Persistent Tasks for Robots in Changing Environments

Stephen L. Smith
Electrical and Computer Engineering
University of Waterloo
Waterloo ON, N2L 3G1 Canada
stephen.smith@uwaterloo.ca

Mac Schwager
GRASP Lab, University of Pennsylvania
Philadelphia, PA 19104 USA, and
CSAIL, MIT
Cambridge, MA 02139 USA
schwager@mit.edu

Daniel E. Soltero Daniela Rus
CSAIL, MIT
Cambridge, MA 02139 USA
soltero@mit.edu, rus@csail.mit.edu

I. Introduction

In this talk we will discuss some of our recent work [12, 13] on the problem of controlling robots to perpetually act in a changing environment, for example to clean an environment where material is constantly collecting, or to monitor an environment where uncertainty is continually growing. Each robot has a small footprint over which to act (e.g. to sweep or to sense). The difficulty is in controlling the robots to move so that their footprints visit all points in the environment regularly, spending more time in those locations where the environment changes quickly, without neglecting the locations where it changes more slowly. This scenario is distinct from most other sweeping and monitoring scenarios in the literature because the task cannot be "completed." That is to say, the robots must continually move to satisfy the objective. We consider the situation in which robots are constrained to move on fixed paths, along which we must control their speed. Figure 1 shows three robots monitoring an environment using controllers designed with our method.

We model the changing environment with a scalar valued function defined, which we call the *accumulation function*. The function behaves analogously to dust accumulating over a floor. When a robot's footprint is not over a point in the environment, the accumulation function grows at that point at a constant rate, as if the point were collecting dust. When a robot's footprint is over the point, the accumulation function decreases at a constant rate, as if the dust were being vacuumed by the robot. The rates of growth and decrease can be different at different points in the environment.

We focus on the situation in which the robots are given fixed, closed paths on which to travel, and we have to carry out the persistent task only by regulating their speed. The idea of decoupling the path planning from the speed control has proved useful in dealing with complex trajectory planning problems [7]. In some cases, paths may be given through motion constraints. When we were are free to plan the path, then we can employ an off-line planner to generate paths that are optimal according to some metric (for example, by using the planner in [11]).

Our approach to the problem is to represent the space of all possible speed controllers with a finite set of basis functions, where each possible speed controller is a linear combination of those basis functions. A rich class of controllers can be represented in this way. Using this representation as our foundation, we are able to solve a linear program (LP) to generate optimal speed controllers.

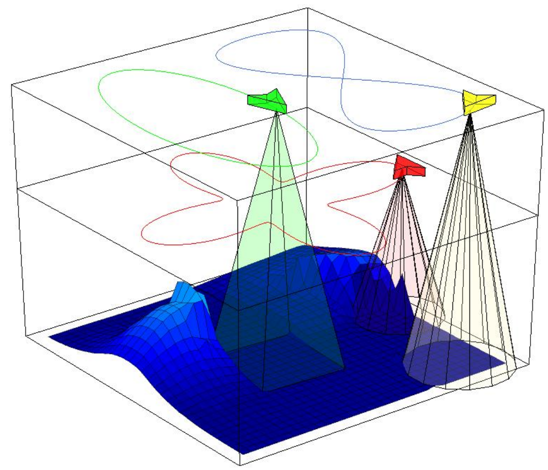


Fig. 1: A persistent monitoring task using three robots with heterogeneous, limited range sensors. The surface shows the accumulation function, indicating the quantity of material to be removed in a cleaning application, or the uncertainty at each point in a sensing application. The accumulation function grows when a robot's footprint is not over it, and decreases when the footprint is over it. Each robot controls its speed along its prescribed path so as to keep the surface as low as possible everywhere.

A. Related Work

There is a large body of related work, and we refer the reader to our recent submissions on the subject [12, 13] for a full review. The main areas of related work include environmental monitoring, sensor sweep coverage, lawn mowing and milling, and patrolling. In the environmental monitoring literature, the goal is commonly to control a robot, or group of robots to estimate the state of the environment [6, 4, 14, 9].

