1995

The Geometry of a Robot Programming Language

Daniel E. Koditschek
University of Pennsylvania, kod@seas.upenn.edu

Recommended Citation

Daniel E. Koditschek, "The Geometry of a Robot Programming Language", *WAFR: Proceedings of the Workshop on Algorithmic Foundations of Robotics* , 263-268. January 1995.

NOTE: At the time of publication, author Daniel Koditschek was affiliated with the University of Michigan, Ann Arbor. Currently, he is a faculty member in the Department of Electrical and Systems Engineering at the University of Pennsylvania.

Abstract
This paper explores the problem of building robot navigation plans via scalar valued functions in the face of incomplete information about the configuration space such as might be available from onboard sensors. It seems as though syntactical aspects of navigation function construction may play an important role. This problem provides an important concrete instance of the need for intelligent control.

The Geometry of a Robot Programming Language

Submitted: First Workshop on Algorithmic Foundations of Robotics
February 17 - 19, 1994
San Fransisco, CA

Daniel E. Koditschek *

Artificial Intelligence Laboratory
EECS Department, College of Engineering
University of Michigan

Abstract

This paper explores the problem of building robot navigation plans via scalar valued functions in the face of incomplete information about the configuration space such as might be available from onboard sensors. It seems as though syntactical aspects of navigation function construction may play an important role. This problem provides an important concrete instance of the need for intelligent control.

1 Introduction

Let a robot be conceived rather broadly as an electromechanical system whose input signals take the form of commands to actuators — motors that deliver torque or force — and for which all goals of interest involve *work* in the physical sense — the application of specified forces over specified motions. If we seek to program such systems then we are bound eventually to compile our more abstractly expressed goals into their "machine language:" forced Lagrangian dynamics.

Automated reasoning about the static properties of the physical environment has been a central concern of computational geometry for at least a decade. The dynamics of physical settings also admits a geometric representation. This has been understood for several decades [1], but has not generally been exploited in engineering practice. One presumes that geometry should offer a natural language for expressing abstract goals relating to the physics of manipulation [6].

The possibility of addressing both the syntactic and dynamical aspects of robotics in the same formalism has nowhere seemed to me as great as in path planning. For here, the goal set is a point located in a set with a boundary, hence, in dynamical terms the task may be encoded as a point attractor with maximal domain of attraction. Such a dynamical setup can be re-expressed algebraically via a special Lyapunov function — a geometric construct that we have termed a *navigation function* in earlier work [9]. In turn, scalar valued functions may be interpreted as expressions of rules. [1] Thus, there is at least a hint of a formal connection between robot rules and robot controls in this setting.

This paper offers a very sketchy and speculative rumination on the possible use and shape of such a connection. There are many syntactical or rule based planners and controllers in the AI world, and various efforts have been made to project these onto the domain of real-time control. It seems certain that this would be a profitable enterprise. It is my prejudice that such projections cannot have wider applicability than case by case tinkering absent a more formal characterization. I manage in this paper merely to offer a glimpse of some questions whose answers might set the stage for such a formalism in the specific context of robot navigation. Thus, I mean by the word "syntax," some formal automaton that outputs symbol strings — "sentences" — with the properties

Tuning: any string represents (and can be"compiled" into) a navigation function for a known (suspected) configuration space

Switching: when a new configuration space is presented for which the present symbol string is not valid, the automaton recognizes this fact, switches state and generates a new valid string

Identifying the nature of the "syntax" that might emerge from this point of view of path planning has two motivations. The first is abstract. A recent

*This work was supported in part by the National Science Foundation under grants IRI-9123266 and IRI-9216823.

[1] This notion seems to be the starting point of fuzzy control.

study [5] has revealed that there are mathematical obstructions to the kind of online smooth tuning one might wish for even when the topology of the configuration space is fixed — apparently, switching is built in to this problem. Yet, on the other hand, we have recently begun to realize that certain "formal" structures seem to yield valid navigation functions for topologically diverse model spaces. It seems important to understand the relation of the algebraic structure to the analytic properties of these constructions. Second, apart from the intrinsic intellectual interest, understanding how to devise programs that yield flexibly re-configurable navigation plans becomes critically important when faced with incomplete information about the environment. The navigation functions offer a concrete means of parametrizing plans according to the information available. As that information about the environment changes, their adjustment will define a hybrid dynamical system.

