Considerations on Trajectory Planning Models in Automated Driving

Miriam Ruf

Fraunhofer IOSB,
Karlsruhe, Germany
miriam.ruf@iosb.fraunhofer.de

Technical Report IES-2015-04

Abstract: The SPARC concept for fully-automated driving was introduced in 2012 as a way to overcome the highly discretized maneuver planning approaches that had previously dominated the DARPA Grand and Urban Challenges. The aim was to define a single planner that is responsible for all forward driving tasks, including intersections, highways and pre-crash situations; that is based on a sound statistical framework including prediction and sensing uncertainties; and that can trade off between goals like safety, comfort and efficiency in a unified and transparent way. Since then, several similar approaches have been presented, most notably the Bertha Benz drive of 2013, in which the 100 km distance between Mannheim and Pforzheim was covered autonomously in a Mercedes–Benz S-500 Intelligent Drive. During the same time, the SPARC approach has been refined in several respects, including its vehicle models and optimization techniques. This technical report provides an overview of the avenues explored to motivate the current form of the SPARC concept, based on frequently-asked questions that arose during the various technical presentations of the concept.

1 Background and Motivation of SPARC

The SPARC (Situation Prediction and Reacting Control) concept (introduced in [Zie12, RZR+14]) was created within the V50 project aimed at developing solutions for fully-automated inner-city driving; the specific task assigned to Fraunhofer IOSB was to provide two modules – situation detection and maneuver planning. The project structure is shown in Fig. 1.1. The approach was chosen to overcome drawbacks with highly discretized approaches that were commonly used previously to this and, for example, dominated the DARPA Grand

and Urban Challenges (cf. [RZR+14]). Examples include the use of discrete velocity profiles for the ego vehicle [UAB+08], discrete "situation states" in which the ego vehicle can be [MBB+08, CSA+09], or arbitrating the responsibility for the driving task dynamically between various dedicated planning systems [UAB+08, MBB+08, WSM09, BBF+08]. The goal of SPARC was to express all driving tasks in a unified ("holistic") way, governed by a universal set of rules that remains constant from parking to pre-crash, and from urban intersections to interstate highways.

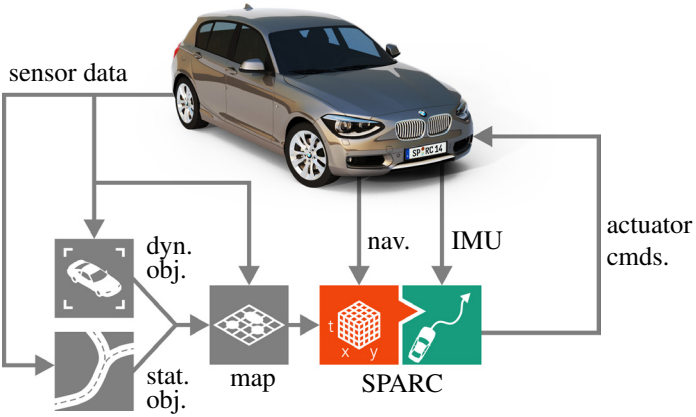


Figure 1.1: Overview of the V50 project. Sensor data is processed to detect dynamic objects (cars, pedestrians, cyclists, ...) and static objects (walls, signs, traffic lights, roads, road markings, ...) in the environment. The information is combined in a dynamic map, which presents the basis for SPARC. The Situation Prediction (SP, red) stage predicts the future development of the scene, determines applicable traffic rules and sets waypoints based on navigation instructions. The Reaction Control (RC, green) stage uses this information to compute an optimal trajectory and execute it using the actuators (steering, acceleration, lights, ...).

The SPARC module receives information from a *dynamic map* of tracked static and dynamic objects and features detected in the vicinity of the ego vehicle. The *Situation Prediction* (SP) stage predicts the behavior of dynamic objects into the near future (5–10 seconds), and estimates for each object at every (future) time step the severity of a possible collision (cf. [RZW+14a]). It also determines applicable traffic rules (such as speed limits, overtaking rules or stop signs) and

assigns penalties to their violation. Finally it uses navigation instructions to set coarse waypoints for the ego vehicle. All of this information is passed to the *Reaction Control* (RC) stage, which uses it to plan an optimal trajectory for the ego vehicle to the proximity of the waypoints, by minimizing collision risks, traffic rule violations, passenger inconvenience, fuel consumption and wear of vehicle parts ([RZR+14, RZW+15]). The earliest control inputs of the optimized maneuver are passed to the vehicle actuators.

This process is repeated at a rate of 10 Hz, so that the planning horizon of the trajectory does not just cover the gap between updates, but assures that the currently executed actuator commands are part of a long-term plan that is considered safe by the currently best prediction. Any unforeseen developments will be accounted for in the following update.

Should the environment predictions not be updated due to a significant system failure (e.g. sensor blackout or crash of the SP stage), during each cycle, the RC stage also computes a fail-safe emergency trajectory that is also based on the same predictions but will not lead the ego vehicle to the next waypoint, but to a safe place on the side of the road (see [RZW+15] for details).

Since the original publications [Zie12, RZR+14], several similar approaches have been presented, most notably [ZBDS14], which successfully completed the fully-automated 100 km Bertha-Benz drive from Mannheim to Pforzheim in 2013. Despite using equivalent models with a similar motivation, the SPARC approach is unique in several design decisions that have aroused some discussion during previous presentations, and shall thus be motivated in the following sections with particular respect to possible alternatives.

2 Model Decisions in SPARC

This section will motivate key decisions and discuss alternatives related to two main areas of SPARC: The problem formulation over predicted, expected risks traded off against comfort, ecology and traffic rule compliance, integrated in an Euler-Lagrange model (Sec. 2.1); and the optimization of this Euler-Lagrange model through a Hidden Markov Model, instead of an iterative solver (Sec. 2.2).

