

Correspondence between Variational Methods and Hidden Markov Models*

J. Ziehn^{1,2}, M. Ruf¹, B. Rosenhahn², D. Willersinn¹, J. Beyerer^{1,3} and H. Gotzig⁴

Abstract—This paper establishes a duality between the calculus of variations, an increasingly common method for trajectory planning, and Hidden Markov Models (HMMs), a common probabilistic graphical model with applications in artificial intelligence and machine learning. This duality allows findings from each field to be applied to the other, namely providing an efficient and robust global optimization tool and machine learning algorithms for variational problems, and fast local solution methods for large state-space HMMs.

I. MOTIVATION

A wide range of mathematical models has been used to describe trajectory planning. In the case of fully automated driving, originally the dominant models were highly discretized and mainly based on graph searches. These models have proven successful in the DARPA Grand and Urban Challenges (cf. [MBB⁺08], [UAB⁺08], [KZW⁺08], [PEF13]). Recent works have adopted methods and models from the continuous domain (e.g. [WRG12], [Zie12], [RZR⁺14], [ZBDS14]). These have the advantages of being close to real-world terms, requiring fewer prior assumptions, and providing solutions of high resolution at comparatively low computational effort. The *calculus of variations*, which is usually at the core of these approaches, has already been applied in computer vision (e.g. [KWT88], [LCO⁺04]), engineering and quantum mechanics ([Wei74], [Kom08]). It is commonly solved with iterative descent methods, which have difficulty to discern between local and global minima and to enforce hard constraints, as given by the physical limits of the vehicle. Numerical tools to address both issues are available, but reliable estimates of robustness are hard to establish—in particular in complex real-world scenarios.

From a different domain with different applications comes the *Hidden Markov Model* (HMM), aimed at reconstructing a sequence of discrete states given a sequence of uncertain measurements. Applications include part-of-speech tagging (e.g. [LGB94]), handwriting recognition (e.g. [HBT96]) and tracking (e.g. [LTHRR11]). HMMs have well-understood

global solutions, found by the so-called *Viterbi algorithm*; their main challenge is a high computational effort when the number of states and possible transitions is very large.

Contributions: This paper shows that HMMs and the calculus of variations represent the very same kind of optimization problem in two different ways with respective limitations and strengths. It demonstrates how the two models can be transformed into each other, to find global optima for variational problems, or to quickly find local optima for large HMMs. Representing a variational problem as an HMM has mainly three applications for fully automated driving:

- Global optimization can produce final trajectories (however, the computational effort may still exceed the capacities at the necessary resolution), or provide a low-resolution “initial guess” close to the actual global optimum, which is then refined by a variational method.
- Knowledge of the global optimum can provide references and benchmarks for developing faster iterative optimization tools and better optimization criteria.
- The HMM training framework, that allows parameters to be obtained through machine learning, creates a possibility to test the planning assumptions and estimate free parameters for variational methods. While the parameters of fully automated driving likely cannot be determined through the use of machine learning alone, it can serve to provide realistic estimates of their scale.

II. MODELS

This section briefly introduces the respective methods, their motivations and the notation used here. References to more thorough presentations are given along the way.

A. The Calculus of Variations

The calculus of variations (see [VB10]) is concerned with finding optimal *functions* given so-called *functionals*. Formally, as used in this paper, a functional \mathcal{S} takes a member of a space of differentiable functions, such as

$$\Xi = \{ \xi \mid \xi : [t_{\text{start}}, t_{\text{end}}] \rightarrow X, \xi \in C^2 \} \quad (1)$$

and maps it onto the real numbers $\mathcal{S} : \Xi \rightarrow \mathbb{R}$, where the operation is denoted $\mathcal{S}[\xi]$. The general goal is now to find

$$\xi^* := \arg \min_{\xi \in \Xi} \mathcal{S}[\xi] \quad (2)$$

often given some constraints to limit the space Ξ from which ξ is taken. We will refer to the $\xi \in \Xi$ as *trajectories*, and to their codomain X as the *state space*.

