

$$\varphi_3(z(t_k), \ldots, z^{(l)}(t_k)) \in \mathcal{A}$$

where $\mathcal{A}$ is the set of admissible acceleration values.
The function φ_3 is as follows:

$$\varphi_3(z(t_k), \ldots, z^{(3)}(t_k)) =$$
$$= \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial t}\|\dot{z}\| \\ \frac{\partial}{\partial t}\frac{(\dot{z}_1\ddot{z}_2 - \dot{z}_2\ddot{z}_1)}{\|\dot{z}\|^2} \end{bmatrix} = \begin{bmatrix} \frac{\dot{z}_1\ddot{z}_1 + \dot{z}_2\ddot{z}_2}{\|\dot{z}\|} \\ \frac{(\ddot{z}_1\ddot{z}_2 + z_2^{(3)}\dot{z}_1 - (\ddot{z}_2\ddot{z}_1 + z_1^{(3)}\dot{z}_2))\|\dot{z}\|^2 - 2(\dot{z}_1\ddot{z}_2 - \dot{z}_2\ddot{z}_1)\|\dot{z}\|\dot{v}}{\|\dot{z}\|^4} \end{bmatrix}$$

— Remake code using good objected oriented structure. It will be good for C++ part (DONE)

ONLINE T_c and T_p (planning horizon) "given" (arbitrary).

$$\tau_k = t_{initial} + kT_c \quad k \in \mathbb{N}$$

Arbitrary detection radius for the robot sensors. Only if the obstacle characteristic position is inside the detection zone the obstacle is considered detected. Using $2m$.

Evaluate for each time interval $[\tau_{k-1}, \tau_k)(k \in \mathbb{N})$ the trajectory beginning at τ_k until $\tau_k + T_p$:

$$\min_{(C_{(0,\tau_k)}, \ldots, C_{(d+n_{knot}-2,\tau_k)})} J_{\tau_k} = \|\varphi_1(z(\tau_k + T_p, \tau_k), \ldots, z^{(l-1)}(\tau_k + T_p, \tau_k)) - q_{final}\|^2 \tag{3.1.3}$$

under the following constraints $\forall t \in [\tau_k, \tau_k + T_p]$:

$$\begin{cases} \varphi_1(z(\tau_k, \tau_k), \ldots, z^{(l-1)}(\tau_k, \tau_k)) & = & q_{ref}(\tau_k, \tau_{k-1}) \\ \varphi_2(z(\tau_k, \tau_k), \ldots, z^{(l)}(\tau_k, \tau_k)) & = & u_{ref}(\tau_k, \tau_{k-1}) \\ \varphi_2(z(t, \tau_k), \ldots, z^{(l)}(t, \tau_k)) & \in & \mathcal{U} \\ d_{O_m}(t, \tau_k) & \geq & \rho + r_m, \quad \forall O_m \in \mathcal{O}(\tau_k) \end{cases} \tag{3.1.4}$$

The period $[\tau_{-1}, \tau_0)$ is what is called by Defoort "the initialization phase" which considers:

$$q_{ref}(\tau_0, \tau_{-1}) = q_{initial}$$

$$u_{ref}(\tau_0, \tau_{-1}) = u_{initial}$$

without no more further changes to the expressions above.

Practical stuff for implementation $q \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. N_s number of time steps used when computing the problem.

Number of equations: $n + m$

Number of inequations (function of τ_k): $N_s(m + \text{card}(\mathcal{O}(\tau_k)))$

dependencies: sudo apt-get install python python-dev libatlas-base-dev gcc gfortran g++

get source: https://pypi.python.org/pypi/scipy

sudo python setup.py install

3.2 The mobile robot

For representing the mobile robot geometry in the planning plane a bounding circle was chosen.

3.2.1 Unicycle kinetic model

3.2.2 Flat output formulation

3.3 The obstacles

Two different representations of an obstacle are supported. Obstacles can be seen as circles or convex polygons.

Representing an obstacle as a circle is probably the most simple way of doing so and has great advantages when calculating point-to-obstacle distance compared to other representations.

Nevertheless, obstacles such as walls, boxes and shelves cannot be satisfactorily represented by circles. Thus the need of a polygon representation.

3.3.1 Robot-to-obstacle distance calculation for the convex polygon representation

As sad before the robot's geometric form is represented by a circle. When calculating the robot-to-obstacle distance this simplified representation is quite useful. The first approach to calculate the distance between a point and an obstacle represented by a convex polygon was to separate the problem in three cases with a different expression for the distance computation each. We see in the Figure 3.1 that the points A, B and C are placed in three different regions with respect to the obstacle. A is "between" the two lines ($r_{0,1}$ and $r_{0,3}$) that pass through the vertex 0 and are orthogonal to the the two adjacent edges. B is "between" the edge s_3, and the orthogonal lines $r_{0,3}$ and $r_{3,2}$. C is in the interior of the obstacle representation, i.e., surrounded by the four edges.