2.1 Use of Expected Risks and the Euler-Lagrange Model

The most basic motivation of the SPARC approach was to define an evaluation *functional* $\mathcal{H}[\xi]$ that maps a given trajectory $\xi(t)$ onto non-negative real numbers,

such that the most desired functions $\overset{*}{\xi}(t)$ are recognized by having assigned the least $\mathcal{H}[\xi^*]$. This section will present the choice of $\mathcal{H}$, the alternatives considered, and the rationale behind the eventual choice.

To embed SPARC into well-developed theoretical frameworks, two key features were considered relevant: Additivity and a stochastic foundation. These two individual properties, along with the combination as *additive stochastic models*, will be motivated and illustrated in this section.

Additivity An evaluation functional based exclusively on a function H whose value is additive

$$\mathcal{H}[\xi] = \int_{t_{\text{start}}}^{t_{\text{end}}} \mathrm{d}t \; H(\xi(t), \dot{\xi}(t), \ddot{\xi}(t), ..., \tfrac{\mathrm{d}^n}{\mathrm{d}t^n}\xi(t), t) \tag{2.1}$$

restricts the expressiveness of $\mathcal{H}$ considerably. Its application is limited to $\xi \in C^n$, and it is impossible to model long-term dependencies, such as requiring that ξ have exactly three roots. It is also impossible to evaluate a dynamic range of derivatives, such as requiring that ξ have an arbitrarily great but even number of nonzero derivatives at each point.

These limitations however bring about a significant advantage: It originally arose from the physical study of particle trajectories, both for photons and matter, and therefore has been studied extensively. A particularly relevant result is the Euler-Lagrange equation [VB10], which provides an efficiently computable necessary criterion for the optimality of a trajectory $\xi^{\circledast}$, namely

$$\delta_\xi \mathcal{H} := \frac{\partial H}{\partial \xi} - \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial H}{\partial \dot{\xi}} + ... + (-1)^n \frac{\mathrm{d}^n}{\mathrm{d}t^n}\frac{\partial H}{\partial \frac{\mathrm{d}^n}{\mathrm{d}t^n}\xi}, \tag{2.2}$$

such that, if $\xi^{\circledast}$ is optimal, $\delta_\xi \mathcal{H}\big|_{\xi^{\circledast}} \equiv 0$, which allows to find solutions analytically. In numerical solutions of (2.1), where ξ is a vector in $\mathbb{R}^{\#t}$ over $\#t$ discrete time steps, $\delta_\xi \mathcal{H}$ takes the role of a gradient (in fact, [ZRR+15] shows that (2.2) can be derived directly as a gradient) that can be used to iteratively optimize from an "initial guess" $\xi^{(0)}$ towards a local optimum ξ^*. At the same time, this formulation of $\mathcal{H}$ allows to cover a wide range of optimality criteria that arise in (automated) driving situations, namely

- Safety, by penalizing combinations of ξ and t that represent a possible future collision with another traffic participant, or lane departures.
- Comfort and ecology, by penalizing strong accelerations $\ddot{\xi}$ and jerks $\dddot{\xi}$.
- Traffic rule compliance, e.g. by penalizing exceeding speed limits via ξ.

Furthermore, [RZW+14b] shows that a significant number of vehicle parameters, such as wheel speeds and steering wheel angles, can be analytically expressed in terms of derivatives, and are thus available for use in H. These parameters can be used inside an Euler-Lagrange model (ELM) if lateral and longitudinal wheel slip are not considered.[1] Simulation results presented therein show, that even under highly dynamic conditions, these simplifications (necessary to make the vehicle model compatible with the ELM) provide a good approximation to the actual vehicle behavior (cf. Fig. 2.1).

Additionally, various types of *boundary conditions* can be added to express properties of ξ at t_{start} and t_{end}, in particular constraining ξ to the current vehicle position at t_{start} and penalizing deviations from a goal position at t_{end}. The only key limitation that remains is that all time instances of ξ must be evaluated independently through H, so that non-local dependencies between any values attained by ξ cannot be modeled.

Stochastical Models Most established mathematical models in artificial intelligence-related fields are equipped with a stochastic interpretation. While their computation often relies heavily on simplifications to preserve tractability (such as the assumption of conditional independence), all simplifications can be made explicit, and thus can be validated or falsified in comparision with real-world statistics. The stochastic interpretation also provides a convenient way of combining different models on a well-understood basis. On the other hand, models that circumvent stochastic modeling are usually less easily analyzed in terms of assumptions and quality of results, and thus less easily combined. For fully-automated driving, the main aspects to be combined through stochastical modeling are:

[1] Estimating the current and future grip of a vehicle under practical driving conditions is considered a highly difficult problem. Therefore, any predicted grip value would likely be speculative anyway.

- Sensor uncertainties (such as noise, blur and occlusion)
- Sensor processing uncertainties (e.g., remaining classification ambiguity)
- Prediction uncertainty (such as uncertain lane changes in other cars)
- Actuator uncertainty (such as unknown friction for a braking maneuver)

Stochastical models must thus be found that both allow to express these uncertainties and fit the limitations of the ELM (2.1).