Another area is sweep coverage, or lawn mowing and milling problems, in which robots with finite sensor footprints move over an environment so that every point in the environment is visited at least once by a robot [2, 3]. Finally, in patrolling, an environment must be continually surveyed by a group of robots such that each point is visited with equal frequency [5, 8], or in some cases with specified frequencies [1].

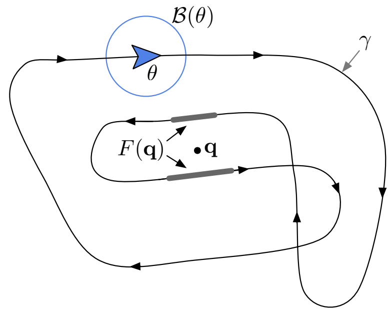


Fig. 2: An illustration of a curve γ followed by one of the robots. The robot is located at θ and has footprint $\mathcal{B}(\theta)$. The set $F(\mathbf{q})$ of robot positions θ for which the footprint covers q are shown as thick grey segments of the curve.

In this work, we consider a more complex environment model than in the sweep coverage and lawn milling and mowing work, but a less sophisticated model than in environmental monitoring. The result is that we are able to obtain speed controllers with strong performance guarantees.

II. Problem Formulation

We consider n robots, indexed by $r \in \{1, 2, \ldots, n\}$. Each robot is constrained to move along a pre-determined path $\gamma_r : [0, 1] \to \mathbb{R}^2$, where $\gamma_r(0) = \gamma_r(1)$. Path γ_r is parametrized by $0 \leq \theta_r \leq 1$, which is assumed to be the arc-length parametrization. The robot's position at time t can be described by $\theta_r(t)$, its position along the curve γ_r. A single robot example is shown in Figure 2. Each robot has a sensor/sweeping footprint $\mathcal{B}_r(\theta_r(t))$. This footprint could be thought of, for example, as the cleaning surface of a sweeping robot. The environment contains a finite number of points of interests $\mathbf{q} \in Q$. These finite points could be the discretization of a continuous environment. A scalar field $Z(\mathbf{q}, t) \geq 0$ is defined over the points of interest $\mathbf{q} \in Q$. The field (called the accumulation function) behaves analogously to dust accumulating over a floor. At a point of interest $\mathbf{q} \in Q$, the field increases at a production rate of $p(\mathbf{q})$ when not covered by any robot footprints, and it is consumed at a rate of $c_r(\mathbf{q})$, by each robot r whose footprint is covering $\mathbf{q}$. More specifically,

$$\dot{Z}(\mathbf{q}, t) = \begin{cases} p(\mathbf{q}) - \displaystyle\sum_{r \in \mathcal{N}_{\mathbf{q}}(t)} c_r(\mathbf{q}), & \text{if } Z(\mathbf{q}, t) > 0, \\ \Big(p(\mathbf{q}) - \displaystyle\sum_{r \in \mathcal{N}_{\mathbf{q}}(t)} c_r(\mathbf{q})\Big)^+, & \text{if } Z(\mathbf{q}, t) = 0, \end{cases} \tag{1}$$

where $\mathcal{N}_{\mathbf{q}}(t)$ is the set of robots whose footprints are over the point $\mathbf{q}$ at time t, $\mathcal{N}_{\mathbf{q}}(t) := \{r \mid \mathbf{q} \in \mathcal{B}_r(\theta_r(t))\}$. We assume that robot r knows the parameters $p(\mathbf{q})$ and $c_r(\mathbf{q})$ for the field evolution. However, the accuracy of the model is not crucial, as we show analytically and in simulations that our method has strong robustness with respect to errors in $p(\mathbf{q})$.

