Energy-Aware Path Planning for Autonomous Mobile Robot Navigation

Renan G. Maidana

Pontifical Catholic University of Rio Grande do Sul

renan.maidana@acad.pucrs.br

November 28, 2018

Overview

Introduction

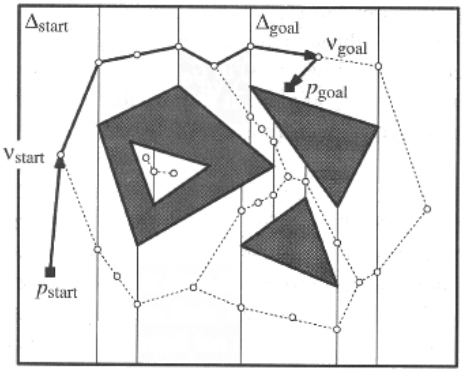
- Embedded processors and Systems-on-Chip have gained attention in mobile robotics recently
  - **Reduced size and weight**;
  - **High performance-per-watt**;
- Many state-of-the-art solutions in mobile robotics require significant computing power
- As SoCs become more powerful to meet this requirement, **the average energy consumption of these solutions increase**
  - Battery life is not a concern for most state-of-the-art algorithms

Introduction

- In this paper, we propose a **path planning** solution for mobile robots, which produces **energy-aware** plans
  - *STRIPS* planning domain with **high-energy zones**;
  - Numerical planning which **minimizes an energy variable at each plan step**;
- We implement a ROS package to integrate the planner with a simulated mobile ground robot, and test it in two experiments:
  - We verify that, due to the energy minimization, our planner **successfully avoids high-energy zones**;
  - We compare the battery discharge curve to a conventional path planning solution, in which our solution **extends the robot's battery life by 1.5 hours, or 42.8%**;

Energy-Aware Planning Domain

- Typically, path planning for autonomous navigation is treated as a graph search problem in a *Configuration Space* (C-Space) [Siegwart et al., 2011]



Energy-Aware Planning Domain

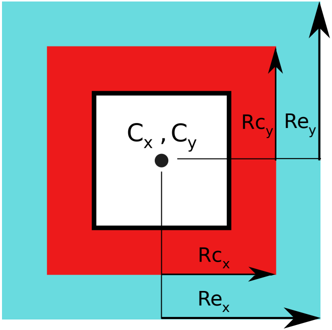
- Being able to describe **different types of actions** is an advantage
  - Both **movement and energy** actions must be considered when planning
- Our objective is to model a planning domain **complex enough to allow planning in an acceptable time** (e.g., $\leq$ 30 seconds), while **considering movement and energy actions**

World Model

- We use the C-Space representation to describe the world in terms of an **occupancy grid** [Moravec and Elfes, 1985]
  - Spaces can be either occupied by an obstacle or free
- In the STRIP domain, we model the world's **boundaries**, the **resolution** and the **obstacles**
  - Due to dimensionality problems, the obstacles are represented as **geometric spaces**

Obstacle Model

$$Obstacle = \begin{cases} \text{Center}: \; C_x, C_y \;\; \in \mathbb{R}^2 \\ \text{Clearance Radii}: \; Rc_x, \; Rc_y \\ \text{Energy Radii}: \; Re_x, Re_y \end{cases}$$



Robot State Model

- The robot's state is modelled as its **2D position (x,y)** and an **energy variable e**, which indicates the energy requirements at each position, according to the obstacles' energy zones
  - If the robot is in a high energy zone, e is increased, and vice-versa

$$Robot = \begin{cases} \text{Position}: & x, y \in \mathbb{R}^2 \\ \text{Energy}: & e \end{cases}$$

Possible Actions

- **Energy**: 2 actions
  - **Increase** robot's e variable, no preconditions
  - **Decrease** robot's e variable, no preconditions
- **Movement**: 16 actions, split between two sets of 8:
  - In **low energy** zones: Up, Down, Left, Right, and diagonals
  - In **high energy** zones: Up, Down, Left, Right, and diagonals

