POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Uncertainty-Aware Motion Planning for a Robotic Manipulator in Vineyard Environments

Laurea Magistrale in Automation and Control Engineering
Ingegneria dell'Automazione

Author: Mattia Canclini

Advisor: Prof. Luca Bascetta

Co-advisor: Başak Sakçak

Academic year: 2021-2022

1. Introduction

In recent years the development of automatic harvesting methods has become crucial, particularly in Europe, to compensate scarcity of manual harvesting, due to the lack in seasonal workforce and the increase in the farm labour cost.
Our work focuses on table grape harvesting, a key sector for the Italian agriculture. Contrary to wine grapes, table grapes cannot be dented, this raises the need for robotic solutions to perform grape harvesting guaranteeing the integrity of the bunch and of the plant throughout the entire operation; hence the idea of developing an uncertainty aware motion planning framework for a mobile manipulator.
This thesis extends the work of Zandegiacomo [16], who introduced the concept of uncertainty-aware planning in simple environments. Building upon this foundation, we explored the integration of uncertainty-aware planning in more complex environments, and investigated the benefits and limitations of this approach.
To feed the planner with a complete model of the environment in the surroundings of the target it has been decided to adopt Next Best View, an algorithm that scans the environment iteratively and selects the most relevant view. The amount of information provided by a view is estimated using a novel definition of Information Gain, which considers the distance between the target and the camera view and the potential unknown volume that the camera can discover. These two components, in combination, allow the planner to adapt easily to a variety of situations, empowering the system with a high level of autonomy.
The framework is then tested on a *Franka Emika Panda*[3] manipulator.

2. State of the Art

2.1. Path Planners

The choice of an efficient planning algorithm is crucial for developing an effective framework, Zandegiacomo[16], after an analysis of planners available in the literature decided to consider ABIT*[15]. This algorithm combines truncated anytime graph-based searches with anytime almost-surely asymptotically optimal sampling-based planners. This allows it to quickly find initial solutions and then converge towards the optimum in an anytime manner.

ABIT* is an improvement of BIT*[6], an informed planner which makes use of heuristics to guide the search towards promising directions, while reusing previous information, alleviating the computational weight. It is shown that it is probabilistically complete and asymptotically optimal. Advanced BIT* (ABIT*) extends BIT* to further leverage the separation of search and approximation in sampling-based planning: separating the search process from the approximation process allows each process to focus on its specific task, leading to improved efficiency. The algorithm considered in this work implements the modifications introduced in [16]: throughout the tree construction and development, the algorithm keeps track of the uncertainty in the robot pose due to non ideality of drives, by integrating a gaussian noise from the start to the goal, allowing to consider a safety margin in the collision checking phase proportional to the estimated accuracy of the pose of the manipulator at that point.

2.2. Automatic Harvesting Technologies

The integration of path planners in configuration space with the real environment requires an interface capable of commuting between task space and configuration space. The identification of the goals in task space and their conversion to the configuration space is a critical phase: the first step requires the identification of the grape and its stem (or peduncle) and definition of the cutting point.
This aim has been achieved in "*Grape stem detection using regression convolutional neural networks*" [13], where, for the first time, stem detection is tackled as a regression problem in a way to alleviate the imbalanced data phenomenon that may occur in vineyard images.
After obtaining the region of the stem by the proposed segmentation algorithm, a geometric model is employed to calculate the exact location of the cutting point. This method is able to perform in difficult cases of occlusions and illumination variations.

3. Problem Formalization

This thesis aims at developing a complete framework for motion planning in intricate cluttered environments, in the presence of uncertainty of the robot pose. To achieve this objective, the work focuses on two main aspects: accurate reconstruction of the environments in the surroundings of the target and robust uncertainty aware path planning.

4. Methods

4.1. Planner Development

As mentioned above, the path planning algorithm is based on ABIT*. [16] introduced the idea of estimating the uncertainty, in configuration space, by integrating a gaussian noise along each edge of the tree. Based on the uncertainty, the planner computes the safety margin using chance constraints, This margin is then exploited in the collision checking procedure accounting for the uncertainty in the pose. The procedure adopted in [16] for computing the safety margin implies the environment to be represented by a set of polytopes, each defined with a finite number of inequalities. However, such representation is only applicable in simple not realistic environments.