III. Computing Speed Controllers

The goal of this work is to compute speed controllers for each robot such that the field (accumulation function) $Z(\mathbf{q}, t)$ is bounded for all time t, and for all points $\mathbf{q}$. We assume that for each robot r, and each position θ_r on the robot's path γ_r, there is a minimum and maximum allowable speed $v_{r,\min}(\theta_r)$ and $v_{r,\max}(\theta_r)$. This allows us to express constraints on the robot speed at different points on the curve. For example, for safety considerations, the robot may be required to move more slowly in certain areas of the environment, or on high curved sections of the path. We first show that we can consider simple speed controllers of the form $v_r : [0, 1] \to \mathbb{R}_{>0}$, which map robot r's position θ_r to a speed $v_r(\theta_r) \in [v_{r,\min}(\theta_r), v_{r,\max}(\theta_r)]$.

We show that a necessary and sufficient condition for stability of the field is that

$$\sum_{r=1}^{n} \frac{\tau_r(\mathbf{q})}{T_r} c_r(\mathbf{q}) - p(\mathbf{q}) > 0,$$

for all points of interest $\mathbf{q} \in Q$, where T_r is the period (or cycle time) of robot r along its path γ_r, and $\tau_r(\mathbf{q})$ is the amount of time per period that robot r's footprint is covering the point of interest $\mathbf{q}$. We then develop a method for producing a speed controller $v_r(\theta_r)$ for each robot r, which maximizes the *stability margin*

$$\min_{\mathbf{q} \in Q} \left(\sum_{r=1}^{n} \frac{\tau_r(\mathbf{q})}{T_r} c_r(\mathbf{q}) - p(\mathbf{q}) \right). \tag{2}$$

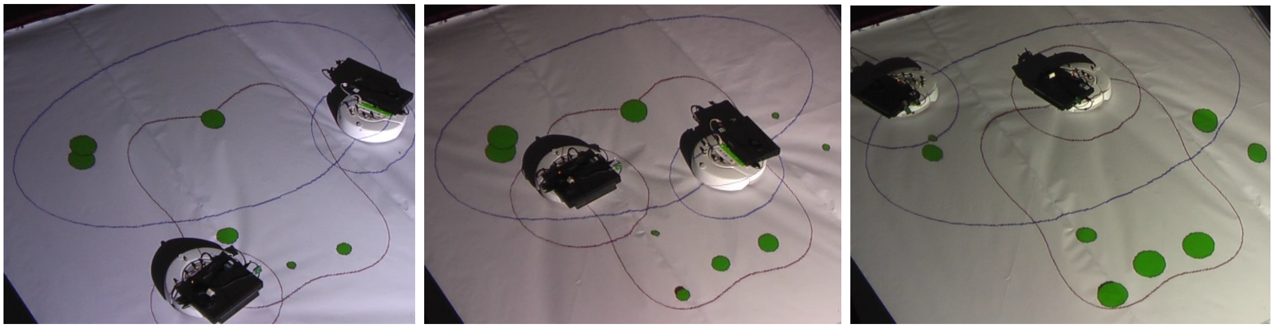
Our approach is to parametrize each speed controller v_r by a finite set of basis functions:

$$v_r^{-1}(\theta_r) = \sum_{j=1}^{n_r} \alpha_{rj} \beta_{rj}(\theta_r),$$

where n_r is the number of basis functions for the rth robot, and $\alpha_{rj} \in \mathbb{R}$ and $\beta_{rj} : [0, 1] \to \mathbb{R}_{\geq 0}$ are robot r's jth parameter and basis function, respectively. We then show that we can optimize over the coefficients of the basis function representation by solving a linear program. It turns out that by maximizing the stability margin, we obtain a controller which has maximum robustness to errors in the field evolution model $p(\mathbf{q})$. In addition, we show that for a single robot, a linear program can be formulated whose solution gives the controller that minimizes the steady-state maximum field value.