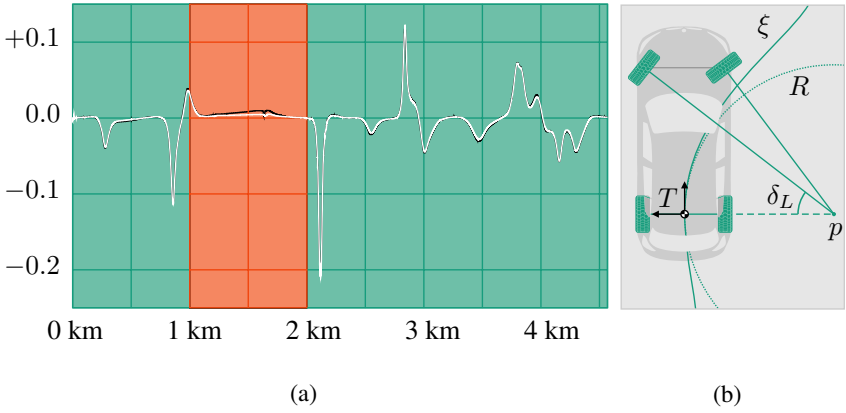


Figure 2.1: The C^2 model presented in [RZW+14b]. (a) Estimated steering wheel angles (white) compared to steering wheel angles obtained in a state-of-the-art physical simulation of the Hockenheimring race track (black). The red section shows deviations during to lateral slip in a bend driven at 170 km/h. Everywhere else, the estimations of the C^2 model match the simulation closely. (b) The geometry of the C^2 model is based on the tangent coordinate system T of the trajectory ξ, and its circle of curvature R centered at p. This allows to describe a multi-track vehicle with Ackermann steering; the left wheel angle δ_L is indicated.

Additive Stochastic Models The two most prominent stochastic models that aim or allow for additivity are log-linear models and independent expected values, both of which rely on assumptions of stochastic independence. We will introduce stochastic independence, and briefly discuss log-linear models, to motivate why expected values were instead preferred for the use in SPARC.

Probabilities that are stochastically independent can be multiplied to obtain the joint probability. A common example is the cast of two dice. Both dice have a probability $p(d=n) = 1/6,\ n \in \{1, ..., 6\}$ of rolling an "n". The probability of one dice rolling a 1, the other a 6, is given as $p(d_1 = 1 \wedge d_2 = 6) = p(d_1 = 1) \cdot p(d_2 = 6) = 1/36$ by virtue of the assumption of stochastic independence; the cast of either dice is not expected to influence the other. A counterexample is rolling a *single* dice *once*, and giving the probability of the *same* dice producing both values in a single cast ($p(d_1 = 1 \wedge d_1 = 6)$). While both individual probabilities remain $1/6$, their joint probability is 0 – the results are not independent, as one will strictly rule out the other.

More often than not in practical applications, stochastic independence is taken as a convenient simplification rather than the true system behavior. For example, motion tracking algorithms may describe their next target motions as independent from previously observed motions, while in most cases the targets are *known* to not act randomly. However, this assumption allows each timestep to be treated in a uniform fashion, and the corresponding models can be more easily parametrized since complex and long-term behavior patterns can be ignored, and individual transition probabilities can be multiplied repeatedly over time to obtain the probability distribution of future locations.

Log-linear models are based on the realization that this multiplicative accumulation corresponds to an additive accumulation in the space of logarithmic proabilities:

$$\log\left(\prod_{n \in N} p_n\right) = \sum_{n \in N} \log(p_n)$$

Thus by taking the logarithm of each probability, a quantity is obtained that can be summed while retaining the stochastic interpretation in the result. The stochastic model is said to be *log-linear*. In this way, models that rely exclusively on probabilities can be subject to linear optimization, usually to *maximize probabilities*.

However, many applications cannot be adequately modeled through probabilities. It stands to reason that automotive traffic (or locomotion in general) is not intuitively a pure maximation or minimization of probabilities. While occasionally argued, a desirable trajectory is not that which minimizes the probability of

an accident, since this trajectory would in most cases remain stationary forever and not even attempt to reach a faraway destination. We can further conceive the case of a pre-crash situation, where the driver can choose between a certain ($p_1 = 1$) rear-end collision with the car in front of him at low speed, or taking an evasive maneuver to the opposite lane and stopping the car amid oncoming traffic. The probability of an accident is "worst" for the rear-end collision, yet this option will usually be preferred to the lower but more severe risk of colliding with oncoming cars at high speed.

The key alternative thus appears to be *risk*, used here as the *expected value of detriment*, however the latter may be defined. For illustration purposes only we can consider the *detriment* H of a collision as given purely by the costs of the damage, which for the low-speed rear-end collision may be $H_1 = \$5\,000$, while a high-speed head-on collision may cost $H_2 = \$100\,000$. Now considering a $p_2 = 1/10$ chance of colliding with oncoming traffic, we have an expected cost of $p_1 \cdot H_1 = \$5\,000$ for the rear-end collision (which is, as stated, certain), and $p_2 \cdot H_2 = \$10\,000$ for the head-on collision, which is considerably higher.

Let us assume that in this case the driver chooses to accept the rear-end collision, but finds himself in almost the same situation on the same road one month later. The collision costs remain the same, but this time he evades to the opposite lane.

Now we can expect the total damage arising from these two situations to be $p_1 \cdot H_1 + p_2 \cdot H_2 = \$15\,000$. We realize that the expected costs were *added*, based on the intuition that the outcomes of the two accidents are independent. But are they?

Not strictly: Had the driver decided differently, and collided with an oncoming car in the first case, the damaged car is less likely to be repaired by then, and thus the collision risk is lower since one possible collision partner is missing. The outcomes are not independent, but considering the sheer number of possible collision partners, the dependence is slight and difficult to model. Whether or not expected values can be added thus also relies on stochastic independence, and whether or not possible dependencies (as in this case) must be included in the model depends on an understanding of the model.

What is gained by modeling with expected values, however, is the inclusion of two parameters – the detriment and the probability – which provides a bridge between pure probabilities and their real-world implications. Once a unit for detriment (here the costs in \$) has been set, stochastic events (such as collisions) and deterministic events (such as detriment from high fuel consumption or uncomfortable accelerations) can be directly compared: What makes human drivers accept risks that are inherent to participating in road traffic is a trade-off between

the risks and the gains. For example, human drivers pass pedestrians on the sidewalk even if there is a risk of them jumping onto the road before them in the last second; the probability is considered extremely low and the detriment of braking to walking speed before passing any pedestrian is considerable in terms of travel time and fuel consumption.