2 Navigation Functions

In this section I explore the relationship between the syntactic and the analytical aspects of navigation problems. Here, attention is focussed on the tuning functions and symbol strings themselves rather than possible tuning dynamics and switching automata that might generate them.

2.1 Analytical Aspects

Let the workspace, $\mathcal{W}$, be some compact connected subset of $\mathbb{R}^3$ with non-empty interior. Let $\mathcal{R}$ be the set of all placements of a specified kinematic chain (formed from a finite number of compact connected subsets of $\mathbb{R}^3$ with non-empty interior) in $\mathbb{R}^3$ and let $\mathcal{Q}(\mathcal{W}, \mathcal{R})$ denote the resulting configuration space — the set of all free placements of the chain in the workspace. Suppose there is a distinguished "destination," $q^* \in \mathcal{Q}$. For the entire paper we shall work within the generalized damper model of dynamics

$$\dot{q} = u$$

where input force, u, is proportional to a velocity rather than acceleration. The extension of these ideas to the true Lagrangian case is quite straightforward [8].

The navigation functions were defined in [9] to be scalar valued maps, Morse functions, taking their maximum value (unity) on the entire boundary of $\mathcal{Q}$, their minimum value (zero) at the destination q^*, and possessing no other local minima. If $\varphi(q)$ is a navigation function then call the vector field $f(q)$ a *plan* if it admits φ as a Lyapunov function. For example, note that the gradient vectory field defined by φ is a plan. Given a plan, f, the feedback,

$$u = f(q)$$

results in a closed loop system whose trajectories never pass through the boundary and converge to q^* from all but a set of initial conditions of zero measure.

We have shown that every compact connected smooth manifold with boundary admits a *navigation function* and that a navigation function for one manifold, $\varphi : \mathcal{M} \to \mathbb{R}$, composed with a diffeomorphism from another, $h : \mathcal{Q} \approx \mathcal{M}$, can serve as a navigation function for the second manifold, $\varphi \circ h : \mathcal{Q} \to \mathbb{R}$ [9]. We have furnished computationally effective recipes for such diffeomorphisms in the case that $\mathcal{M}$ is a punctured sphere and all the information about the boundaries of $\mathcal{Q}$ is available [12].

2.2 Algebraic Aspects

I have just now outlined a "theory" of navigation functions considered as being topologically rigid but geometrically extensible. That is as much about them as I know how to say formally. But in fact, we have employed the same general algebraic structure for several topologically very different models, $\mathcal{M}$. It seems compelling to try to understand what can be formalized about this algebraic structure, how it might be improved, and over what class of models it and its descendants may yield a navigation function.

2.2.1 An Algebraic Construction

Consider the function

$$\varphi = \delta_l \circ \gamma / \beta$$

where, δ_l is a member of a parametrized family of diffeomorphism from $[0, \infty]$ to $[0, 1]$, γ is the Euclidean distance to the goal in $\mathcal{M}$ and β goes to 0 on the boundary, $\partial\mathcal{M}$. Setting the boundary value is achieved by taking β to be the algebraic product of functions that vanish on each connected component of the boundary

$$\beta = \beta_0 \cdot \beta_1 \cdot \ldots \cdot \beta_k.$$

Consider now a family of loci $\{b_i\}_{i=0,k} \subset E^n$, each point, b_j forming the center of a disk, $\mathcal{D}_j$ of radius ρ_j. Suppose all the "smaller" disks are in one larger disk, $\mathcal{D}_j \subset \mathcal{D}_0, j > 0$ but that otherwise they are pairwise disjoint. Take for the model space, the "geometric sphere world"

$$\mathcal{M} = \mathcal{D}_0 - (\mathcal{D}_1 \cup \ldots \cup \mathcal{D}_k).$$