In physical terms the *action* \mathcal{S} of a particle trajectory ξ is

$$\mathcal{S}[\xi] = \int_{t_{\text{start}}}^{t_{\text{end}}} dt L(\xi(t), \dot{\xi}(t), \ddot{\xi}(t), t), \quad (3)$$

*This work was partially supported by Valeo Schalter und Sensoren GmbH within the V50 project, and by the Fraunhofer-Gesellschaft along with the state of Baden-Württemberg within their joint innovation cluster REM 2030.

¹Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB), 76131 Karlsruhe, Germany
{miriam.ruf, jens.ziehn, dieter.willersinn, juergen.beyerer}@iosb.fraunhofer.de

²Institut für Informationsverarbeitung (TNT), Leibniz Universität Hannover, 30167 Hannover, Germany
{rosenhahn, ziehn}@tnt.uni-hannover.de

³Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany

⁴Valeo Schalter und Sensoren GmbH, 74321 Bietigheim-Bissingen, Germany heinrich.gotzig@valeo.com

where t denotes time, $\dot{\xi}$ and $\ddot{\xi}$ denote derivatives of ξ w.r.t. t , and L denotes the particle's *Lagrangian*¹. This particular type of functional is well-understood due to its physical significance, the *principle of least action*; the optimal trajectory is determined using the *Euler–Lagrange equation* given by

$$\delta_{\xi} \mathcal{S} := \frac{\partial L}{\partial \xi} - \frac{d}{dt} \frac{\partial L}{\partial \dot{\xi}} + \frac{d^2}{(dt)^2} \frac{\partial L}{\partial \ddot{\xi}}, \quad (4)$$

where $\delta_{\xi} \mathcal{S}|_{\xi=\xi^*} \equiv 0$ is necessary for the optimality of ξ^* .

1) *Discrete Variational Problems*: While the calculus of variations is originally an analytic formulation, modern applications are mainly based on numerical approaches, using discrete time steps for $\xi(t)$, and (4) for a gradient descent. This discretization is essential for the correspondence established in this paper. Furthermore, discretization transforms the variational problem into classical vector optimization. This has the benefit of unlocking a diverse range of optimization methods, in particular for the introduction of constraints (cf. Sec. II-A.2). This section will outline the terminology of discretization and prove for the case of $L(\xi, \dot{\xi}, \ddot{\xi}, t)$ that optimizing the discretized vector via its gradient is equivalent to optimizing the variational problem using (4) both in its discrete form, and in the continuous limit.

The *functional* \mathcal{S} of a *function*, as in (3), can be turned into a *function* S of a *vector* by dividing $[t_{\text{start}}, t_{\text{end}}]$ into $T \in \mathbb{N}$ intervals of constant width $\Delta t = (t_{\text{end}} - t_{\text{start}})/T$, using a discrete index $\tau \in \{0, \dots, T\}$ and the finite differences approximation $(\Delta/\Delta\tau) y_{\tau} := (y_{\tau+1} - y_{\tau-1})/2\Delta t$ to obtain

$$S = \sum_{\tau=0}^T \Delta t \cdot L_{\tau}(D_{\tau}^0, D_{\tau}^1, D_{\tau}^2) \quad (5)$$

to be used with $D_{\tau}^0 = \xi_{\tau}$, $D_{\tau}^1 = (-\xi_{\tau-1} + \xi_{\tau+1})/2\Delta t$ and $D_{\tau}^2 = (\xi_{\tau-2} - 2\xi_{\tau} + \xi_{\tau+2})/4\Delta t^2$.² For simplification it is useful to, instead of S , consider $\bar{S} := S/\Delta t$ and both S and \bar{S} are now functions of a discrete sequence of points $\xi = [\xi_0, \dots, \xi_T]^{\top}$, so formally $S, \bar{S} : \mathbb{R}^{T+1} \rightarrow \mathbb{R}$.

