Chapter 2

Domains

Figure 2.1: Several platforms used in navigational research. These include RoboCup small-size league soccer (left), a fixed wing UAV (center), and the QRIO humanoid robot (right).

In research has been motivated by several platforms spanning a wide variety of the mobile robot parameter space. For each domain, a navigational system was developed using randomized planning with an iterated approach to replanning. The three primary domains which are the focus in this work are the following:

- RoboCup small-size league multi-robot soccer
- A kinematic theater navigation system for fixed-wing UAVs
- A 2.5D (heightfield) path planner for the QRIO humanoid robot

The properties of each of these domains and their navigational requirements are described in detail the subsequent sections.

2.1 RoboCup Small-Size Multi-Robot Soccer

The primary motivating domain for our navigational research since 2001 has been the the RoboCup F180 "small-size" league [55]. This league involves teams of five small robots, each up to 18cm in diameter and up to 15cm height. The robot teams are entered into a competition to play a type of soccer against opponent teams fielded by other research groups. Two halves of 15 minutes each are played, and during a game no human input is allowed. The match is refereed by a human to ensure fair play, and the referee's signals encoded and sent via a serial link to provided to each team. Thus the two robot teams must compete using full autonomy in every aspect of the gameplay.

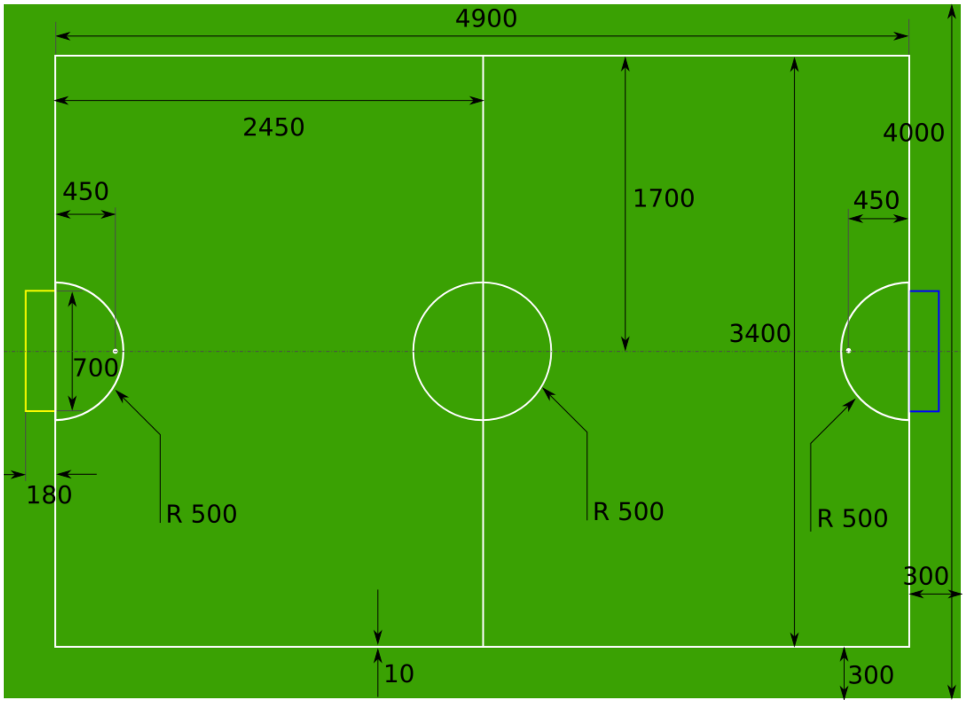

Figure 2.2: The dimensions of the current RoboCup small-size field.

The field of play is a carpet measuring 4.9m by 3.8m, as shown in Figure 2.2. The robots, in addition to fitting within a 18cm cylinder, are also forbidden from covering more than 20% of the ball, defined by the area of the ball falling within the convex hull of the robot when projected onto the ground plane. Robots are not allowed to gain full control of the ball and remove all of its degrees of freedom, promoting team play [79]. An example of a small-size game (on an older, half-size field) can be seen in Figure 2.3

Figure 2.3: Two teams are shown playing soccer in the RoboCup small size league.

For the team control system, offboard sensing, computation, and communication are allowed. This has lead nearly every team to adopt a centralized approach for most of the robot control [19,45,78] Sensing in a typical system is provided by two or more overhead cameras mounted 4m above the field. The camera signals then feed into a central computer to process the image and locate the 10 robots and the ball on the field 30-60 times per second. The systems implement soccer strategies, behaviors, and control on the centralized computer. Finally, velocity commands are broadcast via a radio link the the individual robots, which implement velocity control on the individual robots.