Using expected values, these trade-offs, that are implicit to human driving can be rendered explicit and quantitative for automated driving. Details on how the predicted occupancy probabilities of other traffic participants are obtained can be found in [RZW+14a].

2.2 Local vs. Global Optimization

Given a functional based on the previous criteria, the optimization goal

$$\xi^* = \arg\min_{\xi} H[\xi]$$

uniquely identifies a trajectory or a set of trajectories which are, based on the model, of indistinguishable quality. Obtaining a minimal element, however, is not trivial. The most commonly used method for optimization, an iterative descent such as a gradient descent or extensions of Newton's method (SQP, BFGS, L-BFGS), are risky to apply in automated driving, for reasons that will be detailed in this section based on [RZW+15]. An alternative model is outlined that mitigates these risks while retaining real-time capability.

2.2.1 Risks Associated to Iterative Optimization

Iterative solvers start from an "initial guess" $\xi^{(0)}$, and apply an optimization step $\xi^{(n-1)} \mapsto \xi^{(n)}$, usually with the goal that $H[\xi^{(n)}] < H[\xi^{(n-1)}]$. The process terminates based on criteria for $\xi^{(n)}$, and possibly n, $\xi^{(n-1)}$ and other properties. Common termination criteria are

$$\begin{aligned}
H[\xi^{(n)}] &< T_A && \text{(sufficiently low value)} \\
|H[\xi^{(n)}] - H[\xi^{(n-1)}]| &< T_B && \text{(sufficiently small value step)} \\
\left.\frac{\mathrm{d}H}{\mathrm{d}\xi}\right|_{\xi^{(n)}} &< T_C && \text{(sufficiently small gradient)} \\
n &> N && \text{(computation time exhausted)}
\end{aligned}$$

The main problems associated with this are already apparent from the above formulation. Firstly, none of the above uses truly *global* properties, i.e. the relation of $\xi^{(n)}$ to arbitrary other members of the function space. At best, a small subset of dimension 1 is evaluated. Thus, the process can terminate in a stationary point or local optimum that is far from the global optimum. Secondly, not even local convergence can be strictly measured, but is given in terms of thresholds. Thus, the process can terminate even in points that are not stationary, but merely too slightly sloped. Thirdly, the process can terminate after a fixed number N of iterations disregarding *any* properties that $\xi^{(N)}$ might hold – or run an arbitrarily long time.[2]

In total, it can be said that properties of the obtained solution $\tilde{\xi}^*$ are difficult to predict. Beside the problem structure (which is the only desirable influence), they depend significantly on the initial guess and the optimization step; in many cases, these even dominate the solution. The influence is generally *not* intuitive, as $\xi^{(0)}$ will not necessarily converge to (e.g.) the closest local optimum, or the optimum towards which the first gradient points.

This uncertainty about the quality of the optimization result, combined with the possibility of strongly varying optimization times, suggests a high risk of applying local iterative optimization to high-risk real-time applications such as automated driving. Examples of all of the above claims, arising in simple, everyday traffic situations, will be given in detail in an upcoming publication.

2.2.2 Strategies to Improve Iterated Optimization

One method considered as a candidate for the generation of initial guesses is the partition of the planning space by Voronoi lines between local risk maxima.

As used here, a (Euclidean) Voronoi diagram on a space $\Sigma \subseteq \mathbb{R}^n$, based on a finite set $P \subset \Sigma$ of points, partitions Σ into *cells* C such that $|P| = |C|$,

$$\sigma \in c_i \quad \Longleftarrow \quad p_i \in \arg\min_{p \in P} \|p - \sigma\|_2$$

which implies

$$\bigcup_{c \in C} c = \Sigma$$

[2] It should also be noted that divergence can take many different forms, from infinite loops over increasingly bad solutions up to chaotic behavior. It is thus difficult to detect and even harder to resolve.

but also provides

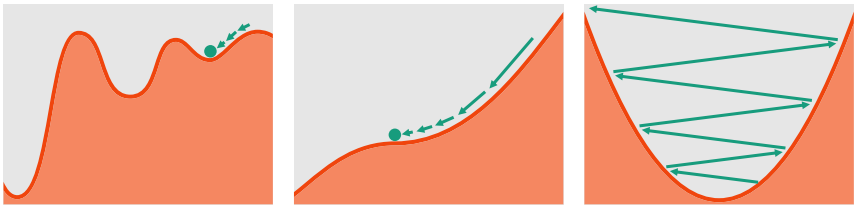
$$V = \bigcup_{c',c'' \in C} c' \cap c''$$

such that for $|P| > 1$ and a convex Σ it holds[3] that $V \neq \emptyset$ and $\dim(V) = \dim(\Sigma) - 1$. We also introduce a finite set

$$I = \{v \in V \mid \exists c', c'', c''' \in C : v = c' \cap c'' \cap c'''\}$$

of *intersections* in V, i.e. points at which at least three cells intersect.

For a two-dimensional risk field $H_R : X \times T \to \mathbb{R}$, a Voronoi diagram between the local maxima of H_R provides a one-dimensional V, which resembles a lattice of lines over $X \times T$ between the local maxima, as shown in Fig. 2.3. As these lines avoid local maxima of H_R, they can be used to provide initial guesses for trajectories $\xi : T \to X$. A straightforward approach would be to perform a graph search by using the elements of I as nodes in a graph G, with edges denoting direct connections in V. Then paths in G can be used to hint at initial guesses for ξ^*.