The τ -th component of gradient $\nabla \bar{S}$ w.r.t. ξ is given by

$$\begin{aligned} \nabla_{\tau} \bar{S} &= \frac{\partial}{\partial \xi_{\tau}} \bar{S} = \frac{\partial L_{\tau}}{\partial D_{\tau}^0} \frac{\partial D_{\tau}^0}{\partial \xi_{\tau}} \rightarrow +1 \\ &+ \frac{\partial L_{\tau-1}}{\partial D_{\tau-1}^1} \frac{\partial D_{\tau-1}^1}{\partial \xi_{\tau}} \rightarrow + \frac{1}{2\Delta t} \\ &+ \frac{\partial L_{\tau+1}}{\partial D_{\tau+1}^1} \frac{\partial D_{\tau+1}^1}{\partial \xi_{\tau}} \rightarrow - \frac{1}{2\Delta t} \\ &+ \frac{\partial L_{\tau-2}}{\partial D_{\tau-2}^2} \frac{\partial D_{\tau-2}^2}{\partial \xi_{\tau}} \rightarrow + \frac{1}{4\Delta t^2} \\ &+ \frac{L_{\tau}}{\partial D_{\tau}^2} \frac{\partial D_{\tau}^2}{\partial \xi_{\tau}} \rightarrow - \frac{2}{4\Delta t^2} \\ &+ \frac{\partial L_{\tau+2}}{\partial D_{\tau+2}^2} \frac{\partial D_{\tau+2}^2}{\partial \xi_{\tau}} \rightarrow + \frac{1}{4\Delta t^2} \end{aligned} \quad (6)$$

¹ $\ddot{\xi}$ is classically excluded from L , but all approaches given here extend to any finite order; $\dot{\xi}$ is included to demonstrate the necessary principles.

²This formulation uses central differences, as these better represent the necessity of upper and lower limits for derivatives. The most intuitive application for HMMs is backward differences, as will be used in Sec. III.

(the boxes indicate substitutions), which can be rewritten as

$$\begin{aligned} \nabla_{\tau} \bar{S} &= \frac{\partial L_{\tau}}{\partial D_{\tau}^0} \\ &- \frac{1}{2\Delta t} \left(\frac{\partial L_{\tau+1}}{\partial D_{\tau+1}^1} - \frac{\partial L_{\tau-1}}{\partial D_{\tau-1}^1} \right) \rightarrow \frac{\Delta}{\Delta\tau} \left(\frac{\partial L_{\tau}}{\partial D_{\tau}^1} \right) \\ &+ \frac{1}{4\Delta t^2} \left(\frac{\partial L_{\tau-2}}{\partial D_{\tau-2}^2} - \frac{2L_{\tau}}{\partial D_{\tau}^2} + \frac{\partial L_{\tau+2}}{\partial D_{\tau+2}^2} \right) \rightarrow \frac{\Delta^2}{(\Delta\tau)^2} \left(\frac{\partial L_{\tau}}{\partial D_{\tau}^2} \right). \end{aligned} \quad (7)$$

and, as per factor rule, $\nabla_{\tau} S = \Delta t \cdot \nabla_{\tau} \bar{S}$. Now a given ξ^* is stationary w.r.t. S iff $\|\nabla S(\xi)\|_2^2|_{\xi=\xi^*} = 0$ with

$$\|\nabla S\|_2^2 = \sum_{\tau=0}^T \Delta t \cdot \left(\frac{\partial L_{\tau}}{\partial D_{\tau}^0} - \frac{\Delta}{\Delta\tau} \frac{\partial L_{\tau}}{\partial D_{\tau}^1} + \frac{\Delta^2}{(\Delta\tau)^2} \frac{\partial L_{\tau}}{\partial D_{\tau}^2} \right)^2 \quad (8)$$

Thus far, Δt has not been significant, as any vector stationary w.r.t. S is also stationary w.r.t. \bar{S} . However, only S converges in the limit of $T \rightarrow \infty$, such that $\Delta t \rightarrow 0$, yielding

$$\lim_{\Delta t \rightarrow 0} \|\nabla S\|_2^2 = \int_{t_{\text{start}}}^{t_{\text{end}}} dt |\nabla \bar{S}|^2 \stackrel{!}{=} 0 \Leftrightarrow \forall \tau : \nabla_{\tau} \bar{S} \stackrel{!}{=} 0, \quad (9)$$

which mirrors the optimality condition of (4). The above condition can be computed only at $\tau \in \{3, \dots, T-2\}$. For $\tau \in \{1, T\}$ the factor $\partial L_{\tau\pm 1}/\partial D_{\tau\pm 1}^1$ cannot be computed; for $\tau \in \{2, T-1\}$, also the factor $\partial L_{\tau\pm 2}/\partial D_{\tau\pm 2}^2$ is undefined.