Due to its competitive nature, teams in the small-size league have rapidly advanced the boundaries of robotic technology applicable to the domain. Since its start in 1997, teams have improved dramatically in capability. Top teams now have holonomic robots which can travel at speeds of over $2m/s$, with maximum accelerations between $3 - 6m/s^2$. Four generations of small-size hardware from the CMUnited and CMDragons teams are shown in Figure 2.4.

Figure 2.4: Five generations of Carnegie Mellon robots: (from left) 1997, 1998-99, 2001, 2002-03, 2006

Kicking devices have been implemented with can propel the ball at up to $15m/s$. Ball control devices composed of spinning rubber bars have been introduced to partially control the ball by imparting backspin, allowing robots to receive passes and also drive short distances [1]. In the last two years, special "chip kicking" devices have been introduced which can impart vertical as well as horizontal velocity to the ball, while remaining within the 20% holding rule. The most advanced of these kickers can propel the ball a distance of $4.5m$ in the air with the ball reaching a maximum height of $1.5m$ from the ground plane.

The speeds involved require every module to run in at real-time rates to minimize latency. Each component of the system must be efficient enough to leave sufficient computing resources for all the other modules and tasks. Since five robots must be controlled at up to 60Hz, this leaves a realistic planning time budget of only $1ms - 2ms$ for each robot. The approach taken with these robots is to adopt a path planner which ignores dynamics, and pair it with a dynamics safety system which maintains safety while incorporating a more complete velocity and acceleration model of the robot. An example of a planning search tree generated in a small-size like environment (with additional obstacles added) can be found in Figure 2.5.

2.2 Fixed Wing UAV

The second platform is an autonomous unmanned air vehicle (UAV), and in particular an autopilot designed for the small UAV shown in Figure 2.1. The navigation system is part of a much larger software system for autonomous collaborative control of multiple UAVs. The

---

[1] The current rules limit a robot to dribbling only 50cm before it must lost contact with the ball. This was introduced in 2004 to promote team play