This is the configuration space for a point robot in a punctured sphere — that is, the workspace is $\mathcal{W} = \mathcal{M}$, and $\mathcal{R} \approx E^2$ is the set of all translations of a point, so that $\mathcal{Q}(\mathcal{M}, E^2) \approx \mathcal{M}$. It can be shown that if β_j vanishes on $\partial\mathcal{D}_j$ then a function of the form φ is a navigation function on $\mathcal{M}$ when δ_l is chosen properly [9]. Implicit in the proof of this result is that more disk punctures can be accommodated in an essentially "automatic" manner. One just adds a new factor, β_{k+1}, to the recipe for β and adjusts δ_l. Thus, relatively simple adjustments to the structure of φ handle topological changes in the the configuration space. Somehow, this should not seem surprising — why?

What does seem surprising is that the same construction works in the case of a "spider robot" in a punctured sphere. That is, keeping $\mathcal{W} = \mathcal{M}$, take $\mathcal{R}$ to be the rigid placements of a locked overlapping pair of disks, $\mathcal{S}$, into $\mathcal{M}$,

$$\mathcal{R} \approx SE(\mathcal{S}, \mathcal{M})$$

so that the configuration space is a solid torus punctured by toral tubes (that may split and join as they are traced through the interior). Rimon [13] has shown that (after a bit more adjustment) the structure remains valid here too — φ is a navigation function on $\mathcal{Q}(\mathcal{M}, SE(\mathcal{S}, \mathcal{M}))$.

Moreover, recent work on potential fields for multiple robots [14] and for robotic assembly [7] suggests that the same construction remains valid when $\mathcal{W} = E^n$ and $\mathcal{R}$ is the simultaneous placement of m disks,

$$\mathcal{R} \approx \prod_{i=1}^{m} SE(\mathcal{D}_i, E^n).$$

The condition for adjusting δ_l to get a rigorous proof in the case $n = 1$ amounts to a convexity condition for φ [7], and this may offer some hint of what is going on. But note that in general, $\mathcal{Q}$ will not itself be a convex space, so there is no chance of φ satisfying a convexity property.

In each of these cases, the metric features of the objects being placed figures prominently. Each is formed from disks in some essential manner. Thus, there seems to be some geometric (rather than topological) property of $\mathcal{R}$ that is favored by φ. What property is this?

2.2.2 A Fuzzy Interpretation

Imagine building a formal syntax of scalar valued expressions in the following manner. Let the product of two functions denote their "conjunction." Represent "negation" of a function by its reciprocal. This yields a sentential logic, $\mathcal{L}$. Is there some model for $\mathcal{L}$?

One might loosely interpret the "sentence"

$$\frac{\gamma}{\beta_0 \cdot \beta_1 \cdot \ldots \cdot \beta_k}$$

to mean "move toward the zero of γ and do not move toward the zero of β_0 and do not move toward the zero of β_1 and..." and so on. This would be an anthropocentric semantics for $\mathcal{L}$. On the other hand, for some non-trivial range of model spaces, $\mathcal{M} = \mathcal{Q}(\mathcal{W}, \mathcal{R})$, the construction is a mere "shaping", δ_l, away from a navigation function φ. Since possessing the navigation property is equivalent to a closed loop plan, one might intepret this as a "robo-centric" semantics for $\mathcal{L}$.

Can these two semantics — the anthropo- and robo-centered — be formalized as models (in the sense of mathematical model theory) of $\mathcal{L}$? Are there other sentences that "mean" the same thing in both semantic systems?

3 Sensor Based Navigation Functions

This section introduces a setting wherein the tuning of diffeomorphisms and switching between symbol strings must be done online and in a coordinated manner. A (perhaps *the*) central problem in robot navigation concerns the incorporation of sensor derived information about the configuration space. Heretofore, we have assumed full a priori knowledge of $\mathcal{Q}$ and precise online information concerning $q \in \mathcal{Q}$. But real sensors provide merely filtered views of the workspace and it is a major problem to devise a means of recovering from the evolving history of sensor readings the configuration space and the robot's state therein.