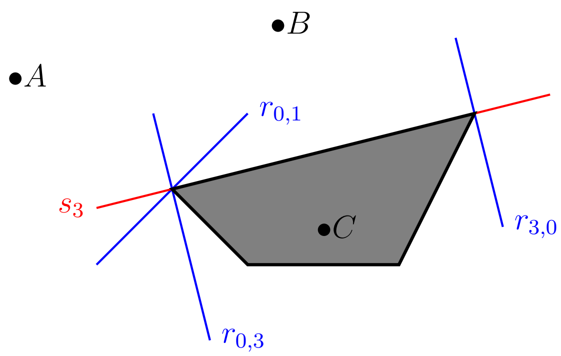


Figure 3.1 – Voronoi regions used for case differentiation.

These cases make use of Voronoi regions [4]. Based on the three types of regions (here we consider the interior of the polygon as a third one) a case differentiation is made and, depending on the case, equations are solved.

It is easy to see that the computation of the point-to-obstacle distance for A is a simple point-to-point distance using the appropriate vertex. For B a point-to-line distance equation can be used. Finally, since C is in the interior of the polygon the penetration distance is calculated. It is considered as the shortest of the four distances from the point C to the four edges multiplied by -1 (so, once more, point-to-line distance).

Of course that the performance of this approach is "number of edges"-dependent and present fast results only or polygons with few edges (less than 10).

10 was arbitrary, improve this finding a meaningful value or delete it

An important remark though is that for a given planning horizon N_s point-to-obstacle distances have to be calculated. Intuitively we can say that there is a high probability that most of the N_s points are inside the same region defined by theirs relative positions to the obstacle. Besides, the probability of finding points inside regions that are "far" from the already occupied zones is smaller. This heuristic can be used to speed up the planning process by having a smarter initialization of point-to-obstacle distance computation when using a convex polygon representation.

Finally, when dealing with more complex obstacles representations and/or with a more complex representation of the mobile robot geometry the Enhanced Gilbert-Johnson-Keerthi distance algorithm [4] is a more suitable and efficient approach.

some code is available on the internet, Google code written in D language and/or the other one on stackoverflow, see bookmarks

3.4 Analysis of the parameters impact on real-time feasibility and solution adequacy

The performance and solution quality of the motion planning algorithm previously presented depends on several parameters. These parameters can be split into two groups. The **algorithm related** parameters and the **optimization solver related** ones. Among the former group, the most important ones are:

- The number of sample for time discretization (N_s);
- The number of internal knots for the B-splines curves (n_{knots});
- The planning horizon for the sliding window (T_p);
- The computation horizon (T_c).

The latter kind depends on the optimization solver adopted. However, since most of them are iterative methods, it is common to have at least the two following parameters:

- Maximum number of iterations;
- Stop condition.

This high number of parameters having influence on the solution and/or on the time for finding a solution makes the search for a satisfactory set of parameters' values a laborious task.

We attempt nevertheless to extract some quantitative knowledge about how these parameters impact the generated solution and its computation time based on several simulations run with different parameters configurations. The main objective here is to be able to support the feasibility of a real-time motion planner based on this algorithm.

After these analyses we shall be able to identify sets of parameters' values that minimize the total time spend to complete the mission, respecting the problem constraints and minimizing the *maximum computation time*/T_c ratio.

3.4.1 Computation time analysis

At first we were interest in finding how variations in N_s and N_{knots} impart the computation time for finding a solution.

Aiming for a scenario invariant understanding of the impact of these parameters three different scenarios were studied. A first scenario where the robot did not had to avoid any obstacle to complete its mission, a second one where three round obstacles were randomly generated in a region where the robot was probably going to pass through and a third similar to the second only with six instead of three obstacles.

In addition, to reduce the problem's size, a unique optimization solver with fixed parameters was used for all simulations. The used parameters can be seen in table 3.1. The subscribed words *first*, *inter* and *last* indicate that the respective maximum numbers of iteration are used for the first optimization problem solving, for all intermediaries ones and for the last one. Different maximum numbers of iterations are used for different stages of the planning process. This is possible because of two aspects of the problem addressed:

- We assume that the initial configuration at time t_0 is static and thus the planning of the first section of the plan does not have to be done within the computation horizon (T_c). We choose then to increase the maximum number of iterations for the first planning section in order to achieve better results;
- According to the change in the algorithm for solving the NPL associated with the last section of the plan we know that the last step uses a number of internal knots (N_{knots}) smaller than or equal to the number used by the other steps. In the other hand, this final NPL has more constraints than the others NPLs solved before for a given N_s. Since the SLSPQ method requires $O(n^3)$ time to find a solution, where n, the dimension of the parameter vector, is directly proportional to the number of internal knots we assume that the number of maximum iterations for the last step can be increased.

Real-time feasibility in this context can be considered as having the *maximum computation time* spend for planning the path sections[1] less than or equal to the computation horizon (T_c). Here though we are only interest in understanding the variation of *maximum*

1. All computational times spend for planning all sections are considered for finding the maximum value but the first one

Table 3.1 – Optimization solver parameters

Optimization solver type	SLSQP
$MAXIT_{first}$	40
$MAXIT_{inter}$	15
$MAXIT_{last}$	20
accuracy	10^{-3}

computation time/T_c with changes in N_s and N_{knots}.