Possible Actions

- Example: Move Up in a **low energy zone**

```
(:action move_up_low_e
  :parameters (?r -robot)
  :precondition (and
    (<= (energy ?r) 0)
    (forall (?o -obstacle)
        (not
            (and
                ; Distance from (y+1 ?r) to (cy ?o) <= (Rc_y ?o)
                ; Distance from (x ?r) to (cx ?o) <= (Rc_x ?o)
            )
        )
    )
    (forall (?o -obstacle)
        (not
            (and
                ; Distance from (y ?r) to (cy ?o) <= (Re_y ?o)
                ; Distance from (x ?r) to (cx ?o) <= (Rc_x ?o)
            )
        )
    ))
  :effect(increase (y ?r) 1)
)
```

Possible Actions

- Example: Move Up in a **high energy zone**

```
(:action move_up_high_e
  :parameters (?r - robot)
  :precondition (and
    (> (energy ?r) 1)
    (forall (?o -obstacle)
        (not
            (and
                ; Distance from (y+1 ?r) to (cy ?o) <= (Rc_y ?o)
                ; Distance from (x ?r) to (cx ?o) <= (Rc_x ?o)
            )
        )
    )
    (exists (?o -obstacle)
            (and
                ; Distance from (y ?r) to (cy ?o) <= (Re_y ?o)
                ; Distance from (x ?r) to (cx ?o) <= (Rc_x ?o)
            )
    ))
  :effect(increase (y ?r) 1)
)
```

Possible Actions

- Separating movement into high and low energy sets induces the planner to **perform energy actions** before entering or exiting a high energy zone
- As the energy is minimized at each plan step, the planner will **choose plans with less energy actions**, thus avoiding movement actions in the high energy set

Constraints

- **To bound the robot's position** to the world dimensions, two constraints are added:
  - $\{x, y\} \leq min_{x,y}$
  - $\{x, y\} \geq max_{x,y}$
- A third constraint, $e \geq 0$, stops the planner from infinitely choosing actions which decrease energy

Practical Implementation

- To test the solution in an autonomous navigation application, **a ROS package** (*ros_enhsp*) **was implemented**
- It wraps a numerical planner called **Expressive Numeric Heuristic Search Planner** (ENHSP)
    - Supports linear and non-linear numerical expressions (e.g., euclidean distance)
    - Supports planning constraints and metric minimization (in this case, energy)

Practical Implementation

- The package has 3 nodes and 1 service:
  - **Problem Interface** (Node)
  - **Problem Generator** (Service)
  - **Planner Interface** (Node)
  - **Planner Dispatch** (Node, optional)
- Apart from ROS and ENHSP, it has **no explicit dependencies**
  - All relevant data is obtained directly through ROS

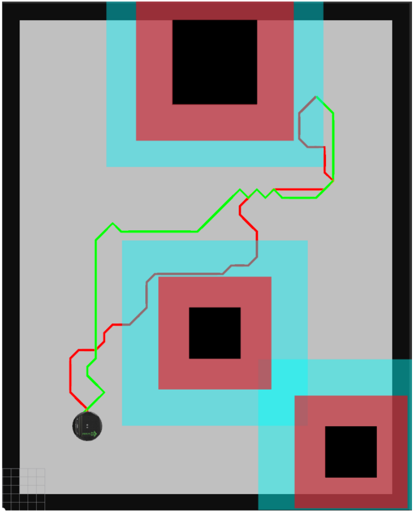
Experiments and Results

- To evaluate the planning domain and implemented package, we perform **two experiments**
  1. To see **if energy minimization significantly changes the plans**
  2. To measure **how longer the robot's battery life is extended** with our solution, compared to a typical ROS path planning solution

- The experiments are run in the Stage simulator:
  - Simulated world
  - Simulated Turtlebot 2 robot
  - Running on an **NVIDIA Jetson TX2**, powered with a **11.1V 1300 mAh LiPo battery**

Experiment 1

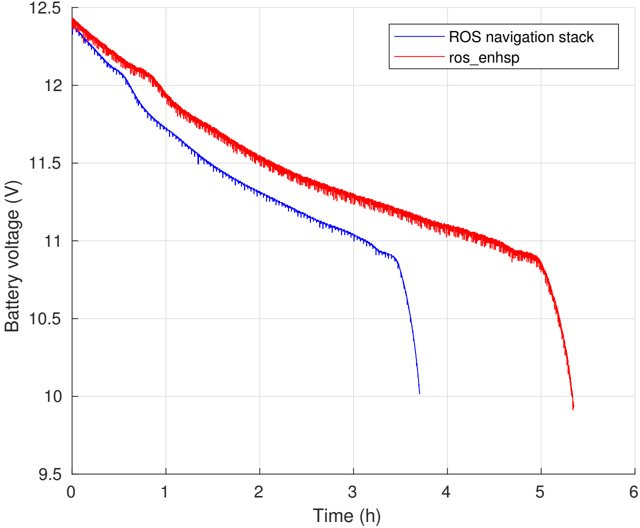
- Red path: **Plan ignoring the energy metric**
- Green path: **Plan considering the energy metric**