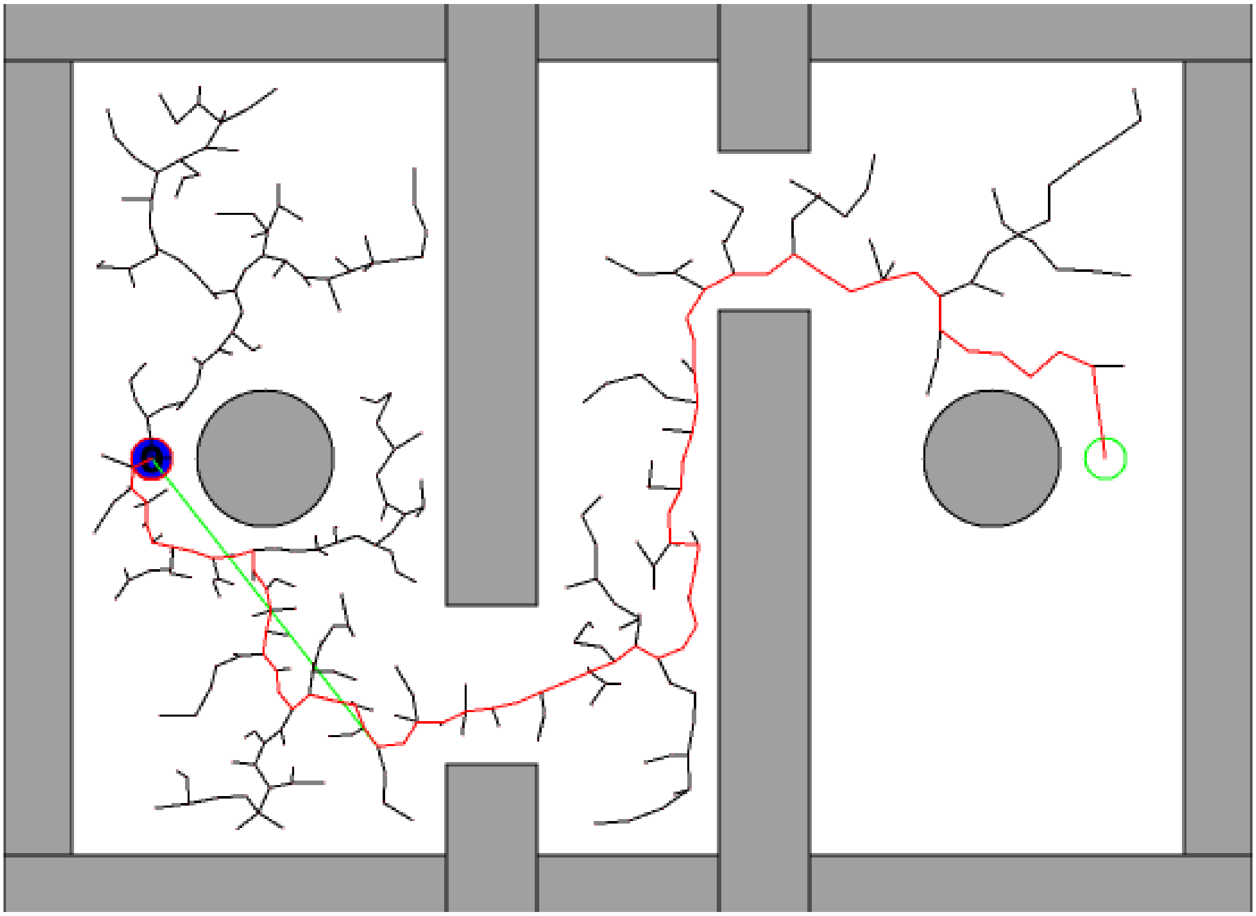

Figure 2.5: A robot on the left finds a path to a goal on the right using the ERRT algorithm.

overall software system consisted of an operator interface, numerous behaviors, a market-based task assignment system, an area coverage planner, and a route planning module. This was then fed to the hardware system which consisted of a radio link to a UAV autopilot. In this project, this stage could be simulated in software or using a hardware-in-the-loop (HIL) simulator which paired real radio links and autopilot hardware modules with a software aircraft simulator. The system was designed to easily incorporate real UAVs in a possible follow-on to the project.

The author was responsible for developing the route planning module which planned from a given UAV state to either a final destination point or a destination orbit. The motion planner had to support terrain data as well as primitive obstacles (no-fly zones), and model the kinematic constraints of the fixed-wing UAV. The aircraft, and thus the planner must operate in 3D at low to intermediate altitude, avoiding both the terrain and the user specified obstacles. The UAV's kinodynamics are highly constraint compared to a ground vehicle or rotorcraft; Despite the UAV's relatively small size ($3.5m$ wingspan), the minimum turning radius is $300m$. In addition, climb and descent rates are limited to $5m/s$. The output of the planner was a set of of feasible waypoints. The autopilot can accept small sets of waypoints every few seconds, but the planner interfaced with a path buffer which would manage the waypoints so that the number of waypoints was not a practical limitation. The route planning module was called both for actual navigation problems as well as for deriving cost estimates for the market based task allocation of multiple UAVs. Thus, while the timing was relaxed, answers would have to be returned in less than 10 seconds to allow the operator interface to operate smoothly, with shorter paths expected to require less planning time. Although the timing budget is much greater than the small-size environment, due to the 3D nature of the problem and the constrained kinodynamics, the problem to be solved is much more difficult. An example of a kinematically constrained search tree is shown in Figure 2.6.

2.3 QRIO Humanoid Robot

The Sony QRIO robot is a small humanoid robot with both complex actuation and significant sensory capabilities [41]. The robot, shown in Figure 2.7, is able to use a stereo pair located in the robot's head to generate a 3D occupancy grid model of the environment. This data can be used to generate a 2D grid of height values, or a *heightmap*, which can be used for local path planning. Existing robot models are able to generate the 3d occupancy grid relative to the robots torso location projected onto the ground. A locomotion system was also provided to ensure balance and generate walking steps to reach ego-relative positions [41]. Finally, a module exists for traversing step obstacles if the robot is positioned approximately in front of the transition. Such a module helps compensate for the finite resolution of the occupancy