No obstacles scenario

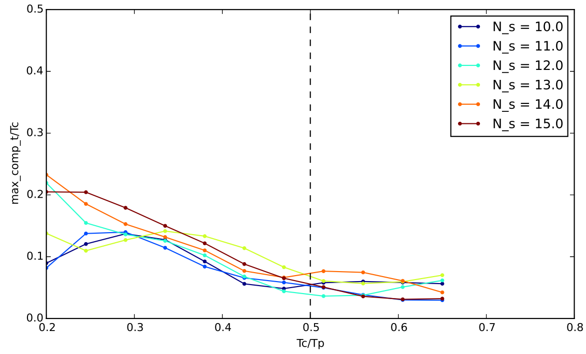
The images in the Figure 3.2 try to make prominent the effect of changes in the in the number of samples (N_s) and number of internal knots (N_{knots}). In the ordinate axis we have the *maximum computation time*/T_c ratio and in the abscissa we have the T_c/T_p. For each N_s we took the average of the *maximum computation time*/T_c ratio for a give T_c/T_p among different T_p values in order to be T_p invariant.

We can see that for a "no obstacles" scenario the overall performance with respect to the *maximum computation time*/T_c ratio is only slightly impacted by variations in the number of samples (N_s) and in the number of internal knots (N_{knots}). Within a given image the lines are close together showing that variations in N_s have weak impact. In addition, comparing the three images (3.2a, 3.2b, 3.2c) we see that variations in the N_{knots} have also a weak influence for this scenario.

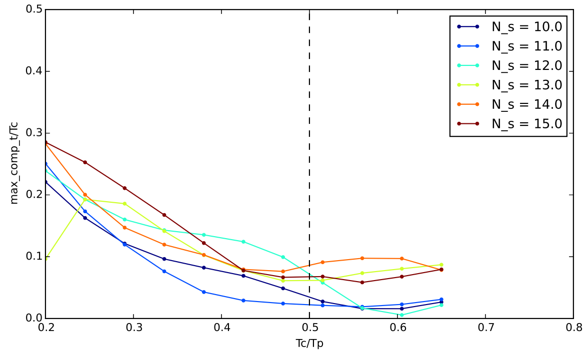
For the sake of an example we present a simulation result in the Figure 3.3 run with the parameters presented in the table 3.2 for a "no obstacles" scenario.

Table 3.2 – Motion planner main parameters

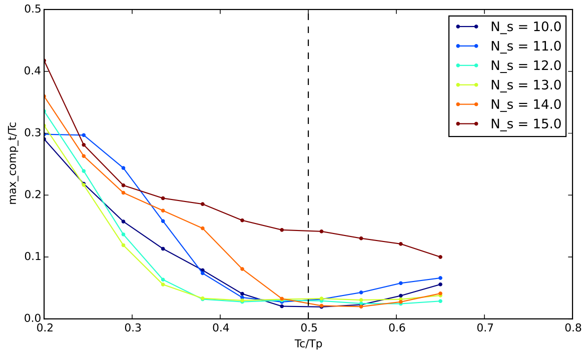
T_p	2.00 s
T_c	0.40 s
N_s	9
N_{knots}	5
v_{max}	1.00 m/s
ω_{max}	5.00 rad/s
q_{inital}	$[-0.05\ 0.00\ \pi/2]^T$
q_{final}	$[0.10\ 7.00\ \pi/2]^T$
u_{final}	$[0.00\ 0.00]^T$
u_{final}	$[0.00\ 0.00]^T$



(a) Four internal knots. Average variance between lines is 0.062×10^{-2}

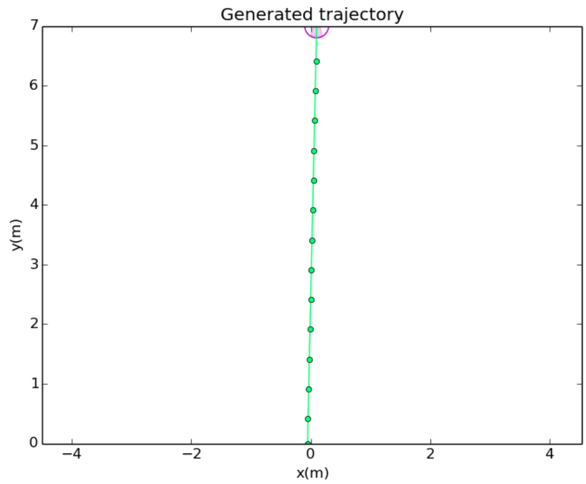


(b) Five internal knots. Average variance between lines is 0.115×10^{-2}

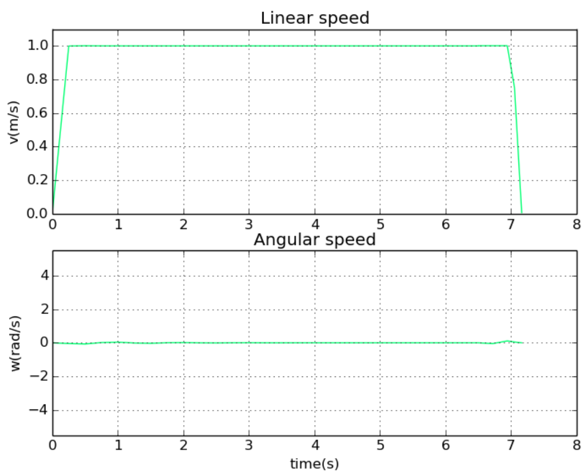


(c) Six internal knots. Average variance between lines is 0.182×10^{-2}

Figure 3.2 – Zero obstacles scenario.