IV. Collision Avoidance and Experiments

In the solution procedure described above, robot paths may intersect one another, and thus there may be collisions when robots execute their paths. We have developed a method to address this issue in [13], where we developed a collision avoidance procedure for persistent monitoring. The procedure is based on avoiding collisions by stopping robot motion. To do this efficiently, we have determined methods for quantifying the effect of stopping on the stability margin of the system. The collision avoidance operates by identifying *collision zones* in which collisions could occur. We then avoid collisions by stopping and restarting robots so that at most one robot occupies a given collision zone at any moment in time. We have also design a procedure to avoid *deadlocks*; a situation



(a) t = 5 seconds

(b) t = 12 seconds

(c) t = 34 seconds

Fig. 3: Snapshots at different times of a distributed implementation for the persistent monitoring task with collision avoidance for two robots. The points of interest are represented as green-filled circles, whose size is proportional to the value of the accumulation function $Z(\mathbf{q}, t)$ for each point $\mathbf{q}$. Each robot's footprint is represented by a disk centered at the robot's location, and it is the same color as the trajectory that robot is following.

in which a group of robots are all stopped, and are waiting for each other to move before resuming motion.

We have implemented persistent monitoring controllers with collision avoidance on a multi-robot system consisting of two iRobot Create robots. Figure 3 shows three snapshots of the evolution of the system in the implementation. This implementation was executed in a distributed way. Each robot only knew information about itself, and communicated with the other robot when entering a collision zone in order to decide whether to continue its trajectory or stop to avoid collision. The robots tracked their paths with their speed profiles using a controller based on dynamic feedback linearization [10]. We have also performed an implementation on quadrotor helicopters.

V. Conclusions

In this work we propose a model for persistent sweeping and monitoring tasks and derived controllers for robots to accomplish those tasks. We specifically considered the case in which robots are confined to pre-specified, closed paths, along which their speed must be controlled. We found speed controllers by solving simple linear programs. We also discussed a collision avoidance procedure, and and showed results from recent physical implementations.

References

[1] M. Ahmadi and P. Stone. A multi-robot system for continuous area sweeping tasks. In *IEEE Int. Conf. on Robotics and Automation*, pages 1724–1729, Orlando, FL, May 2006.

[2] E. M. Arkin, S. P. Fekete, and J. S. B. Mitchell. Approximation algorithms for lawn mowing and milling. *Computational Geometry: Theory and Applications*, 17 (1-2):25–50, 2000.

[3] H. Choset. Coverage for robotics – A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):113–126, 2001.

[4] J. Cortés. Distributed Kriged Kalman filter for spatial estimation. *IEEE Transactions on Automatic Control*, 54 (12):2816–2827, 2009.

[5] Y. Elmaliach, N. Agmon, and G. A. Kaminka. Multi-robot area patrol under frequency constraints. In *IEEE Int. Conf. on Robotics and Automation*, pages 385–390, Roma, Italy, April 2007.

[6] B. Grocholsky. *Information-Theoretic Control of Multiple Sensor Platforms*. PhD thesis, University of Sydney, 2002.

[7] K. Kant and S. W. Zucker. Toward efficient trajectory planning: The path-velocity decomposition. *International Journal of Robotics Research*, 5(3):72–89, 1986.

[8] N. Nigram and I. Kroo. Persistent surveillance using multiple unmannded air vehicles. In *IEEE Aerospace Conf.*, pages 1–14, Big Sky, MT, May 2008.

[9] J. Le Ny and G. J. Pappas. On trajectory optimization for active sensing in gaussian process models. In *IEEE Conf. on Decision and Control*, pages 6282–6292, Shanghai, China, December 2009.

[10] G. Oriolo, A. De Luca, and M. Vendittelli. WMR control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, 10(6):835–852, 2002.

[11] S. L. Smith and D. Rus. Multi-robot monitoring in dynamic environments with guaranteed currency of observations. In *IEEE Conf. on Decision and Control*, pages 514–521, Atlanta, GA, December 2010.

[12] S. L. Smith, M. Schwager, and D. Rus. Persistent robotic tasks: Monitoring and sweeping in changing environments. *IEEE Transactions on Robotics*, July 2010. Submitted. Available at `http://arxiv.org/abs/1102.0603`.

[13] D. E. Soltero, S. L. Smith, and D. Rus. Collision avoidance in trajectory tracking for multi-robot persistent tasks. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, San Francisco, CA, October 2011. Submitted.

[14] F. Zhang and N. E. Leonard. Cooperative filters and control for cooperative exploration. *IEEE Transactions on Automatic Control*, 55(3):650–663, 2010.