Probabilistic Optimisation applied to Spacecraft Rendezvous on Keplerian Orbits

Grégory Saive\textsuperscript{a}, Massimiliano Vasile\textsuperscript{b}
\textsuperscript{a}Université de Liège, Faculté des Sciences Appliquées, Belgium
\textsuperscript{b}Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano, Italy

Abstract
A probabilistic approach for optimal path planning has been investigated in order to design trajectories for the terminal phase and the rendezvous phase of docking manoeuvres on a general Keplerian orbit. This approach is based on a probabilistic search for the optimal path connecting two points in space. This is also known as randomised A* tree expansion. Prior to the development of the probabilistic algorithm, the solution to the equations of relative motion for general Keplerian orbits has been investigated. Finally a potential improvement of the probabilistic search for an optimal path using interval algebra for pruning unpromising branches will be presented.

Introduction
A docking and rendezvous manoeuvre has three distinct phases: far rendezvous phase, intermediate rendezvous phase, and terminal rendezvous phase. The first one includes the launch and one or several phasing orbits to get close to the target. If this part is quite well mastered for circular orbit, it is not the case on elliptical or hyperbolical orbit. Technologies like tracking, communications, trajectories, etc… are not available yet. The second phase consists of moving the spacecraft from 20 km of the target to 1km. When the spacecraft is as far as 1km of the target, the terminal rendezvous begins. The spacecraft cannot be seen as point anymore and the attitude has to be taken into account. For this phase, the usual physical laws have to be transformed to describe the relative motion \cite{2},\cite{3}, between two spacecrafts and not the absolute motion anymore. To solve the problem of motion planning, quite a lot of techniques \cite{1},\cite{5},\cite{6}, exist and can be used in the case of relative motion. The construction of a trajectory may be designed thanks to a dynamic tree approach. One first seeds the search space with a start and goal configuration, then new waypoints are created and edged to old one. Bad edges and bad waypoints are discarded and optimal way is harvest from the tree. The key to finding optimal path quickly is in choosing how and when to branch. A good exploration of the space to be covered can be done by using probabilistic choose. The proof of the completeness of the probabilistic process has been done before and it can be found under the name of the paradigm of Probabilistic Roadmap. The extension of the tree is done by choosing randomly from the tree a waypoint (based on weighted distribution where the weight is function of a heuristic cost), and then applying an impulsive control to this point. The resulting point is then branched to the final point. This method is called the randomised A* tree expansion and in this work it will be applied to the docking phase and preliminary phase. This is an extension to all kind of Keplerian orbit of an algorithm that can be found in literature \cite{1}. Here it is used to optimise the trajectory during the docking even
when attitude control has to be taken into account. Finally some attempts to improve the algorithm by using interval algebra are presented.

Terminal rendezvous equations

The equations of motions of two spacecraft in a central force field are not linear, nevertheless, for proximity operations, these equations can be linearised in neighbourhood of the orbit of one of the two spacecraft (commonly the so called target). In particular if the target is moving on a quasi-circular orbit the linearised equations of motions can be found under the name of Hill’s equations or Clohessy-Wiltshire’s equations. On the other hand if the target satellite moves on an arbitrary eccentric orbit the proximity motion is described by the Tschauner-Hempel’s equations. It can be easily shown that Tschauner-Hempel equation reduce to Clohessy-Wiltshire equations as their respective solutions if the eccentricity of the reference orbit tends towards zero. The work presented here is based on the development of Humi and Carter [2],[3] of the Tshauner-Hempel equations, expressed in orthogonal frame bound to the target:

\[
y''_1 = 2y'_2
\]

\[
y''_2 - \frac{3}{1 + e \cos \theta} y_2 = -2y'_1
\]

\[
y''_3 + y_3 = 0
\]

where primes denote differentiation with respect to \( \theta \), the true anomaly of the reference point. The initial variables can be found back thanks to the relations:

\[
y_i = \sqrt{\omega} x_i, \quad i = 1,2,3
\]