(a) Robot's path.

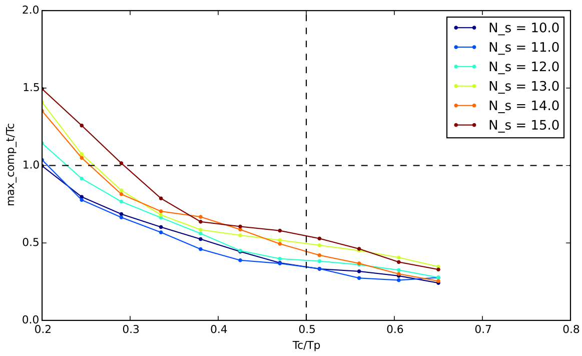


(b) Robot's input.

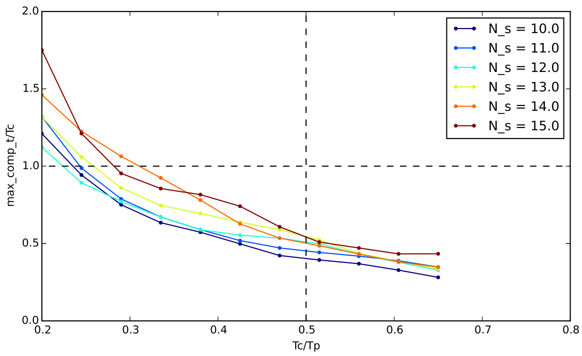
Figure 3.3 – No obstacle scenario simulation example where the *maximum computation time* was about 78% of T_c and the mission total time equals to 7.16 s.

Three obstacles scenario

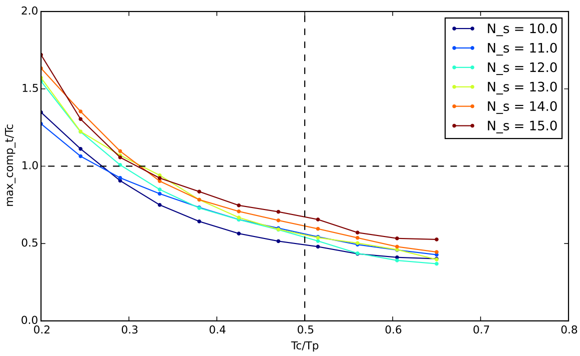
For this new scenario, a greater impact of the number of samples (N_s) and number of non-null internal knots (N_{knots}) is observed. The greater the N_{knots} or the N_s the greater is the *maximum computation time*/T_c. This behavior is the one expected since the number of constraints and the number of arguments for the cost function to be minimized depend on these two parameters respectively.



(a) Four internal knots. Average variance between lines is 1.047×10^{-2}



(b) Five internal knots. Average variance between lines is 0.972×10^{-2}



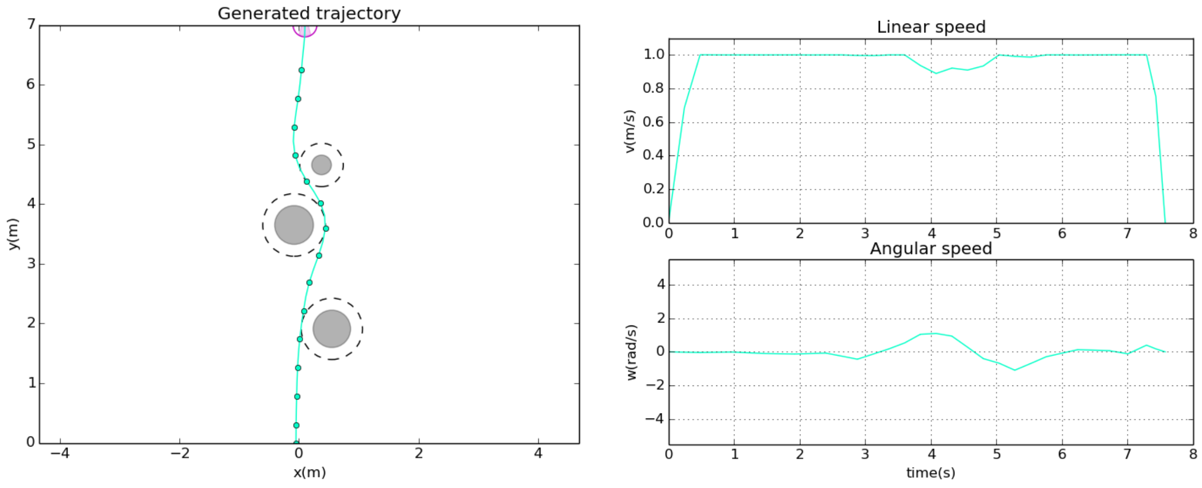
(c) Six internal knots. Average variance between lines is 0.587×10^{-2}

Figure 3.4 – Three obstacles scenario.