There are really two hard problems. First, how can one infer configuration space properties from the highly filtered views of the workspace given by sensors? Second, how can one successfully adjust the plan in an online manner to be consistent with the sensor derived adjustments in the presumed configuration space. I speculate briefly on the manner in which navigation functions relate to the first problem and then move on to consider their role in addressing the second.

3.1 Navigation from Sensory Data

Loosely following Donald [3], we will suppose that a "sensor," s, provides reports to a "sensorium," S, concerning the classes of "perceptually equivalent" configuration space locations

$$s : \mathcal{Q} \to S,$$

so that given a report, $\sigma \in S$ we "know"

$$q \in Q_\sigma := s^{-1}(\sigma)$$

only up to an equivalence class.

3.1.1 Projections of a Known a Configuration Space

Assume complete information about Q. We will loosely follow Erdmann [4] in proposing an adaptation of navigation function methods to this situation. Since $\mathcal{R}$ is always a Lie Group, it is parallelizable [2], that is,

$$T\mathcal{R} \approx \mathcal{R} \times E^k$$

where it is understood that the Euclidean vector space E^k represents the Lie Algebra. Directions of infinitesimal motion in $\mathcal{R}$ can thus be represented (uniformly over $\mathcal{R}$) by S^{k-1} — unit ball of of E^k. Consider the "cone of progress" generated over a perceptual equivalence class, Q_σ, by a navigation function

$$\mathcal{V}_\sigma := \left\{ \frac{grad\,\varphi\,(q)}{\|grad\,\varphi\,(q)\|} : \exists q \in Q_\sigma \right\} \subset S^{k-1}$$

If $\mathcal{V}_\sigma$ is contained in a hemisphere, $\mathcal{V}_\sigma \subset \mathcal{H}^+$, then any direction in the opposing, hemisphere, $w_\sigma \in \mathcal{H}^-$ will result in progress. That is to say, the output feedback rule

$$u(\sigma) = w_\sigma$$

generates a plan: it results in a closed loop system for which φ is a Lyapunov function.

Hemispheric containment is a very strong condition. It seems to imply, for example that there be a "beacon" at the destination $q^* \in Q$ — that is to say, we would require a privileged sensor value, $\sigma^* \in S$ whose perceptual equivalence class is the singleton at the destination, $Q_{\sigma^*} = \{q^*\}$. Such restrictive assumptions need to be replaced with a filtering theory.

In general, even in linear control theory problems, output feedback is rarely sufficient to achieve stabilization, and what is really needed here is some "structural observer theory" that would use the time history of the sensor reports, $\sigma(t)$ to refine the "raw" partition induced by single measurements. That, of course, is what many researchers are presently trying to work out, and I will say no more about such efforts here.

3.1.2 Estimation From Projections of an Unknown Configuration Space

As the previous discussion shows, when a sensor provides information about a known environment, there is, in control theoretic parlance, an observer problem. When the sensor provides information about an unknown environment there arises an adaptive observer problem.

Suppose there is a parametrized family of possible workspaces, $\{\mathcal{W}_p\}_{p\in\mathcal{P}}$, into any one of which the robot, $\mathcal{R}$ may have been placed. Denote by Q^p the configuration space $Q(\mathcal{W}_p, \mathcal{R})$. There is now a different sensor map for each workspace, $s_p : Q^p \to S$. We require an estimation procedure for adjusting p by comparing the sensor reports, σ, with the estimates, $s_p(q)$. This is an "adaptive" problem in the sense that both q and p are unknown and must be each adjusted online. I will now simply presume the choice of some parameter tuning method that results in a trajectory of estimates over time, $p(t)$.