The straightforward solution is to set $\nabla_{\tau} S := 0$ for $\tau \in \{1, 2, T-1, T\}$. This effects that during a gradient descent the elements ξ_0, ξ_1, ξ_{T-1} and ξ_T are not optimized and remain fixed (to the values in the ‘‘initial guess’’). The first and last points and velocities must thus be known in advance. This mirrors the assumption of fixed boundary points and velocities in the classical derivation of (4). If $\dot{\xi}$ is not used in L , then ξ_1 and ξ_{T-1} (and thus the velocities) can be optimized and only the boundary points must be given. For optimizable start- and endpoints, see Sec. II-A.2.a.

2) *Constraints and Generalizations*: The previously described derivation has three significant properties that limit its application: The trajectory ξ is assumed to lie between fixed start- and endpoints, $\xi(t_{\text{start}}) \stackrel{!}{=} x_{\text{start}}$ and $\xi(t_{\text{end}}) \stackrel{!}{=} x_{\text{end}}$ ³; if the Lagrangian as in (3) includes $\dot{\xi}$, then furthermore the start and end velocities \dot{x}_{start} and \dot{x}_{end} are fixed; other values of $\xi, \dot{\xi}$ or $\ddot{\xi}$ however are unconstrained.

a) *Fixed Endpoints or Natural Boundary Conditions*:

The need for fixed endpoints is connected to the derivation of the Euler–Lagrange equation, which takes the simple form of (4) only under these conditions. If one of these must be variable for a given application, the fixed endpoints and end velocities must give way to *natural boundary conditions* (NBCs), which describe the condition of endpoints being optimal w.r.t. the action \mathcal{S} (see [VB10] for details).

b) *Additional Constraints*: Other than for the start and endpoints, the Euler–Lagrange equation does not address constraints. If required, such constraints can be introduced either by choice of parameters, or (in cases where this is not

³For clarity, ξ_{τ} (or $\xi(t)$) $\in X$ will be used to refer to elements of the sequence (or function) ξ at time τ (or t), and $x_i \in X$ to refer to one of the possible state space elements in general, not necessarily part of a given trajectory. For example, ξ_3 is the third step along a trajectory, but x_3 is merely some point in space. The same applies to the later use if σ_{τ} and ς_i .

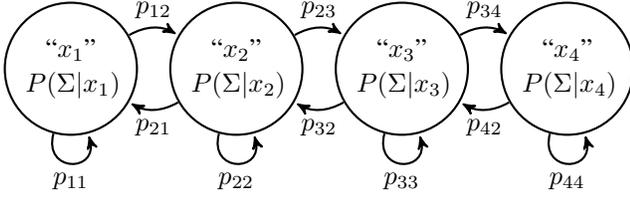


Fig. 1: An HMM with four states and probability distributions for the emission of symbols in each particular state. p_{ij} here is short for $p(x_j \leftarrow x_i | x_i)$.

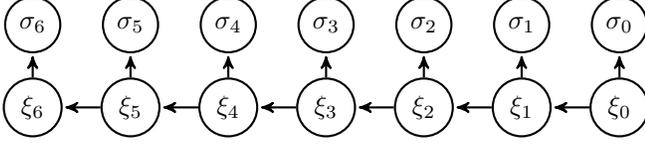


Fig. 2: Graphical representation of a process with a given HMM (possibly the one from Fig. 1). During six time steps, seven states $[\xi_0, \dots, \xi_6]$ are assumed by the HMM, and seven symbols $[\sigma_0, \dots, \sigma_6]$ are emitted. Usually only the symbols are considered known, and used to infer the most likely sequence of the underlying, “hidden” states.

feasible) as part of the optimization method. For example, trajectories of vehicles can be constrained to a fixed path by optimizing arc length instead of a higher-dimensional position in space (e.g. [RZR⁺14], [RZW⁺14]). However, usually the transformation into a lower-dimensional, unconstrained problem is not as straightforward. In these cases, the common approach is to use an optimization tool that is able to enforce the constraints, restricting the solution to the *feasible region* Ξ (methods include reprojection to Ξ after every optimization step, barrier methods or active set methods, see [NW06]).