Equations (1) have been obtained under some simplifying assumptions: first of all, the gravitational field is homogenous and produced by a single point mass, then the chaser is moving in a free environment and the target is just a point, lastly, attitude control and trajectory control are completely uncoupled. Notice that, although the Tschauner-Hempel equations reduce to the Clohessy-Wiltshire if one makes tend the eccentricity towards zero, the Clohessy-Wiltshire equations are expressed in terms of the time. This makes them easier to manipulate. For that reason, if the eccentricity is equal to zero, the algorithm uses these ones instead of the Tshauner-Hempel ones. Writing the solutions under the state transition matrix form, it allows an easier implementation under any programming languages. The solution is well known and can take the form of the state transition matrix

\[
y(\theta) = \Phi(\theta) \Phi(\theta_0)^{-1} \cdot y(\theta_0)
\]

where \( \Phi \) is the solution matrix of Equation (3). First of all, we will use the following convention to denote a definite integral of any function

\[
S(f(\theta)) = \int_{\theta_0}^{\theta} f(\vartheta) d\vartheta
\]

The solution matrix is then
\[
\Phi = \\
\begin{bmatrix}
2S(f_1) & 2S(f_2) & S(2f_3 + 1) & 1 & 0 & 0 \\
f_1 & f_2 & f_3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \cos(\theta) & \sin(\theta) \\
2f_1 & 2f_2 & 2f_3 + 1 & 0 & 0 & 0 \\
f'_1 & f'_2 & f'_3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\sin(\theta) & \cos(\theta)
\end{bmatrix}
\]

(5)

and its inverse
\[
\Phi^{-1} = \\
\begin{bmatrix}
0 & 4S(f_2) + f'_2 & 0 & -2S(f_2) & -f_2 & 0 \\
0 & -(4S(f_1) + f'_1) & 0 & 2S(f_1) & f'_1 & 0 \\
0 & -2 & 0 & 1 & 0 & 0 \\
1 & 2S(2f_3 + 1) + f'_3 & 0 & -S(2f_3 + 1) & -f_3 & 0 \\
0 & 0 & \cos(\theta) & 0 & 0 & -\sin(\theta) \\
0 & 0 & \sin(\theta) & 0 & 0 & \cos(\theta)
\end{bmatrix}
\]

(6)

where
\[f_1 = \rho(\theta)\sin(\theta)\]
\[f_2 = -6e^2 f_1(\theta)K(\theta) + \frac{2e \sin^2(\theta)}{\rho(\theta)} - \frac{\cos(\theta)}{\rho(\theta)}\]
\[f_3 = 6ef_1(\theta)K(\theta) - \frac{2\sin^2(\theta)}{\rho(\theta)^2} - \frac{\cos^2(\theta)}{\rho(\theta)} - \cos^2(\theta)\]

(7)

with
\[\rho(\theta) = 1 + e \cos(\theta)\]
\[K(\theta) = \int \frac{\sin^2(\vartheta)}{\rho(\vartheta)^4} d\vartheta\]

(8)

All mathematical singularities have been removed from these solutions, written in terms of \(K\). Thanks to this set of equations, relative motion is defined for every kind of input. By using adequate variables change, the function \(K(\theta)\) can be evaluated for the whole range of eccentricity \(e\).

**The BOB manoeuvre for a general Keplerian orbit**

The simplest manoeuvre to perform a docking is the bang-off-bang (BOB) manoeuvre, which consists of two impulsive controls, the first at beginning of the manoeuvre to reach the target and the second one at the end in order to cancel the residual relative velocity. The BOB manoeuvre cannot be used for real docking manoeuvre since the resulting residual velocity at the target spacecraft is often quite significant with a consequent high risk of destructive impacts both for the chaser and the target and high demand on control authority of the engines. Besides this, if any obstacles appear during the transfer, no countermeasure could be applied to palliate to this problem. On the other hand, the BOB manoeuvre can be used as an indicator of the cost in \(dv\) needed for the manoeuvre.
Therefore an analysis of BOB manoeuvre for general Keplerian orbits, will provides us with useful information to reduce the cost of a real multiple-burn docking mission. By inspecting the Clohessy-Wiltshire and the Tschauner-Hempel equations, one can see that the critical inputs are the time in which the manoeuvre is performed and the position on the orbit where it takes place (that means the true anomaly).