3.2 Tuning and Switching

We have just seen that sensor based navigation problems give rise to evolving views of the true configuration space, $Q^{p(t)}$ resulting from a motion in parameter space, $p : I\!R \to \mathcal{P}$, consequent upon some sensor interpretation scheme applied over the course of exploration. For some portions of the motion, we expect geometric changes in the configuration space, and for others we may expect topological changes.

3.2.1 Tuning: Adaptation to Changing Geometry

In those situations where we have succeeded in constructing them, our diffeomorphisms are defined in terms of the specific geometric features of the particular configuration space [12]. Thus for a selected p^* we may think of an associated diffeomorphism, $h_{p^*} : Q^{p^*} \to \mathcal{M}$ and, for some neighborhood of $\mathcal{N}(p^*) \subset \mathcal{P}$ one might imagine an associated family diffeomorphisms, $\{h_p\}_{p\in\mathcal{N}(p*)}$. Then a planner might "track" the changing sensory views with the navigation function sequence

$$\varphi(t,q) =: \varphi \circ h_{p(t)}(q).$$

3.2.2 Switching: Adaptation to Changing Topology

But it must be expected that local changes in the shape of the workplace (or the robot's shape) will cause topological changes in the configuration space.

Say that $(\mathcal{W}_1, \mathcal{R}_1)$ is *equivalent* to $(\mathcal{W}_2, \mathcal{R}_2)$ if $Q(\mathcal{W}_1, \mathcal{R}_1)$ is diffeomorphic to $Q(\mathcal{W}_2, \mathcal{R}_2)$. Using the Hausdorf "shape" metric, the set of all robot-workspace pairs has a natural topology. Say that a pair $(\mathcal{W}, \mathcal{R})$ is *structurally stable* if it lies in some open neighborhood of equivalent pairs; otherwise say that it is a *bifurcation pair*.

I now conjecture that the structurally stable pairs form an open dense subset of the space of pairings. This would impose a cellular partition on $\mathcal{P}$, the parameter space that summarizes the a priori knowledge of the workspace structure. Each open cell would correspond to a structurally stable class separated from its neighboring class by an empty interior bifurcation set. If so, then it is to be expected that a one-parameter family of pairings — a sequence of estimates of the workspace returned over time as the robot directs its sensors through the environment — will generally consist of intervals of equivalence punctuated by bifurcations.

Thus, there is a recognition problem and an action problem for the syntax generating automaton. Will it be possible using local clues from the sensor to deduce a bifurcation? There is a discrete sequence of models, $\{\mathcal{M}_k\}_{k\in K}$, to be determined. How might it be possible to choose the model, a model function, φ_k, a nominal geometry, p_k^* and a tuned family $\{h_p\}_{p\in\mathcal{N}(p_k^*)}$?

3.2.3 Tuning and Switching Must be Mixed Together

Topologically equivalent (smooth) configuration spaces admit the same navigation function up to diffeomorphism. Thus, we might have imagined that $\{h_p\}_{p\in\mathcal{N}(p_k^*)}$ is a smooth parametrization. Unfortunately, this may not always be possible.

In recently reported work, Hirsch and Hirsch [5] have examined the fiber bundle, $\pi : \bar{\mathcal{E}} \to \mathcal{B}$, of navigation functions to the family of two-spheres with a finite number of punctures. Recall that a two-sphere, S^2, with k punctures is a topological model for the configuration space of a (rotationally symmetric) mobile robot in a closed room with some number of obstacles. The location of the punctures defines a point in

$$\mathcal{B} = \underbrace{S^2 \times ... \times S^2}_{k\text{times}},$$

and for each $b = (b_1, ..., b_k) \in \mathcal{B}$, the fiber $\bar{\mathcal{E}}_b$ denotes the set of Morse functions that take their only minimum on b_1, their maxima at $b_2, ...b_k$ and have some number of other saddle points as well.

It shouldn't matter where the punctures are on the sphere since all the sets parametrized by $\mathcal{B}$ are diffeomorphic. Thus, one imagines the possibility of assigning to each b a distinguished member of its fiber, $\varphi_b : S^2 \to I\!R^2$ such that a smooth trajectory $b(t) \in \mathcal{B}$ lifts to a smoothly parametrized family of navigation functions, $\varphi_{b(t)}$. Apparently, this is not possible — the Hirsch $\bar{\mathcal{E}}$ admits no smooth section.