B. Hidden Markov Models

Hidden Markov Models (HMMs) are probabilistic graphical models, and an extension of *Markov chains*. Both models make use of the *Markov property*, which states that a stochastic process is *memoryless* in that its future behavior only depends on the current state, but not on the previous states (or, equivalently, on just a fixed and finite number of previous states, so that the information stored in the *current state* can be expanded to account for the limited history).

1) *Markov chains*: A Markov chain is a graphical model that has *states* $x \in X$ representing the nodes of the graph, and possible *state transitions* that represent directed edges $(x_j \leftarrow x_i) \in X \times X$ (the right-to-left temporal order will simplify the later notation). There are probabilities assigned to the transitions, $p(x_j \leftarrow x_i | x_i)$, and also probabilities for initial states $p(x_i \leftarrow \cdot)$. The goal is now to determine the probability $p(\xi)$ of a sequence $\xi = [\xi_0, \xi_1, \dots, \xi_T]^T$ over $T \in \mathbb{N}$ time steps (with each $\xi_\tau \in X$) or the most probable sequence $\xi^* := \arg \max_\xi p(\xi)$. It is also possible to determine the most likely set of parameters to the Markov chain given some training sequences.

2) *Extension to Hidden Markov Models*: HMMs (see [MZ97]) extend Markov chains by having the states *emit symbols* $\varsigma \in \Sigma$ at random, one per time step. For this, each state x_i has an individual probability distribution $P(\Sigma | x_i)$, such that some states are more likely to emit certain symbols than others (cf. Fig. 1). Now, the internal state of an HMM cannot be observed (it is *hidden*), but the observed sequence of symbols $\sigma = [\sigma_0, \dots, \sigma_T]^T$ gives some indication about

the probability of a proposed sequence of states ξ :

$$p(\xi | \sigma) \propto p(\xi, \sigma) = \left(\prod_{\tau=1}^T \underline{p(\sigma_\tau | \xi_\tau)} \cdot p(\xi_\tau \leftarrow \xi_{\tau-1} | \xi_{\tau-1}) \right) \cdot \underline{p(\sigma_0 | \xi_0)} \cdot p(\xi_0 \leftarrow \cdot), \quad (10)$$

where the underlined terms are distinctive of the HMM model, whereas the rest describes the basic Markov chain.

III. CORRESPONDENCE

To map the calculus of variations onto an HMM, it is necessary that the codomain or the image of the trajectory ξ can be effectively expressed as a finite set of states.⁴ For the reverse mapping, it must be possible to order and interpolate the states of the HMM to yield a continuously differentiable state space.⁵ The main corner points of the transformation are to relate a sequence of states over time in an HMM to values taken by a function over time in a variational problem; to relate the maximization of multiplicative probabilities to the minimization of additive Lagrangians; and to relate the local, finite Markov memory to the local, finite Taylor expansion used to evaluate the Lagrangian. Shared by all of the following transformations is how the boundary point types from the calculus of variations relate to terms of the HMM, given in Tab. I. The goal is a correspondence of both the optima and optimal values, such that for the limit of an infinitely fine HMM state space

$$\arg \max_{\xi \in \Xi} p(\xi | \sigma) = \arg \min_{\xi \in \Xi} S(\xi) = \xi^* \quad (11)$$

$$p(\xi^* | \sigma) = 1/Z \cdot \exp(-S(\xi^*)). \quad (12)$$

with a normalization constant Z s.t. $\sum_{\xi \in \Xi} p(\xi | \sigma) = 1$.⁶

A. Separable Lagrangian, First-Order Temporal Dependence

The most immediate correspondence can be established for a Lagrangian of the form (here referred to as *separable*)

$$L(x, \dot{x}, t) = L_1(x, t) + L_2(x, \dot{x}), \quad (13)$$

namely such that the velocity is penalized in a time-invariant way (which is not uncommon, e.g. [KWT88], [RZR⁺14]). In this case, it can be said in terms of the HMM parameters,⁷

$$p(\sigma_\tau | x_2) = 1/Z_1 \cdot \exp(-L_1(x_2, \tau \cdot \Delta t + t_{\text{start}})) \quad (14)$$

$$p(x_2 \leftarrow x_1 | x_1) = 1/Z_2 \cdot \exp(-L_2(x_2, (x_2 - x_1)/\Delta t)) \quad (15)$$

or equivalently for the Lagrangian

$$L_1(x, t) = -\log p(\sigma_{[(t-t_{\text{start}})/\Delta t]} | x) \quad (16)$$

$$L_2(x, \dot{x}) = -\log p(x \leftarrow [x - \Delta t \cdot \dot{x}] | [x - \Delta t \cdot \dot{x}]), \quad (17)$$

⁴This means that discretizing X is possible, and ξ or $\dot{\xi}$ is bounded (one implies the other on a closed time interval for $\Xi \subseteq C^1$, cf. (1)).