For an ellipse, Figure 1 shows that the BOB manoeuvre is optimal when performed at the apocentre of the target orbit. This poses an additional issue with respect to a similar manoeuvre on a circular orbit: if the encounter is missed or impossible at the prescribed apocentre pass, the chaser spacecraft has either to wait for the next transit by the apoapsis (and the wait time can be very long for high eccentric orbit) or resolve itself to perform the manoeuvre in a less efficient way.

Two regions have been cut off from the Figure 1. These two regions are physical singularities and have been removed from the graphic to allow a good comprehension of the behaviour of the equations. These two singularities appears for a time of transfer equal to zero and equal to the period of the orbit. The reason of the first one is obvious. It is impossible to perform any manoeuvre in an infinitely short time. The second is more due to the model. Actually, all the trusts are impulsive. However, it can be shown that an impulsive control changes the orbit of a spacecraft in a new one that crosses the old one exactly on the point of the impulse. Then, it is impossible, after one revolution, for the spacecraft, to be, in the same time close to the target and at the same position than at the beginning of the manoeuvre.

The difficulty to perform a rendezvous operation on a hyperbolic orbit (Figure 2) is glaring. If one misses the opportunity or the target, it will never come back. Furthermore, the time in which to perform the docking is crucial because, otherwise, one hazards not to be able to reach a periodic orbit again. Nevertheless, it also exists optimal set of parameters to reduce the cost. This region is located before the pericentre. In an actual case of docking, it can be really difficult to reach the hyperbola from an ellipse before the pericentre. The point where the manoeuvre should begin would be at the pericentre or after, in the less efficient area.

Figure 1: Total Δv for a BOB manoeuvre on a 0.25 eccentric elliptical orbit
The randomised $A^*$ tree expansion algorithm

The algorithm developed here is based on the approach for robotic path planning and docking manoeuvre presented in [3] and has been extended to treat any Keplerian reference orbit and to solve also the phasing problem from and towards any conic trajectory. The optimisation is based on a tree expansion. The selection of a point in the tree to be extended is done randomly thanks to weighted distribution while the choice of the impulsive controls follows a normal distribution in ranges formerly selected. This weight is calculated as follows:

$$\text{weight} = \frac{f(\text{order})}{f(A^* \_\text{cost}) \cdot f(\text{out \_degree})}$$  \hspace{1cm} (9)$$

$\text{Order}$ represents the kinship of the point with the starting point. The value of order for the starting point is one; the value for direct children points is two and the value for children of these points is three, and so on. By giving more weight to recent nodes, we force the tree to grow preferentially from these points. $\text{Out \_degree}$ represents the number of times the waypoint has been expanded. By this way, the number of times the same point is chosen is limited and we avoid the tree to be stuck in local minima. The $A^* \_\text{cost}$ is a prediction of the overall cost. It is the sum of the cost already calculated and the estimated cost. The estimated cost is calculated thanks to a heuristic function. Heuristic function is widely used in graph searching, especially in an algorithm called $A^*$, the reason of the name of this argument. In our case, this heuristic function is simply the BOB manoeuvre from the point to the goal. It serves to focus the search to waypoints that present a good chance to lead to a low energy path. The functions used for these arguments were integer power because of the simplicity of implementation. Anyway different functions might be used depending on the problem under study.

Terminal phase and docking problem

A typical tree obtained thanks to the algorithm can be seen on Figure 3. It represents a tree of 10000 points and the minimum path found into it is 1.43 times
the BOB manoeuvre. The mean time to obtain a tree is 15 seconds. The equations used to expand the tree are the Clohessy-Wiltshire equations in the circular case and the Tshauner-Hempel in any other case. All the controls are impulsive for the thrust and the attitude control. The rocket equation is used to total cost of the combined attitude and trajectory manoeuvre. The weight function useful in the point selection is in this case:

\[
weight = \frac{\text{(order)}^2}{(A*\text{cost})^3\text{(out\_degree)}^2}
\]  

Figure 3: Tree of possible paths for a circular reference orbit

A population of 1500 trees of 10000 waypoints each has been calculated. For each of these trees, the minimum path has been stored. The mean value for all the paths discovered by the algorithm is 2.2 times the BOB manoeuvre while the minimum path is 1.3 times the BOB manoeuvre.