4.1.1 Uncertainty in Complex Cluttered Environments

The main contribution of this work for what concerns the uncertainty aware motion planning consists in extending this idea to complex cluttered realistic environments. For this reason it has been necessary to abandon the polytopes representation in favor of other representations. The environment model that best fit our needs was OctoMap [8], an integrated framework based on octrees for the representation of three-dimensional environments as a hierarchical data structure, where each node represents the space contained in a cubic volume (voxel). This volume is recursively subdivided into eight subvolumes, until a given minimum voxel size is reached. This representation allows for data retrieved by a variety of sensors (e.g., LiDARS, Depth Camera,...) to be directly interpreted. The model hence obtained, however, no longer allows the use of chance constraints.
This thesis introduces the idea of defining a volume in which a given point may be located given the uncertainty of the robot. Such volume can be defined as the confidence interval of a multivariate normal distribution with a deter-

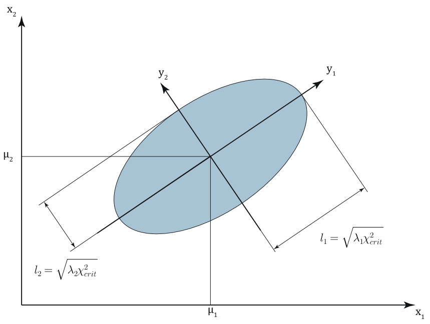


Figure 1: Confidence region in the case of a 2D distribution.

mined mean and covariance, conveniently converted from configuration space to task space through the jacobian of the manipulator.
By setting the critical value of the χ^2 distribution with n degrees of freedom, it is possible to define an hyper-ellipsoid that includes, with the given probability, the position of the considered point (Figure 1). Such ellipsoid is centered in the mean of the distribution, has axis aligned with the eigenvalues of the covariance matrix, and has length of the semi-axis equal to:

$$l_i = \sqrt{\lambda_i \chi^2_{crit}}, i = 1, .., n$$

Once the hyper ellipsoid is defined, it is possible to compute the safety bound for the considered point, which can be chosen as the greatest semi axis. Propagating covariance in joint space allows for the computation of margins for multiple points in the kinematic chain, allowing for a collision checking tailored on the actual uncertainty of each link. Algorithm 1 encodes the previously introduced procedure. It receives as inputs the joint configuration (x), the covariance matrix of the considered configuration (P_{JS}), the list of points of interest on which safety margin calculations are desired ($Points$), and it outputs the list of safety margins for each of the points of interest. P_{TS} represents the covariance matrix in task space, V is the list of the eigenvectors, Λ is the list of the eigenvalues, L is the list of semi-axis of the hyper ellipsoid, χ^2_{crit} is the critical value of the chi-square distribution given the number of degrees of freedom and significance level, and $margins$ contains the list of safety margins. The computed safety margins will be sent to the collision-checking algorithm which will asses if the considered edge is safe to traverse or if it will cause collisions.

Algorithm 1 $SafetyMargins(x, P_{JS}, Points)$

$margins \leftarrow \emptyset$
for all p_i **in** $Points$ **do**
  $J \leftarrow$ `computeForwardKinematics`(x, p_i)
  $P_{TS} \leftarrow J P_{JS} J^T$
  $(V, \Lambda) \leftarrow$ `extractFeatures`(P_{TS})
  $L \leftarrow \emptyset$
  for all λ_i **in** Λ **do**
    $L \xleftarrow{+} \sqrt{\lambda_i \chi^2_{crit}}$
  end for
  $margins \xleftarrow{+} \max_i l_i \in L$
end for
return $margins$

4.1.2 Collision Checking

Collision checking is a crucial aspect of path planning. The most widespread collision checkers, also present in MoveIt![2], are FCL[11] and Bullet[5]. Thanks to its flexibility, speed and compatibility with Octomaps and meshes it has been decided to adopt FCL. This package, however, does not allow to consider a safety margin when checking collisions between two objects. A more suitable alternative, still compatible with the chosen collider types is a modified version of FCL: HPP-FCL[9]. Thanks to its improvements, it is possible to perform a collision check taking into account a collision safety margin. In addition to that, tests shown an improved speed allowing for an overall faster collision checking procedure.