(a) Local optimum (b) Stationary point (c) Divergence

Figure 2.2: Increasingly undesirable convergence behaviors of an iterative solver. (a) The solver converges to a local optimum that is considerably worse than the global optimum. (b) The solver converges to a stationary point instead of any optimum. (c) The solver does not converge at all. Each case can occur under realistic conditions; in real-time high-risk scenarios, the consequences can be serious.

[3] Proof: As $|P| > 1$ (and finite) then $\forall p_1 \in \Sigma\ \exists p_2 \in \Sigma$ s.t. $p_2 \in \arg\min_{p \in P} \|p_1 - p\|_2$. Then, as Σ is convex, $\exists \sigma_m = (p_1 + p_2)/2 \in \Sigma$; also $\arg\min_{p \in P} \|p - \sigma_m\|_2 \supseteq \{p_1, p_2\}$. Thus $\sigma_m \in V$.

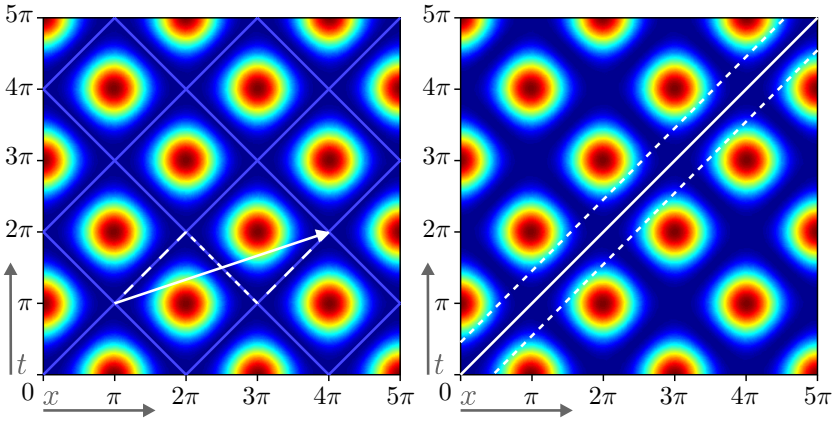


Figure 2.3: Inabilities in the Voronoi enumerating the local optima. (a) V (light blue) and a path through G (white dashed, dotted) using an edge backward through time (dotted) that would be omitted. No valid connection in G between source and goal point exists, even though there is a potential local optimum (arrow). (b) Effect of some $\alpha > 0$ (weighting factor of some goal speed) limiting even all *local* optima to within the dashed band of width $2d$. The Voronoi diagram cannot account for this limitation.

This, however, entails a series of problems:

- V does not generally include the point in $X \times T$ that is considered the initial position of the vehicle $\xi(0)$. Thus a new connection between $\xi(0)$ and V must be created that is difficult to motivate within the model.
- Paths in G can move backwards through time. These cases can and must be excluded, but this issue hints at a deeper problem:
- Paths in G are based on geometric considerations that consider the two space dimensions as interchangeable. Therefore, temporal distances and spatial distances are considered equal, which requires the choice of a conversion factor. This factor determines the solution (cf. Fig. 2.5) even though it is unlikely to stem from any physical motivation. In this case, the number of free parameters, that are not motivated through the problem, increases. This is generally undesirable, as these parameters are not easily adapted to changes in the problem.

- Paths in G depend exclusively on the structure of H_R but have generally no relation to, for example, the dynamical model of the vehicle or target positions, speeds and accelerations. Thus, in particular the notion of locality can differ significantly: The basically possible paths through V (moving forward through time, respecting speed limits, ...) do *not* enumerate the local optima of $\mathcal{H}[\xi]$ since $\mathcal{H}[\xi]$ has a (significantly) higher dimension than H_R, and this dimension is highly relevant, as can be seen in the following examples:

  - Additional local optima can arise from penalties on, e.g., speeds, that cannot be derived from H_R. Fig. 2.3 gives an example where significant local optima would be missed this way.
  - When paths "backwards in time" are simply removed, possible local optima are ignored, since V does *not* consider the "best" path between the optima in terms of the optimization functional, but the one with the highest space-time distance regardless of physical possibility. Consider the risk field[4]

    $$H_R(x,t) = (\cos(x) - \cos(t))^2\,, \quad x \in [0, 5\pi], t \in [0, 5\pi] \tag{2.3}$$

    Then $i_1 = [\pi, \pi]^{\mathrm{T}}, i_2 = [2\pi, 2\pi]^{\mathrm{T}}, i_3 = [3\pi, \pi]^{\mathrm{T}}, i_4 = [4\pi, 2\pi]^{\mathrm{T}}$ with $i_1, i_2, i_3, i_4 \in I$, but no edge between i_1 and i_4. There exists a path $i_1 \to i_2 \to i_3 \to i_4$, but this is not physically possible since $i_2 \to i_3$ moves backwards in time. It would thus naïvely be discarded, but (unlike in other similar cases, which are discarded rightfully) there exists a potentially locally optimal direct path connecting i_1 and i_4 that is physically possible and avoids all local maxima (albeit not at the furthest possible distance).
  - It is even possible that the number of local optima is *lower* than the paths through V. Consider again Fig. 2.3 and H_R as in (2.3), combined with the Lagrangian

    $$H(\xi, \dot{\xi}, t) = H_R(\xi, t) + \alpha \cdot (\dot{\xi} - 1)^2, \quad \alpha > 0$$

    which penalizes, in addition to H_R, deviations from the goal speed of $\dot{\xi} = 1$, weighted by α. Analytically we obtain the Euler-Lagrange

[4] And do notice that the lack of physical dimensions is not the fault of the example, but rather symptomatic of what a purely geometric model allows and, in a sense, requires.

equation as

$$\underbrace{2\sin(\xi)\big(\cos(t)-\cos(\xi)\big)}_{A} - \underbrace{\alpha\cdot\ddot{\xi}}_{B} \equiv 0 \tag{2.4}$$

which is, for the boundary conditions $\xi(0)=0,\ \dot{\xi}(0)=1,\ \xi(5\pi)=5\pi$, trivially satisfied for $\xi^*(t) = t$, a global optimum regardless of the choice of α, as $\mathcal{H} \geq 0$ and here $\mathcal{H} = 0$. Other local optima may exist that also satisfy (2.4). However there exists some α_T such that for any $\alpha > \alpha_\mathrm{T}$, all local optima but the global optimum vanish. This can be proven by contradiction.