In the case of elliptical and hyperbolical orbits the mean time to produce a tree of 10000 points was 150s. This time was mostly spent during the calculation needed to change from the true anomaly variable to the time variable, as this computation has to be done for each point. The weight function for elliptical orbit of eccentricity lower than 0.4 is the same as Equation (10). For higher eccentricity (highly eccentric elliptical orbit and hyperbolic orbit), the weight function had to be changed.

\[
weight = \frac{(order)^3}{(A*\text{cost})^3\text{(out\_degree)}^3}
\]  

The modification of Equation (10) was necessary to avoid the tree being trapped by in local minima. In fact increasing the power of order and out\_degree forces the tree to look further the space. Unfortunately, this means that the heuristic function is less dominant and then a larger number of iterations is needed to find an optimal path. The tree in Figure 4 has been computed with an elliptical orbit as orbital reference. The manoeuvre is started around the apocentre as suggested by the
analysis in the section above. The shape of the tree and the fact that it is wider than for circular orbit is mainly due to the change of the weight function.

Figure 4: Tree of possible paths for elliptical reference orbit (e = 0.71)

A population of 100 trees have been generated for an elliptical reference orbit and a hyperbolic reference orbit. The minimum value found for the elliptical reference orbit is 1.89 times the BOB estimate and the average value is 2.27. In the case of the hyperbolical orbit, the minimum value registered is 2.67 times the BOB manoeuvre and the mean is 3.64. Then a comparison of the influence of the eccentricity on the randomised A* tree expansion algorithm has been performed.

Six different eccentricities from almost 0 to 2.8 has been tested with as input parameters a relative position of 500 m along x₁ axis and x₂ axis, 0 for the x₃ axis and relative velocity equal to zero in each direction. Each time, the manoeuvre begins at the pericentre for a total time of simulation equal to 3000s. For each eccentricity, 50 trees of 7500 waypoints each have been grown. An increasing in both the total dv of the path and in the ratio is observed. The difference in the total $\Delta v$ cost and the BOB manoeuvre between the hyperbolical and the elliptical orbits is apparently due to the fact that all the manoeuvre have begun at the pericentre, a less efficient region in the elliptical case.

<table>
<thead>
<tr>
<th>Eccentricity</th>
<th>Elliptical orbits</th>
<th>Hyperbolical orbits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0036 0.5931 0.7443</td>
<td>1.4692 2.0557 2.8768</td>
</tr>
<tr>
<td>Mean total dv</td>
<td>3.8546 4.0129 4.1009</td>
<td>1.6547 1.7168 1.8593</td>
</tr>
<tr>
<td>$\Delta v$ for a BOB manoeuvre</td>
<td>1.5429 1.5843 1.6122</td>
<td>0.6097 0.6290 0.6542</td>
</tr>
<tr>
<td>Ratio</td>
<td>2.4983 2.5329 2.5436</td>
<td>2.7140 2.7295 2.8420</td>
</tr>
</tbody>
</table>

Table 1: Comparisons of results as a function of the eccentricity

**Phasing problem**

The principle of probabilistic optimization has also been applied to the phasing problem. The tree is seeded with the initial and the final position and velocity of the chasing spacecraft. For this problem, the heuristic function is calculated by solving
the Lambert's problem between the final point of the branches and the target's position for a coast time equal to the remaining time allowed to complete the mission. Thus, the reference two-impulse transfer is simply the Lambert's transfer performed in the time allowed for the mission between the two point defined as the target and the initial position of the chaser. In this way, the initial orbit and the orbit of the target are completely defined in the inertial reference frame centred in the Earth. Empirically, the weight function giving the best results was found to be:

\[
weight = \frac{\text{order}^2}{(A^4 \cos\theta^\circ \text{out} \_\text{degree}^2)}
\]  

(12)

In Figures 5 and 6, we can see some graphical results obtained with the algorithm.