4.1.3 Multiple goals planning

Planning with multiple goals is a fundamental feature when planning in configuration space with redundant manipulators: in fact these robots allow to reach a target with multiple possible configurations. In addition to that, this approach opens to the possibility of defining multiple goals in task space (i.e. different grasping positions), increasing the chances of finding a feasible path.

Goals sampling in Task Space As discussed in Section 2, [13] provides a single cutting point at the center of the stem, however, to increase the chance of success of the planning procedure it has been decided to feed the planner with a set of possible goals.

To obtain a set of goals uniformly distributed in a grasping volume it is possible to adopt the *Workspace Goal Region*[4] technique, which defines a set of six constraints, three positional and three angular (RPY) in the Task Space. A generic WGR consist of three parts:

- $\mathbf{T}^0_w$: reference transform of the WGR from w, the frame that identifies the target (grasping object frame) to the world coordinates represented by 0, a common fixed reference frame;
- $\mathbf{T}^{sample}_{sample'}$: end-effector offset transform from the sampled grasping pose ($sample$) to the end-effector grasping pose, ($sample'$);
- $\mathbf{B}^w$: 6×2 matrix of bounds in the coordinates of w:

$$\mathbf{B}^w = \begin{bmatrix} x_{min} & x_{max} \\ y_{min} & y_{max} \\ z_{min} & z_{max} \\ \psi_{min} & \psi_{max} \\ \theta_{min} & \theta_{max} \\ \phi_{min} & \phi_{max} \end{bmatrix}.$$

Sampling from a single WGR is done by first sampling a random value within the bounds defined by $\mathbf{B}^w$ with uniform probability. These values are stored in the d^w_{sample} vector which expresses the displacement of the sample from the w frame. d^w_{sample} is then converted from (x, y, z, R, P, Y) into the transformation matrix $\mathbf{T}^w_{sample}$, and finally into world coordinates, applying the transformation matrix from the reference frame to the w frame.
An additional transformation matrix is needed to cope with geometrical offsets of the end effector geometry:

$$\mathbf{T}^0_{sample'} = \mathbf{T}^0_w \mathbf{T}^w_{sample} \mathbf{T}^{sample}_{sample'} \tag{1}$$

where $sample$ identifies the frame of the sample, while $sample'$ accounts for the additional displacement of the end effector.

Task Space to Configuration Space Goals Definition The computed goals set has to be converted in configuration space, so that it can be managed by the path planner. This operation requires computation of the Inverse Kinematics. However, having the selected manipulator 7 joints, the Inverse Kinematics problem does not generally have closed form. In addition to that, the manipulator has non Euler shoulder and elbow, making the kinematic chain too complex for the majority of solvers available in the literature. [14] introduced a novel analytical IK solver for Franka Emika Panda, completely based on robot geometry.
To obtain a unique IK solution, the last joint angle $q7$ must be specified as redundancy parameter. Its value is randomly selected and treated as known by the IK solver.
The algorithm returns for each pair of one of the h goals in task space and a random $q7$, a set of possible joint configurations which are sent to the planner, defining the complete goal set.

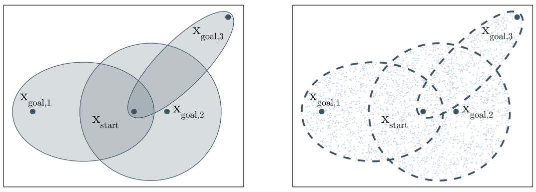
Algorithm 2 $SampleGoals(wgr, m, l, h)$

$GoalsJS \leftarrow \emptyset$
for m **times do**
  $s^{WGR} \leftarrow SampleWGR(wgr.B^w)$
  $s^0 \leftarrow toFixedRefFrame(s^{WGR}, wgr)$
  for l **times do**
    $q7 \leftarrow random(q7_{min}, q7_{max})$
    $GoalsJS \xleftarrow{+} IK(s^0, q7, h)$
  end for
end for
return $GoalsJS$