Assume there exists some $\xi^{\circledast} \not\equiv \xi^*$ such that (2.4) is satisfied, so there exists $t \in T$ and some $d \neq 0$ such that $\xi^{\circledast}(t) - \xi^*(t) = d$.

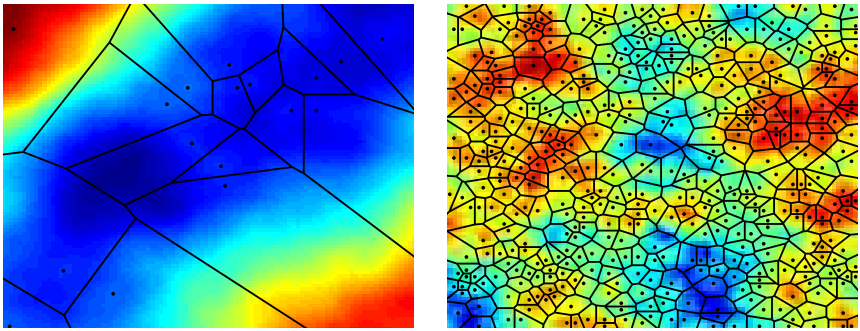
Reaching a given d within T from the fixed starting point requires an acceleration of $|\ddot{\xi}^{\circledast}(t)| > |4d/25\pi^2|$ at some instant t,[5] and thus, in (2.4), $|\alpha 4d/25\pi^2| < |B|$, presenting a lower bound for $|B|$ given α and d. In turn, A of (2.4) lies strictly within $[-3\sqrt{3}/2, +3\sqrt{3}/2]$. For $A - B$ to vanish, it is thus necessary that

$$\min|B| \leq \max|A| \Rightarrow \left|\alpha\frac{4d}{25\pi^2}\right| \leq \left|\frac{3\sqrt{3}}{2}\right| \Rightarrow |d| \leq \frac{75\sqrt{3}\pi^2}{8\alpha}$$

Intuitively, to reach a larger d (given α), there exists at least one $B(t)$ so great that no position gradient $A(\xi, t)$ can outweigh it to satisfy (2.4). Thus for large α, $|d|$ can be forced arbitrarily small, limiting any $\xi^{\circledast}$ to $|\xi^* - \xi^{\circledast}| \leq d$. Solutions outside this band are impossible, but present valid paths in G, as it is based exclusively on H_R and does not vary with α. Thus, while the choice of α can reduce the number of local optima down to 1, the number of paths in G is invariant to α.

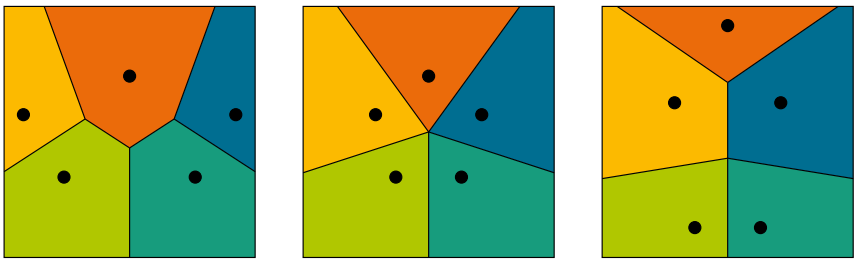
There is no obvious way to remedy these issues: The practical disadvantages of the Voronoi approach lie in the fact that it treats space and time as equivalent (and not in a relativistic sense) and cannot account for vehicle dynamics – which are a significant part of the topology of the trajectory space. Extending the dimensionality of V so that it is not based on H_R alone, but on the abstract trajectory space Ξ, is generally possible; however, building a Voronoi diagram in Ξ requires

[5] In fact, only when applied during *all* t this acceleration achieves d; and only at t_{end}, not the true goal point. Thus the inequality marks no *tight* lower bound, just some definitive lower bound. Any valid $\xi^{\circledast}$ visiting $\xi^{\circledast}(t') = t' \pm d$ and returning to $\xi(5\pi) = 5\pi$ would even require a higher acceleration.



(a) $\gamma = 2.5$ (b) $\gamma = 1.5$

Figure 2.4: Brownian noise fields with different power spectrum distributions $S(\omega) \propto \omega^{-\gamma}$ (color map), local maxima (dots) and corresponding V (lines). Assuming (as used throughout this paper) the current position to be the lower left corner, it must be noted that V does not include this point. Next, in (a), V does not provide a path to the upper right corner even though it is intuitively desirable. In (b), V is a highly complex structure including many intersections and many paths along high H_R values. A valuable reduction of complexity with respect to the original pixel grid is not apparent.



(a) (b) (c)

Figure 2.5: Voronoi diagrams for different aspect ratios between the horizontal and the vertical axis. It can be seen that the resulting topology differs in each case. If the axes represent different physical quantities, as time and space, this means that the solution is not invariant to the choice of units.

finding the local maxima, which corresponds to enumerating all *worst* trajectories in Ξ. If this initial step was efficiently possible, one could simply reverse the problem to enumerate all the *best* trajectories, and pick the global optimum directly, without the additional step of creating and searching a Voronoi diagram.

