3D Voxel Grid Based Path Planning for Robotic Manipulators using Matrix Multiplication Technique

Alternatives:
- Efficient distance calculation / Efficient repulsive field calculation technique
- Matrix Multiplication-Driven Repulsive Fields for 3D Voxel-Based Robotic Manipulator Path Planning
- Robotic Manipulator Path Planning Optimization Using Matrix-Derived Repulsive Fields Based on 3D Voxel Grid

Jakob Baumgartner, Gregor Klančar

November 14, 2023

Abstract

1 Introduction

2 Background

1-2 PAGES

- PRESENT DISTANCE CALCULATION FOR MANIPULATORS (sensors, lidar, ir, bounding boxes)
- PRESENT PATH PLANNING METHODS FOR MANIPULATOR (optimization, sampling, biological, learning)
- similar to Distance Transform (a kind of inverted distance transform)
- method was inspired by Khatib APF (however, it evolved into a different method)
- different existing APF manipulator implementation articles
- VFH

3 Methodology

3 PAGES

3.1 Optimization Algorithm

- optimization algorithm
- robot kinematics (include the exact-reduced method)
- primary task of distance goal
- secondary task of repulsive field
- damped least squares
- task slowdown option
- secondary task of manipulation measure

3.2 Task Constraints

-equations of occupied and empty space

3.3 Attractive Velocity

In the context of our robotic manipulator, the primary task requires an attractive force that directs the end effector (EE) to the desired position and orientation. In the seminal work of Khatib, this task is defined in the context of the joint coordinate system and joint error. However, we have chosen to define the error of the primary target task in Cartesian coordinate space as the orientation and position error of the EE. This approach allows us to guide the manipulator to its target position and orientation without prior knowledge of its final joint configuration. This is particularly important in scenarios where the manipulator is redundant and there are multiple or potentially infinite positions that meet the requirements of the target and secondary tasks. Our method proves beneficial in efficiently identifying an optimal position in such cases since the target joint configuration of the manipulator is unknown.

To compute the positional error of the end effector (EE), we evaluate the gradient of the squared difference between the vectors representing the current position of the EE and its goal position. Mathematically, this is expressed as:

$$U_{goalAttraction} = -\nabla\left(\frac{(x - x_g)^2}{2}\right)$$

Here, x denotes the vector of the current EE position, and x_g is the vector of the goal position. The term $(x - x_g)^2$ represents the squared Euclidean distance between these two points. The gradient of this squared distance, indicated by $-\nabla$, provides the direction and magnitude of the error in the EE's position.

- attractive field calculation (normalization of attractive force)

3.4 Repulsive Velocity

- object detection is done in task domain and not c-space (more logical)
- repulsive field calculation - matrix "convolution" method
- matrix size and shape selection

- equation for repulsive kernel values (non-linear)
- PLOT: (ERK) kernel graphics
- PLOT: (ERK) linear kernel graphics
- PLOT: kernel field shape
- interpolation of the repulsive field
- what if there are obstacles behind wall (usually not the case, depth sensors show only thin walls, some noise doesnt matter, non-linear kernel, possible additional pre-convolution to convert obstacle grid to edges)
- effecient calculation in dinamic environments, lacking prediction capabilities (MPC)

3.4.1 Kernel Selection

3.4.2 Interpolation

4 Implementation

5 Results

3 PAGES

- include execution times
- PLOT: kernel on robot graphics

6 Discussion

1 PAGE

- add the limitations of such method (already mentioned by Khatib)
- the limitations of local search
- number of parameters that need to be tuned (are there actually that many?)

7 Conclusion