Figure 5: Phasing trajectory from 2000km altitude circular orbit to a 8000km altitude circular orbit (180deg transfer)

Figure 6: Phasing trajectory from a circular orbit to an elliptic orbit (initial orbital parameters : \(a=6878\text{km}, e=0, i=23^\circ, \Omega=0^\circ, \omega=0^\circ, \theta=180^\circ\); final orbital parameters : \(a=54378\text{km}, e=0.6, i=45^\circ, \Omega=20^\circ, \omega=10^\circ, \theta=0^\circ\))
A set of 50 trees of 10000 points each have been grown for several configurations of the two satellites. For example, for the case drawn in Figure 6, the two-impulse Lambert transfer costs 5.3km/s while the probabilistic search, on average, was able to find trajectories 71.8% less expensive (1.4km/s) than the two-impulse transfer and the best solution was 22.3%(1.18km/s) of the reference ∆v. For the case on Figure 6 the reference two-impulse transfer costs 8.8km/s while the randomised search algorithm found trajectories of 69.3%(6.1km/s) of the reference Lambert transfer, on average, and a minimum solution with a cost of 64.9%(5.7km/s) of the two-impulse one.

**Pruning the tree using interval algebra**

In principle interval algebra could be used to prune the tree in order to reduce the number of generated branches and to focus the exploration just in promising areas. In this work two different attempts have been made to enhance the algorithm using interval algebra. The first one optimises the tree expansion by using a global optimisation algorithm [4] based on interval algebra. The second one pre-processes the search in order to reduce the space in which the tree has to be grown. The first approach in spite of the improvement of the results leads to a long computational time necessary to produce a tree. Therefore this approach could become advantageous only if performed on interval hardware computer.

The second idea is to perform all the calculations at each step of the randomised A* tree expansion algorithm with the interval algebra, generating, in this way, a tree of intervals. The resulting interval path was used as a corridor bounding the expansion of the tree generated with the classical algorithm: the boundaries found with to interval analysis are implemented as fictive obstacles; then, for each calculated point, a collision test is performed, if the branch crosses the boundary, the branch is pruned and a new one is generated.

![Figure 7: Interval tree of possible docking path on a circular orbit](image)

Anyway, due to the dependency problem, typical of interval analysis, the described dynamic expansion after a few iterations generates an interval tree that covers a large part of the space as it can be seen on Figure 7. The result, after few iterations,
is an overestimation of the values of each node and the pruning is not effective anymore. The solution could be to use a static decomposition of the domain using intervals instead of a dynamic expansion of the tree; this last approach is still under study and will not be presented here.

**Final Remarks**

In this paper the design of rendezvous and docking manoeuvres on general keplerian orbits has been performed using a probabilistic approach. The obtained solutions demonstrate the effectiveness of the proposed approach when applied to the global search of optimal path both for the terminal phase and for the phasing problem. It has been shown that a trajectory can be find whatever initial conditions are. For both applications, the solutions found by the algorithm can be used to find a first guess to more classical optimisation program. A further improvement of the tree expansion has been carried out using interval algebra to prune the tree. A first full global interval optimisation was performed leading to good results but with a consequent excessive computing time. Then and interval tree was dynamically generated as a pre-definition of an optimal corridor for the expansion of a normal tree. In this case the dependency problem causes after few iterations an overestimation of the bounding interval with a consequent reduction in effectiveness. A third approach with a static partitioning of the space is still under study. The results obtained with the randomised A* tree expansion seem promising, and, at present, a more realistic case, taking into account the avoidance of the target and the movement of the docking port, is under study.

**Acknowledgments**

The authors would like to thank Pr. Swings from the University of Liege for making benefit of his relationship. They are also grateful for the help of the Advanced Concept Team from ESTEC

**References**