On the whole, while the paths through the Voronoi diagram certainly correlate with the true local optima, they provide neither an upper nor a lower bound. It was shown that the assumptions governing the Voronoi solution are not sound within the original model; therefore, extensive post-processing is necessary to make the Voronoi solutions even just valid *candidates* for the original model; in particular, a significant part of solutions must be discarded as they are physically impossible. Given all its inaccuracy, setting up and searching the Voronoi graph is anything but simple for complex environments (see Fig. 2.4). While eliminating the free parameter of $\xi^{(0)}$ (the "initial guess"), the Voronoi diagram introduces at least one new free parameter (the relative scale of time and space) that has an even less intuitive effect on the solution; further free parameters lie in how the obtained solutions are discarded or transformed into valid solutions of the ELM. For these reasons, the Voronoi approach was abandoned in favor of a solution that would both retain the key model features throughout the solution process (as, for example, the distinction between time and space), and that would provide clear algorithmic properties (computational effort and quality of results) to be compared with the previous ELM.

2.2.3 Global Optimization

The previous considerations suggest that iterative optimization is generally not able to guarantee the quality of a solution, or that it is found in limited time. Any obtained solution depends critically on parameters that cannot be determined by the problem understanding alone.

In this section, an approach to *global* optimization that provides *global* optima in finite time to the given problem will be presented. It can help address the issues with iterative optimization in two ways. First of all, the obtained global optimum can be used directly in maneuver planning. However, global optimization of sufficiently general problems comes at a price in computational effort, that may exceed either real-time constraints or acceptable hardware costs. Should global optimization prove too costly for real-time series integration (which has not been established yet, since this section will shed light on significant parallelization potential), global optimization can still provide benchmarks for iterative solvers, in the sense that previously, the true global solution of such a problem was not

known at all, and thus the evaluation of iterative solvers had to rely on comparing different iterative solutions to each other.

To address the named pitfalls of iterative optimization without abandoning key properties of the original model, an approach based on [ZRR+15] is chosen: The ELM is transformed into an equivalent Hidden Markov Model (HMM) such that, given a sufficiently fine discretization,

1. Globally optimal trajectories of the ELM correspond directly to the globally optimal state sequence attained by decoding the HMM through the Viterbi algorithm.
2. Values of $H[\xi]$ are provided a one-to-one mapping to sequence probabilities, such that in particular $H[\xi^*]$ can be obtained.

[ZRR+15] shows that an ELM, whose Lagrangian can be separated into

$$H(\xi, \dot{\xi}, t) = H_1(x, t) + H_2(x, \dot{x}),$$

can be transformed into an equivalent HMM by choosing sufficiently fine discretizations $X' = \{x_{\min}, x_{\min} + \Delta x, x_{\min} + 2\Delta x, ..., x_{\min} + n\Delta x\}$ $(x_1 \in X)$ and $T' = \{t_{\text{start}}, t_{\text{start}} + \Delta t, t_{\text{start}} + 2\Delta t, ..., t_{\text{start}} + m\Delta t\}$ and choosing emission probabilities $p(\sigma_\tau | x_2)$ and transition probabilities $p(x_2 \leftarrow x_1 | x_1)$ as follows

$$p(\sigma_\tau | x_2) = \frac{1}{Z_1} \cdot \exp(-H_1(x_2, \tau \cdot \Delta t + t_{\text{start}}))$$
$$p(x_2 \leftarrow x_1 | x_1) = \frac{1}{Z_2} \cdot \exp(-H_2(x_2, \tfrac{x_2 - x_1}{\Delta t}))$$

where the normalization constants Z_1 and Z_2 are uniquely defined. The correspondence in [ZRR+15] extends to more general cases, but the above formulation was found sufficient to represent the SPARC approach for generalized coordinates including first-order derivatives. This transformation allows to apply the Viterbi algorithm, which determines the optimal sequence of states in X' over time T' within a fixed space of memory and in a fixed number of time steps:

$$\mathsf{VITERBI} \in \mathsf{TIME}(T' \cdot |X'|^2) \cap \mathsf{SPACE}(T' \cdot |X'|).$$

This can be contrasted with the computational complexity of iteratively solving the original ELM with a gradient descent (EL^1) or a variant of SQP using a full Hessian (EL^2), which can be written as

$$\mathsf{EL}^1 \in \mathsf{SPACE}(T') \quad \text{and} \quad \mathsf{EL}^2 \in \mathsf{SPACE}(T'^2),$$

if convergence occurs – however there is no limit on the TIME complexity, as arbitrarily many iterations can be required until then, and divergence (an infinite number of steps) may occur as well.

So, the HMM formulation addresses several key issues of the iterative optimization, namely:

- The HMM formulation does not require an "initial guess", an iteration step function $\xi^{(n)} \mapsto \xi^{(n+1)}$ or termination criteria as presented in Sec. 2.2.1. Their role as free parameters is replaced by the choice of a discretization Δx, which is considered considerably more intuitive in its results (cf. [ZRR$^+$15, RZW$^+$15]).
- The HMM solution through the Viterbi algorithm cannot diverge or require an arbitrary number of computation steps; instead, the solution is known to be found after a fixed number of computations.
- The Viterbi algorithm cannot get stuck in undesirable local optima. Its result is always the global optimum given the discretization, which does not directly equal the continuous global optimum, but is considerably less sensitive to parameters and the shape of the underlying problem $\mathcal{H}$.

Despite this, it can be said that on average, the HMM optimization requires more computations to finish than a gradient descent or SQP, and it scales particularly badly with the state space X', such that optimizing a pure two-dimensional path can already be prohibitively expensive ([RZW$^+$15] provides an adequate choice of X' that limits the computational effort while retaining sufficient expressiveness to model a multilane road).