⁵Neighboring states must have similar probability distributions, and there should be larger monotonous intervals for their parameters. Otherwise, the transformation to the variational problem would be mathematically valid, but unfit for gradient descent optimization for lack of meaningful derivatives.

⁶(12) resembles the *Boltzmann distribution* $p(x) = 1/Z \cdot \exp(-\beta E(x))$ ([Dem10]) with a *thermodynamic beta* $\beta = 1/k_B \cdot T = 1$.

⁷The constants Z , Z_1 and Z_2 are relevant to define actual probabilities; however, the optimum found by the Viterbi algorithm is invariant to an overall constant scaling factor, therefore $Z = Z_1 = Z_2 = 1$ is *algorithmically* admissible, if the values that subsequently occur in the algorithm are not mistaken for probabilities. Furthermore, the transformation from p to L completely omits any possible constants (scale and offset) for clarity.

CALC. OF VARIATIONS	HIDDEN MARKOV MODEL
startpoint fixed to x_{start}	$p(x \leftarrow) = \delta_{x, x_{\text{start}}}$
startpoint NBC, arbitrary	$p(x \leftarrow) = 1/ X $
startpoint NBC, penalized via $P_{\text{start}}(x)$	$p(x \leftarrow) \propto \exp(-P_{\text{start}}(x))$
endpoint fixed to x_{end}	backtracking from x_{end}
endpoint NBC, arbitrary	backtracking from x with max. p
endpoint NBC, penalized via $P_{\text{end}}(x)$	multiplying end state probabilities with $\exp(-P_{\text{end}}(x))$, backtracking from new most probable end state

TABLE I: CORRESPONDING BOUNDARY POINT TYPES.

where the square brackets denote rounding to the closest applicable state or integral τ in the model.

B. Joint Lagrangian, First-Order Temporal Dependence

The above transformations assume, in accordance with the HMM model, that the *states* emit symbols, not the *state transitions*, thus making the probability of states dependent on σ , while the transitions are considered time-invariant. This in effect leads to the separation into L_1 and L_2 as given in (13), instead of a joint $L(x, \dot{x}, t)$ as assumed in the calculus of variations. This, however, is not a fundamental limitation. A classical HMM can be thought of as a probabilistic *Moore* automaton (emissions during states), while the general Euler–Lagrange model would correspond to *Mealy* automata (emissions during transitions); both models can be transformed into one another (see [Gil60]). Thus any HMM can as well be thought of as emitting symbols during (and thus dependent on) transitions, instead of states. To achieve this, a “Mealy HMM” can be set up by replacing (14) with

$$p(\sigma_\tau | x_2 \leftarrow x_1) \propto \exp\left(-L(x_2, \frac{x_2 - x_1}{\Delta t}, \tau \cdot \Delta t + t_{\text{start}})\right), \quad (18)$$

which can then be turned into a Moore automaton that describes the classical HMM. It should be noted that this is mostly a theoretical consideration; for practical applications, the more straightforward way is to extend the Viterbi algorithm to rate the transitions based on a complete $L(x, \dot{x}, t)$.

C. Higher-Order Temporal Dependence

Another aspect allowed for in the calculus of variations, but not explicitly modeled in the HMM, is the dependence on more than just a single transition (which, in the variational model, corresponds to derivatives higher than \dot{x} , and in the HMM to a Markov memory of more than one previous state). In this case, the “state” must be extended to also represent the first $m := n - 1$ derivatives, $\hat{x} := [x, \dot{x}, \ddot{x}, \dots, d^m x / (dt)^m]^\top$. The *state transitions* then represent $d^n x / (dt)^n$ and thus a transition from \hat{x}_1 can lead to an approximate new state

$$\hat{x}_2 = \left[x_1 + \Delta t \cdot \dot{x}_1, \dot{x}_1 + \Delta t \cdot \ddot{x}_1, \dots, \frac{d^m x_1}{(dt)^m} + \Delta t \cdot \frac{d^n x_1}{(dt)^n} \right]^\top. \quad (19)$$

It should be noted that we refer to the state space always as X , and the transition space always as \dot{X} , regardless of which derivative the elements of \dot{X} represent.