Algorithm 2 explains the sampling procedure which allows to obtain the desired number of goals in joint space. The functions here introduced are:

- $SampleWGR(wgr)$ returns a randomly extracted sample among the limits of the passed WGR;
- $toFixedRefFrame(s^{WGR}, wgr)$ converts the sample from the WGR frame to the fixed reference frame, introducing, if present, the offset of the EE. The wgr is required because it contains the transformation matrices necessary to compute the conversion introduced in Equation (1);
- $random(q7_{min}, q7_{max})$ returns a random angle for the 7^{th} joint between its limits;
- $IK(s^0, q7, h)$ runs the IK algorithm from [14], requiring to compute h IK solutions for point s^0 given $q7$.

The obtained configurations are not guaranteed to be collision free, for this reason the path planner performs as first step a collision check removing all the unfeasible configurations.



(a) (b)

Figure 2: Sampling from arbitrarily overlapping sets **(a)** with independent sampling, **(b)** by probabilistically rejecting samples in proportion to their individual membership. Images from [7]

Multiple goal sampling region definition Having multiple goals, however, arises a new problem. It is not possible to simply define as many ellipsoids as the number of goals and then sample in each of them separately: it will lead to over-sampled areas, i.e. where the ellipsoids intersect, and under-sampled areas elsewhere (Figure 2a). A consistent solution was found in [7], where an algorithm guaranteeing uniform sampling density among multiple hyper-ellipsoids is introduced. This requirement can be achieved by probabilistically rejecting samples in proportion to their membership in individual sets. This creates a uniform sample distribution for multi-goal L^2 informed sets defined by arbitrarily overlapping individual informed sets (Figure 2b).

4.2. Next Best View

For an effective planning, prior knowledge of the environment is fundamental. Next Best View (NBV), introduced in 1985 in [1], answers the desire of having complete models of a scene. In order to obtain a complete model, NBV selects iteratively the view which, in some sense, provides the most relevant information. [10] addressed the autonomous exploration-inspection problem, introducing the idea of regions of interest towards which to direct the mapping based on information from a radiation sensor.
Building on this idea we developed the weighted sum based information gain of Equation (2) combining discoverable unknown volume and a quality assessment factor of the candidate view, based on the position of the camera with respect to the target.

$$iG = k_1 \cdot unknownVoxels - k_2 \cdot distanceAngle \tag{2}$$

The $unknownVoxels$ represent the unknown volume discoverable from the selected point of view, and it is computed by a dedicated function inspired by [12], and returns the volume of visible unknown voxels in the field of view of the camera.
The $distanceAngle$ factor is computed according to:

$$distanceAngle(\alpha) = \begin{cases} \sqrt{target_y^2 + target_z^2}, & \text{if } 0 \leq \|\alpha\| < \pi/2 \\ 2 \cdot \|target\| - \sqrt{target_y^2 + target_z^2}, & \text{if } \pi/2 \leq \|\alpha\| \leq \pi \end{cases}$$

where α is computed as

$$\alpha = cos^{-1}\left(\frac{target_x}{\|target\|}\right)$$

These formulas lead to the second plot in Figure 3, which is strictly increasing for $0 \leq \alpha \leq \pi$ and strictly decreasing for $-\pi \leq \alpha < 0$, which favors views with the camera close to the target and pointing at it.
This algorithm, executed before the planning allows for an accurate reconstruction of the relevant portion of environment interested in the planning.

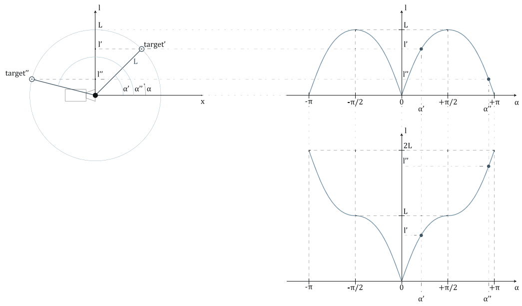


Figure 3: Representation of the distance as function of the angle.