However, comparing the number of necessary computations can be misleading, as parallelization can be a significant factor. For the ELM, the parallelization dimension is T', since for a vector in $R^{|T'|}$, each component of the gradient (2.2) can be computed in parallel to all others. However, the number of iterations (the critical factor of the iterative solver) must always be computed in sequence (for obvious reasons), so that the classical ELM approach can only benefit to a limited extent from parallel capabilities. For the HMM on the other hand, the situation is entirely different. The bottleneck of the HMM is the size of the state space $|X'|$, which enters the TIME complexity quadratically. In this case, however, the state space is the parallelization dimension of the Viterbi algorithm, as all states can be evaluated in parallel, but the $|T'|$ time steps of the model must be computed in sequence. Therefore, unlike for the ELM, the bottleneck of the HMM can successfully be addressed by parallelization; provided that sufficient

parallel capabilities exist, the state space of an HMM can grow arbitrarily large with little impact on computational effort. The sequential time dimension in turn is, for both ELM and HMM, of $|T'| \approx 20$, considering a practical prediction horizon of 5–10 seconds.

While the HMM version of SPARC has yet to be implemented on a parallel platform (a GPU or an FPGA), the above considerations indicate that the HMM is more suited for real-time high-risk applications (such as automated driving) than the iterative optimization of an ELM.

3 Conclusion and Outlook

This paper has reviewed several key decisions in modeling and solving the SPARC concept whose early version was presented in [Zie12, RZR+14]. Two main points were considered of particular interest: Firstly the motivation of trading off estimated collision risks and comfort, which has been considered controversial in many instances; and secondly the decision to replace the iterative solver of the ELM with a global solver based on Hidden Markov Models, instead of improving the iterative solver heuristically. The decisions discussed here are considered the main points that set the SPARC approach apart from related approaches that have been published since then, and thus provide a reference for comparing the respective models.

Research and development on SPARC is still in progress. Current topics include validating the C^2 model for vehicle dynamics in actual test drives, not in simulations as presented in [RZW+14b]; collecting statistical traffic data for developing the prediction framework based on the principles presented in [RZW+14a]; extending the SPARC model to incorporate predictive interaction between the ego vehicle trajectory and trajectories of other traffic participants, both in cases of uncertain interaction and of coordinated cooperation between fully-automated vehicles using Car2Car communication; implementing the HMM-based solver on parallel hardware and evaluating the resulting speedup and costs; and finally integrating SPARC into a real vehicle, and validating its performance.

Bibliography

[BBF+08] A. Bacha, Ch. Bauman, R. Faruque, et al. Odin: Team VictorTango's entry in the DARPA Urban Challenge. *Journal of Field Robotics*, 25(8):467–492, 2008.

[CSA+09] Yi-Liang Chen, V. Sundareswaran, C. Anderson, A. Broggi, P. Grisleri, P. Porta, P. Zani, and J. Beck. TerraMax: Team Oshkosh Urban Robot. In M. Buehler, K. Iagnemma, and S. Singh, editors, *The* DARPA *Urban Challenge*, volume 56 of *Springer Tracts in Advanced Robotics*, pages 595–622. Springer Berlin Heidelberg, 2009.

[MBB+08] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, S. Thrun, et al. Junior: The Stanford Entry in the Urban Challenge. *Journal of Field Robotics*, 2008.

[RZR+14] M. Ruf, J.R. Ziehn, B. Rosenhahn, J. Beyerer, D. Willersinn, and H. Gotzig. Situation Prediction And Reaction Control (SPARC). In B. Färber, K. Dietmayer, K. Bengler, M. Maurer, Ch. Stiller, and H. Winner, editors, *9. Workshop Fahrerassistenzsysteme (FAS 2014)*, pages 55–66, March 2014.

[RZW+14a] M. Ruf, J.R. Ziehn, D. Willersinn, B. Rosenhahn, J. Beyerer, and H. Gotzig. A Continuous Approach to Autonomous Driving. In *Proceedings of the Conference on Vehicle and Infrastructure Safety Improvement in Adverse Conditions and Night Driving (VISION 2014)*, October 2014.

[RZW+14b] M. Ruf, J.R. Ziehn, D. Willersinn, B. Rosenhahn, J. Beyerer, and H. Gotzig. Evaluation of an Analytic Model for Car Dynamics. In *Proceedings of the IEEE International Conference on Mechatronics and Control (ICMC 2014)*, pages 2446–2451, July 2014.

[RZW+15] M. Ruf, J.R. Ziehn, D. Willersinn, B. Rosenhahn, J. Beyerer, and H. Gotzig. Global Trajectory Optimization on Multilane Roads. In *Proceedings of the IEEE 18th International Conference on Intelligent Transportation Systems (ITSC 2015)*, pages 1908–1914, September 2015.

[UAB+08] Ch. Urmson, J. Anhalt, Hong Bae, et al. Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics Special Issue on the 2007* DARPA *Urban Challenge, Part I*, 25(8):425–466, June 2008.

[VB10] B. Van Brunt. *The Calculus of Variations*. Springer, 2010.

[WSM09] J. Wille, F. Saust, and M. Maurer. Segmentübergreifende Bahnplanung mittels eines analytischen Optimierungsverfahrens für die autonome Fahrzeugführung auf dem Braunschweiger Stadtring. In Ch. Stiller and M. Maurer, editors, *6. Workshop Fahrerassistenzsysteme (FAS 2009)*, 2009.

[ZBDS14] J. Ziegler, P. Bender, T. Dang, and C. Stiller. Trajectory planning for Bertha – A local, continuous method. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 450–457, June 2014.

[Zie12] J.R. Ziehn. Energy-based collision avoidance for autonomous vehicles. Master's thesis, Leibniz Universität Hannover, Germany, October 2012.

[ZRR+15] J.R. Ziehn, M. Ruf, B. Rosenhahn, D. Willersinn, J. Beyerer, and H. Gotzig. Correspondence between Variational Methods and Hidden Markov Models. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV 2015)*, pages 380–385, June 2015.