4 Conclusion: Is there an "Intelligent" Dynamics

Recently, Narendra and I have asserted that an "intelligent" controller will require the ability to switch and tune [11]. Surely, sensor based robot navigation represents an example of the need for such intelligence. There seem to be at least two significant gaps in the ideas explored by this paper.

First, as discussed in Section 3.1.2, one requires an "estimation dynamics" to move around in $\mathcal{P}$ on the basis of the reports from s_p. This may be addressed by standard algorithms or it may give rise to presently unsolved problems, depending upon the parametrization — that is, how p enters $\mathcal{Q}^p$ and, thereby, enters s_p and h_p. At any rate, there is a large existing body of theory and practice to inform such investigations [10].

Second, as discussed immediately above in Section 3.2, one requires a means of switching between models and model functions. Here, the problem seems less well delimited. There is a presumption that the manner in which one equivalence class of $(\mathcal{W}, \mathcal{R})$ pairs bifurcates into another class may be captured by some discrete dynamics. This is indeed presumptuous, but it is perhaps not impossible to imagine for restricted classes of problems. One would then need to understand how the "dynamics" of configuration space bifurcations could be captured in the "syntax" generating automaton that produces the sequence φ_k.

References

[1] Ralph Abraham and Jerrold E. Marsden. *Foundations of Mechanics.* Benjamin/Cummings, Reading, MA, 1978.

[2] William M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry.* Academic Press, New York, 1975.

[3] Bruce Donald and Jim Jennings. Constructive recognizability for task-directed robot programming. In *IEEE Int. Conf. Rob. Aut.*, 1992.

[4] Michael Erdmann. Towards task-level planning: Action based sensor design. Technical Report CMU-CS-92-116, Carnegie Mellon University, Pittsburgh, PS, Feb. 1992.

[5] Michael Hirsch and Morris Hirsch. On the nonexistence of certain morse functions. Talk pre-

sented at the 1993 Mid-West Dynamical Systems Conference, Oct. 1993.

[6] D. E. Koditschek. Task encoding: Toward a scientific paradigm for robot planning and control. *Journal of Robotics and Autonomous Systems*, 9:5–39, 1992.

[7] D. E. Koditschek. An approach to autonomous robot assembly. *Robotica*, (to appear), 1994.

[8] Daniel E. Koditschek. The Application of Total Energy as a Lyapunov Function for Mechanical Control Systems. In J. Marsden, Krishnaprasad, and J. Simo, editors, *Control Theory and Multibody Systems*, volume 97, pages 131–158. AMS Series in Contemporary Mathematics, 1989.

[9] Daniel E. Koditschek and Elon Rimon. Robot navigation functions on manifolds with boundary. *Advances in Applied Mathematics*, 11:412–442, 1990.

[10] K. S. Narendra and A. Annaswamy. *Stable Adaptive Systems.* Prentice-Hall, NY, 1988.

[11] K. S. Narendra and D. E. Koditschek. Intelligent control systems design. (Funded Proposal: National Science Foundation), 1991.

[12] E. Rimon and D. E. Koditschek. The construction of analytic diffeomorphisms for exact robot navigation on star worlds. *Transactions of the American Mathematical Society*, 327(1):71–115, Sep 1991.

[13] Elon Rimon. A navigation function for a simple rigid body. In *Proc. IEEE Int. Conf. Rob. and Aut.*, pages 2–7, Sacramento, CA, April 1991. IEEE Computer Society.

[14] Louis L. Whitcomb, Alfred Rizzi, and Daniel E. Koditschek. Comparative experiments with a new adaptive controller for robot arms. In *Proc. IEEE Int. Conf. Rob. and Aut.*, pages 2–7, Sacramento, CA, April 1991. IEEE Computer Society.