Experiment 2

- **Task**: Autonomous navigation
  - The robot must visit 4 user-defined waypoints while autonomously moving and avoiding obstacles
- With a fully charged battery, **we run the task in loop**, once with the conventional path planning and once with our implemented solution, **until the battery runs out**
  - The battery voltage is measured throughout the runs with the Jetson TX2's internal power sensors
  - In ros_enhsp, the energy actions correspond to switching the Jetson TX2's **operation mode**

Experiment 2



Related Work

- Energy efficiency is often a bonus of optimal path planning, and **few works consider energy consumption** as a key aspect in their planning domains [Stentz, 1994, Kruger et al., 2007, Mei et al., 2006, Cabreira et al., 2018]
- For example, [Ooi and Schindelhauer, 2009] proposes a path planner to **minimize energy consumption for both mobility and communication** in a robot, by considering the **distance to the goal** as well as the **transmission power** required for communication

Related Work

- [Plonski et al., 2013] use dynamic programming to find **energy-minimal paths** for a solar-powered ground robot, based on a **power draw model** and a **"solar map"** of an environment
- [Franco and Buttazzo, 2015] explore **energy-aware coverage path planning** using an energy model derived from real measurements.

Conclusions

- We proposed a path planning solution which **accounts for high energy zones** and **integrates energy-changing actions** in the plans
  - We model a STRIPS domain to represent obstacles as geometric spaces to be avoided
  - High energy zones are avoided by minimizing the robot's energy at each plan step
- We **integrate the domain in a ROS package** and test it in two experiments
  - We verify the **significance of energy minimization** in the plans
  - We compare the battery discharge curve of our solution to a conventional path planner, **extending battery life by 1.5 hours (42.8%)**

Limitations

- Our solution has **three main limitations**:
  - Compared to the conventional path planner, our solution's **planning time is longer**
  - We assume that **the robot's response time is directly related to the operating mode**, which may not be true
  - The ros_enhsp package is **specific to the Jetson TX2** embedded computer

Future Work

- Test the solution in a **real environment and robot**;
- Assert the assumption that **response time and operating mode are directly related**;
- Build a tool to **create geometric space representation** for obstacles directly from ROS' occupancy grid maps;
- **Define other energy actions** (e.g., switching off an unnecessary sensor), to maximize energy efficiency in a generic robot;

References I

Cabreira, T. M., Franco, C. D., Ferreira, P. R., and Buttazzo, G. C. (2018).
Energy-aware spiral coverage path planning for UAV photogrammetric applications.
IEEE Robotics and Automation Letters, 3(4):3662–3668.

Franco, C. D. and Buttazzo, G. (2015).
Energy-aware coverage path planning of UAVs.
In *IEEE International Conference on Autonomous Robot Systems and Competitions*, pages 111–117.

References II

Kruger, D., Stolkin, R., Blum, A., and Briganti, J. (2007).
Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments.
In *IEEE International Conference on Robotics and Automation*, pages 4265–4270.

Mei, Y., Lu, Y.-H., Lee, C. S. G., and Hu, Y. C. (2006).
Energy-efficient mobile robot exploration.
In *IEEE International Conference on Robotics and Automation*, pages 505–511.

Moravec, H. and Elfes, A. (1985).
High resolution maps from wide angle sonar.
In *IEEE International Conference on Robotics and Automation*, volume 2, pages 116–121.

References III

- Ooi, C. C. and Schindelhauer, C. (2009).
  Minimal energy path planning for wireless robots.
  Mobile Networks and Applications, 14(3):309–321.

- Plonski, P. A., Tokekar, P., and Isler, V. (2013).
  Energy-efficient path planning for solar-powered mobile robots.
  Journal of Field Robotics, 30(4):583–601.

- Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D. (2011).
  Introduction to Autonomous Mobile Robots.
  The MIT Press, 2nd edition.

- Stentz, A. (1994).
  Optimal and efficient path planning for partially-known environments.
  In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3310–3317.