Again, in the Figure 3.3 we show a simulation example run with the parameters' values presented in table 3.4.

Table 3.3 – Motion planner main parameters

T_p	2.40 s
T_c	0.48 s
N_s	11
N_{knots}	4
v_{max}	1.00 m/s
ω_{max}	5.00 rad/s
q_{inital}	$[-0.05\ 0.00\ \pi/2]^T$
q_{final}	$[0.10\ 7.00\ \pi/2]^T$
u_{final}	$[0.00\ 0.00]^T$
u_{final}	$[0.00\ 0.00]^T$
O_0	[0.55 1.91 0.31]
O_1	[−0.08 3.65 0.32]
O_2	[0.38 4.65 0.16]

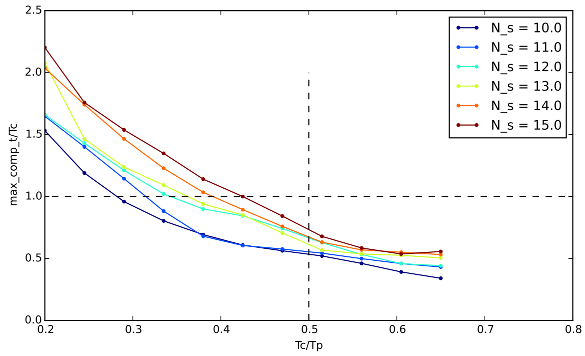


(a) Robot's path. (b) Robot's input.

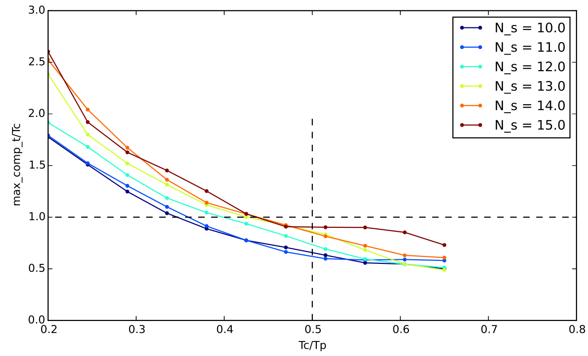
Figure 3.5 – Three obstacle scenario simulation example where the *maximum computation time* was about 84% of T_c and the mission total time equals to 7.57 s.

Six obstacles scenario

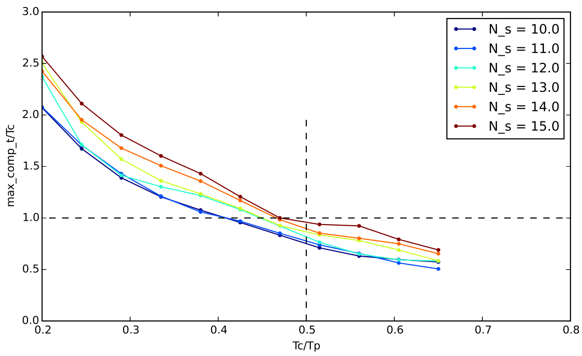
As for the scenario with six obstacles we realize that the observations for the latest scenario are accentuated.



(a) Four internal knots. Average variance between lines is 2.272×10^{-2}



(b) Five internal knots. Average variance between lines is 2.635×10^{-2}



(c) Six internal knots. Average variance between lines is 1.526×10^{-2}

Figure 3.6 – Six obstacles scenario.

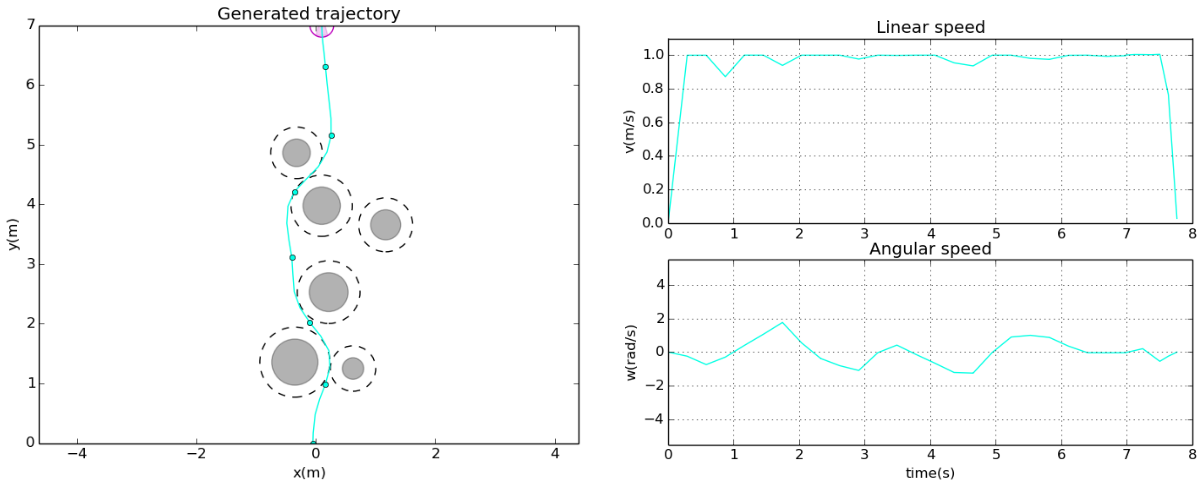
add compt time X max number of detected obsts

A deeper analysis showed us that the SLSQP solver requires $O(n^3)$ time for finding the NPL solution, n being the parameters dimension which is directly proportional to N_{knots}. However the solver require $O(N)$ time, N being the constraints dimension which is directly proportional to N_s.