5. Results

The proposed framework was evaluated in simulation in a realistic vineyard environment applying disturbances at the robot base and on the joint angles. The framework demonstrated the ability to efficiently plan paths while considering uncertainty for obstacle avoidance. The positional error on the x, y and z axis caused by the uncertainty on the joints is plotted in Figure 4. In addition the base was subject to

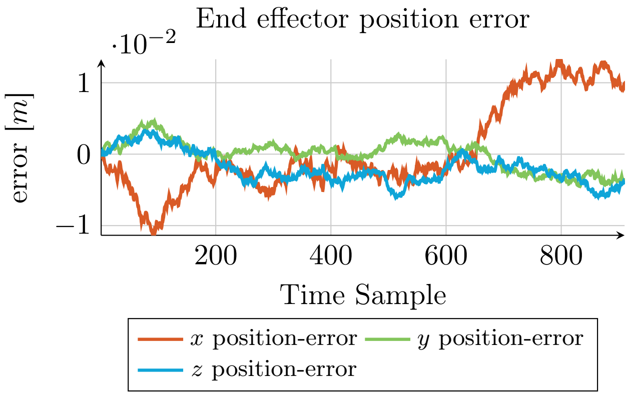


Figure 4: Simulated position error of the end effector.

a random value between $\pm 0.01m$ on each axis. The Target-oriented Next Best View approach was able to accurately reconstruct the environment by selecting informative viewpoints for the sensor to capture. The safety margin method effectively increased the safety of the robot by allowing it to avoid collisions with obstacles.

6. Conclusions

This thesis developed a complete framework for motion planning that can deal with uncertainty in a natural and cluttered environment such as a vineyard, which combines the use of ABIT* for efficient path planning, Target-oriented Next Best View for accurate environment reconstruction, and a novel method for defining a safety margin based on the estimated time-projected uncertainty of the manipulator's pose represented as a χ^2 distribution. The incorporation of a multi-goal approach increased the robustness and exploration of the space, allowing for better manipulation.

References

[1] Connolly C. The determination of next best views. In *Proceedings. 1985 IEEE ICRA*, volume 2, pages 432–435, 1985.

[2] Dave Coleman. Moveit. `https://moveit.ros.org/`, 2007.

[3] Franka Emika. Robot and interface specifications, 2021.

[4] Berenson Dmitry et al. Manipulation planning with workspace goal regions. In *2009 IEEE ICRA*, pages 618–624, 2009.

[5] Fazioli F. et al. Implementation of a soft rigid collision detection algorithm in an open-source engine for surgical realistic simulation. In *2016 IEEE ROBIO*, pages 2204–2208, 2016.

[6] Gammell Jonathan D. et al. Batch informed trees (bit): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In *2015 IEEE ICRA*, pages 3067–3074, 2015.

[7] Gammell Jonathan D. et al. Informed sampling for asymptotically optimal path planning. *T-RO*, 34(4):966–984, 2018.

[8] Hornung Armin et al. Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[9] Jia Pan et al. Hpp-fcl: an extension of the flexible collision library., 2022.

[10] Naazare Menaka et al. Online next-best-view planner for 3d-exploration and inspection with a mobile manipulator robot. *Robot*, 7(2):3779–3786, 2022.

[11] Pan Jiaet et al. Fcl: A general purpose library for collision and proximity queries. In *2012 IEEE ICRA*, pages 3859–3866, 2012.

[12] Selin Magnus et al. Efficient autonomous exploration planning of large-scale 3-d environments. *Robot*, 4(2):1699–1706, 2019.

[13] Theofanis Kalampokas et al. Grape stem detection using regression convolutional neural networks. *Comput Electron Agric*, 186:106220, 2021.

[14] Yanhao He and Steven Liu. Analytical inverse kinematics for franka emika panda – a geometrical solver for 7-dof manipulators with unconventional design. In *2021 9th ICCMA*, pages 194–199, 2021.

[15] Marlin P. Strub and Jonathan D. Gammell. Advanced bit (abit): Sampling-based planning with advanced graph-search techniques. In *2020 IEEE ICRA*, pages 130–136, 2020.

[16] Nicola Zandegiacomo Mazzon. Planning harvesting motion of a manipulator for table grape harvesting, October 2021.