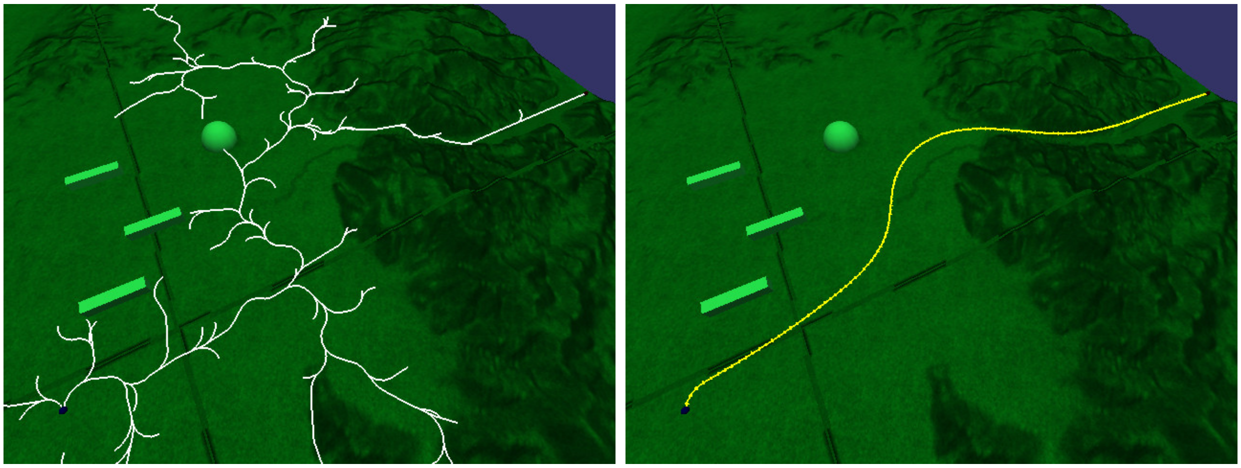

Figure 2.6: A kinodynamicly-limited search tree (left), and the corresponding simplified plan (right) for the UAV. The plan length is approximately 13km long.

grid representation, which is not sufficient for the high accuracy needed to traverse a step obstacle. This is because QRIO, like many humanoids, is limited in its step length, so positioning at the transition must be accurate for traversal to succeed.

A navigation module for QRIO is responsible for generating paths on a plane which avoid obstacles, as well as finding plane transitions that the robot can traverse. The the robot operates in a "2.5D" segmented heightfield, and the planner must generate both regular egomotion goals for the walking controller as well as appropriate calls to the climbing/descent module when transitions are reached. Due to the Open-R architecture [40], planning can either run onboard the robot, or on an offboard computer via a wireless ethernet link.

The robot has a slower relative navigation speed when compared to the small-size robots, and thus a more relaxed timing constraint. Planning can take up to a few seconds, and batches of step commands can be executed which take from 2-10 seconds, at approximately 2 seconds per full step. The environment is complex in terms of obstacle geometry as it is derived from an occupancy grid model which is updated incrementally with stereo data. An example of a domain is shown in Figure 2.8, where a robot was commanded to navigate to the top of an obstacle after starting from the floor in the office environment.

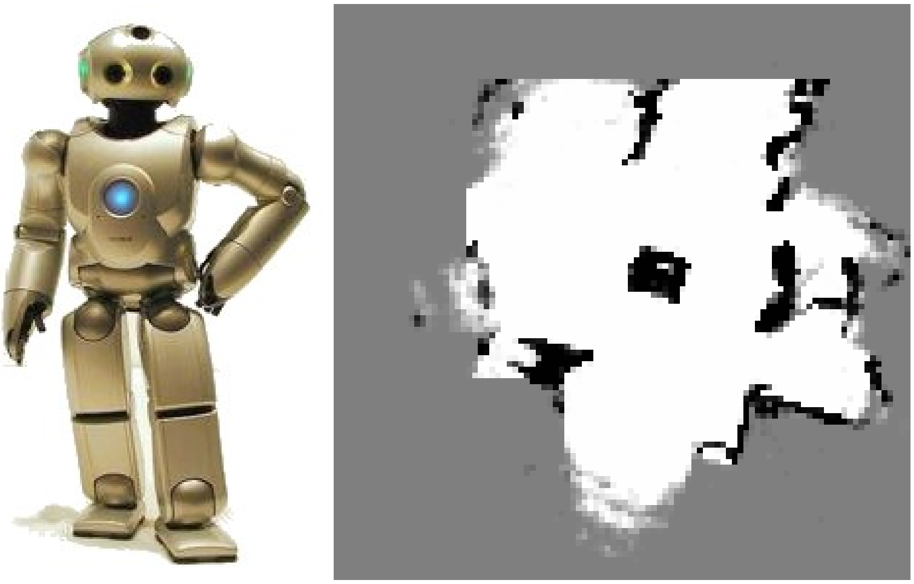

Figure 2.7: The Sony QRIO robot (left), and a stereo generated occupancy grid environment map (right).

2.4 Summary of Domains

The main properties of the domains can be summarized in Table 2.4. The small-size domain is used throughout this document for testing and development of algorithms, while the solutions developed for the UAV and humanoid domains are described further in Chapter 4.7.

Domain	Environment d.o.f.	Planning d.o.f	Planning Time	Control Period
RoboCup F180 (static)	2	2	2 ms	1/60 s
RoboCup F180 (dynamic)	2	6	2 ms	1/60 s
QRIO Humanoid	3	2.5	1-2 s	10 s
Fixed-Wing UAV	3	5	5-10 s	30-60 s

Table 2.1: A summary of the planning properties of several real robot domains

Figure 2.8: QRIO on top of an autonomously reached step obstacle.