Worth noticing though that the complexy orders just presented are specific for this optimization solver. Other solvers presenting different stategies for solving the NPL will behave differently. In addition, the number of internal knots tends to be quite small (less than 10) for usual scenarios (TODO: specify what is a usual scenario, mobile robot linear speed, concentration of obstacles). Finally, increasing N_{knots} indefitalely does not mean an improvement of the solution adequacy (TODO: develop why, saying that more control points than needed does help). While an increasing in the sampling number N_s

Table 3.4 – Motion planner main parameters

T_p	3.20 s
T_c	1.28 s
N_s	12
N_{knots}	6
v_{max}	1.00 m/s
ω_{max}	5.00 rad/s
q_{inital}	$[-0.05\ 0.00\ \pi/2]^T$
q_{final}	$[0.10\ 7.00\ \pi/2]^T$
u_{final}	$[0.00\ 0.00]^T$
u_{final}	$[0.00\ 0.00]^T$
O_0	$[-0.35\ 1.36\ 0.39]$
O_1	$[0.21\ 2.53\ 0.33]$
O_2	$[-0.32\ 4.86\ 0.23]$
O_3	$[0.10\ 3.98\ 0.31]$
O_4	$[0.62\ 1.25\ 0.18]$
O_5	$[1.17\ 3.66\ 0.25]$



(a) Robot's path.

(b) Robot's input.

Figure 3.7 – Six obstacle scenario simulation example where the *maximum computation time* was about 90% of T_c and the mission total time equals to 7.76 s.

always improve solution adequacy with respect to obstacle penetration as shown in the next section.

3.4.2 Detection radius impact

As the detection radius of the robot increases more obstacles are seen at a time which linearly increases the number of constraints in the NPL. Figure 3.8 gives a example of how the ratio varies as the detection radius progressively increases. The numbers that can be seen over the blue curve are the maximum number of obstacles seen at once during the whole mission.

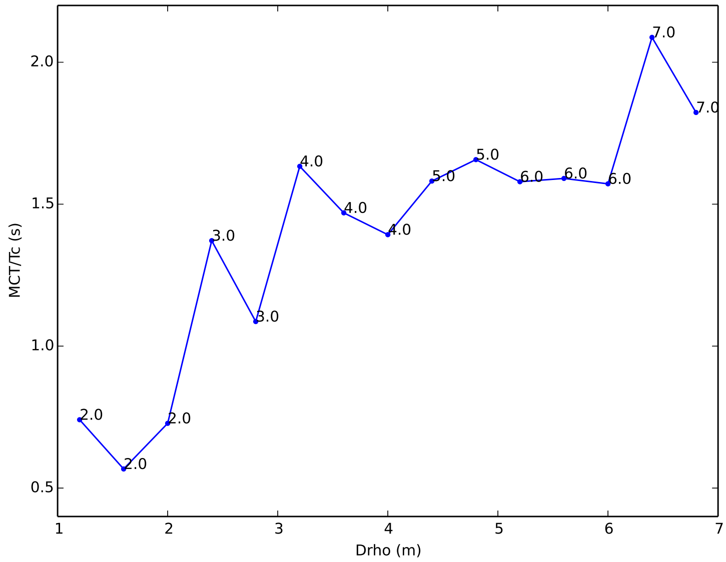


Figure 3.8 – MCT/T_c ratio varying with ρ_d.

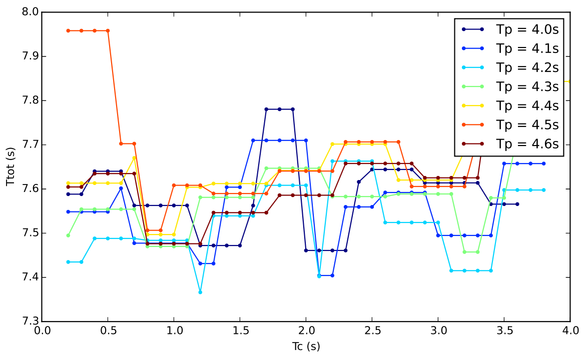
3.4.3 Solution adequacy

The time spend for finding the solution for a given set of parameters values does not impact the solution itself. Let's analyze then how the solution behaves according to some parameters.

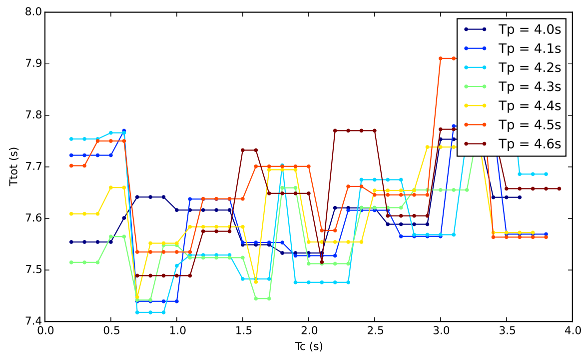
Two important notions when trying to quantify the adequacy of the solution are the total time spend for completing the mission and some metric about how much the planned path avoids obstacles.

Total mission time

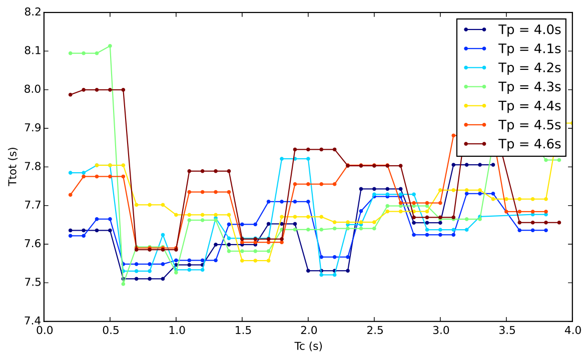
Figure 3.9 show how the time spend for completing the mission behaves with respect to three parameters: T_c, T_p, N_s.



(a) $N_s = 10$



(b) $N_s = 11$



(c) $N_s = 12$

Figure 3.9 – Variation of total mission time with the computation horizon (T_c) for differet planning horizons (T_p) and N_s and three obstacles.

We observe an overall tendency that for a given number of internal knots and N_s the total mission decreases as the planning horizon decreases. This can be explained by the fact that the trajectory quality (optimality) is degradated as the density of internal knots N_{knots} within a T_p decreases too much.

One may notice as well that the total mission time is invariant with respect to T_c in the sense that no pattern can be observed besides oscillations of the total time due to the scenario specific configuration.

Another relevant observation is that the overall time for completing the mission decreases as the sampling number N_s decreases. This misleading improvement in the solution adequacy hides the fact that the fewer the samples the greater will be the obstacle penetration as shwon later in this section.

Detection radius We can also be interest in considering how the total mission time changes as the detection radius of the robot increases. In Figure 3.10 we see the rapidly decrease of the the mission time as the ρ_d increases showing that a more optimal trajectory can be found when a better knologde of the environment is possible.

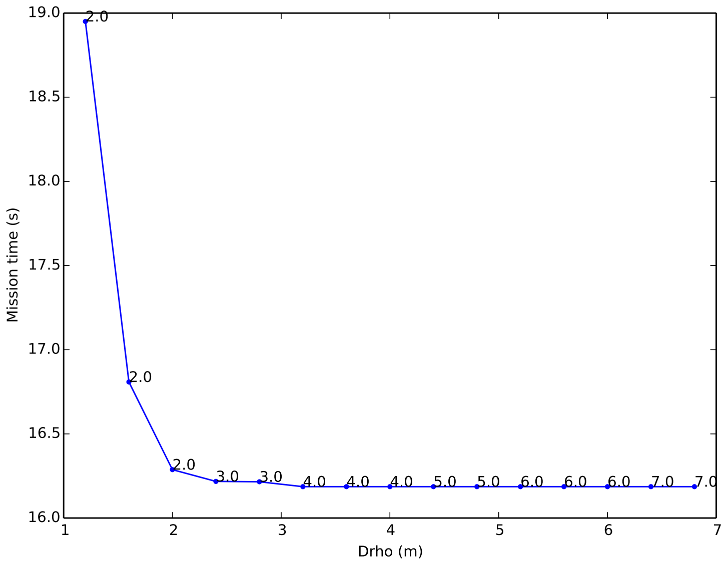


Figure 3.10 – Total mission time varying with ρ_d.

Get no obstacles information too

Obstacle penetration

Rather than using a metric for obstacle avoidance we used its conterpart called here obstacle penetration. The total obstacle penetration area (P) can be calculated as the some of all the penetration areas (P_w), each one being the penetration area found for a given planning section w. P_w for a round obstacle can be estimated as follows:

$$\mathrm{P}_w \simeq \frac{T_c}{N_{Tc}} \sum_{o=0}^{M-1} \sum_{k=0}^{N_{Tc}} \frac{r_o p_k - \frac{p_k^2}{2}}{r_o - p_k} v_k$$

where r_o is the radius of the oth obstacle, p_k is the length of the penetration and v_k is the linear velocity for a given time instant k within the computing horizon w.

As expected, the greater the N_{ssol} (which in turn increases N_{Tc}) the more accurate is the area found. We show in Figure 3.11 that the area value converges to the precise area as N_{ssol} increases.

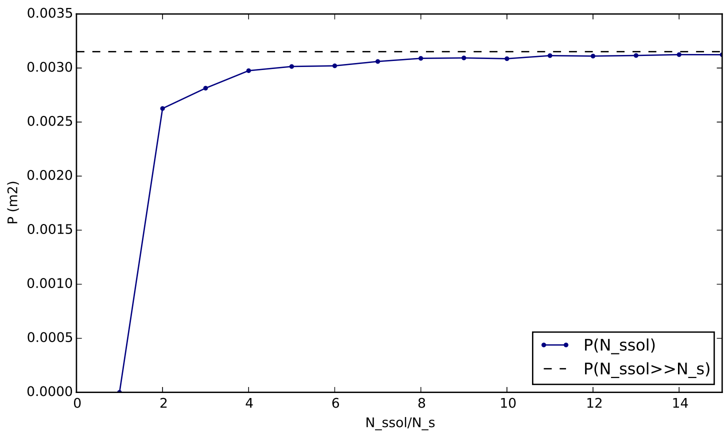


Figure 3.11 – Convergence of the area computation.

After some tests we saw that a good estimate for the penetration area can be found using $N_{ssol} > 10N_s$.

Ideally we would have $\mathrm{P}_w = 0$. A way of guarantying that would be to increase the obstacles radius computed by the robot's perception system by the maximum distance that the robot can run within the time spam T_p/N_s (equivalent to T_c/N_{T_c}). However simple, this approach represents a loss of optimality and will not be considered in this work.

Influence of N_s on penetration area

As state before in subsection or section TODO the sampling was a sensible parameter regarding the computation time. The greater the number of samples the greater the number of inquations constraints and consequently the computation time for solving the NLP. Now, analyzing the impact of the sampling on the penetration area we see that the greater the number of samples the smaller is the penetration area (as expected). In Figure 3.12 we see that the penetration total area rapidly decreases as N_s increases.

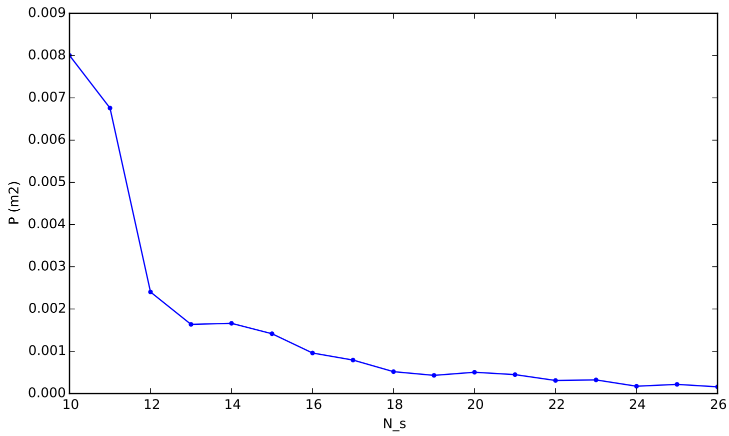


Figure 3.12 – Penetration area varying with N_s.

3.4.4 Conclusions

Valid for SLSQP-based sovler:

- MCT/T_c is $O(n^3)$ where n is directly proportional to N_{knots};
- MCT/T_c is $O(n)$ where n is directly proportional to N_s;
- MCT/T_c increases as the detection radius of the robot increases;
- T_{tot} decreases as N_s decreases (up to a minimum value);
- T_{tot} is not influenced by T_c in a observable way;
- T_{tot} decreases as the N_{knots} increases (but not indefitalely);
- P increases as N_s decreases (up to a maximum value);

"Semantic" interpretation/analysis of the obstacles can speedup NPL solving by reducing the number of contraints.

"Calibration" of the parameters can be done once knowing the real application conditions:

- Approx. obstacles "density";
- Robots max speed;
- Approx. distance;
- etc.

by simulating before and pay attention to the behavior of important values such as MCT/T, T_{tot}, P.

Bibliography

[1] John T Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics*, 21(2):193–207, 1998.

[2] M. Defoort, J. Palos, a. Kokosy, T. Floquet, W. Perruquetti, and D. Boulinguez. Experimental Motion Planning and Control for an Autonomous Nonholonomic Mobile Robot. *Proceedings 2007 IEEE International Conference on Robotics and Automation*, (April):10–14, 2007.

[3] Michael Defoort. Contributions à la planification et à la commande pour les robots mobiles coopératifs. *Ecole Centrale de Lille*, 2007.

[4] C. Ericson. *Real-Time Collision Detection*. M038/the Morgan Kaufmann Ser. in Interactive 3D Technology Series. Taylor & Francis, 2004.

[5] Dieter Fox, Wolfram Burgard, Sebastian Thrun, et al. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, 1997.

[6] a. Kelly and B. Nagy. Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control. *The International Journal of Robotics Research*, 22(7-8):583–601, 2003.

[7] Jean-Claude Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.

[8] Mark B Milam. *Real-time optimal trajectory generation for constrained dynamical systems*. PhD thesis, California Institute of Technology, 2003.

[9] Jorge Nocedal and Steve J. Wright. *Numerical optimization : with 85 illustrations*. Springer series in operations research. Springer, New York, Berlin, Heidelberg, 1999. Tirage corrigé: 2000.

[10] Ruben E. Perez, Peter W. Jansen, and Joaquim R. R. A. Martins. pyOpt: A Python-based object-oriented framework for nonlinear constrained optimization. *Structures and Multidisciplinary Optimization*, 45(1):101–118, 2012.

[11] Jacob T. Schwartz and Micha Sharir. A survey of motion planning and related geometric algorithms. *Artificial Intelligence*, 37(1):157–169, 1988.

[12] Workshop. Workshop on on-line decision-making in multi-robot coordination. http://robotics.fel.cvut.cz/demur15/, 2015. Accessed: 2015-06-22.

Appendix A

Random Graphs

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.