IV. EVALUATION AND COMPARISON

The evaluation of the variational model and the HMM is twofold: A practical application will demonstrate the

different nature of results produced by them given the same example problems; a theoretical comparison will provide a brief and more general overview of each model’s properties.

A. Practical Application

In the following examples, the Lagrangian is of the form

$$L(x, \dot{x}, \ddot{x}, t) = \Phi(x, \tau) + a \|\ddot{x}\|_2^2 + b \delta(t - t_{\text{end}}) \|x - x_{\text{end}}\|_2^2, \quad (20)$$

where Φ is the given scalar field (or “potential”) normalized to $[0, 1]$ ([black, white] in the figures), a and b are weights and δ is the Dirac delta. The values are given in the respective figure captions, but should be understood as examples only. In each case, the HMM is connected as in Fig. 1, i.e. only neighboring states have transitions and each state has a loop.

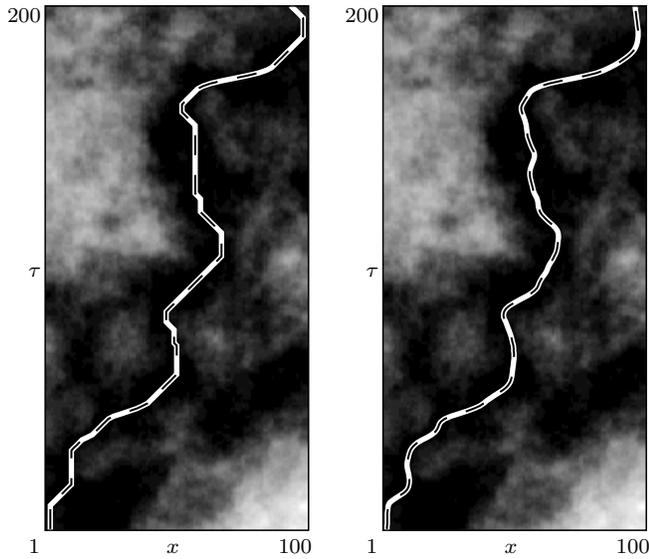
1) *Application to Brownian Noise Fields*: To not presuppose a specific problem structure, in this application, shown and discussed in detail in Fig. 3, the two approaches are compared on random fields based on two-dimensional ($X \times \{1, \dots, T\}$) Brownian noise, resulting in fields that feature both obvious large-scale structures for a clear global distribution of optima, and fine details affecting their precise location. The aim of this application is to show, how the choice of free parameters (i.e. parameters specific to the solver, not the problem) affects the obtained solution. For HMMs, this is mainly the degree of discretization; for variational descent methods, it is mainly the initial guess (and, to a lesser degree, the descent method; not discussed here). The best solution (cf. Fig. 3e) is attained using HMMs to find an approximation that is then used as an initial guess to be refined by an Euler–Lagrange descent (cf. Fig. 3f).

Due to the unoptimized nature of the implementations, a comparison of computation times is not meaningful. What can be stated is that the BFGS-based SQP implementation requires an average of 18.6 iterations to converge, and an average of 46.8 function evaluations of S per iteration; as S represents the full action \mathcal{S} , each S evaluation comprises 25 evaluations of the Lagrangian.⁸ This leads to a total of about $2.1 \cdot 10^4$ Lagrangian evaluations. The HMM requires a fixed 37,160 Lagrangian evaluations per time step, which, for 25 time steps, leads to $9.2 \cdot 10^5$ evaluations of the Lagrangian, roughly one and a half orders of magnitude more.

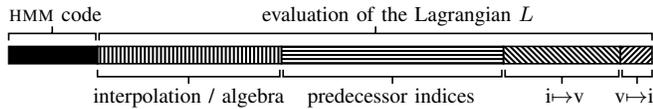
It is assumed that generally HMMs are less sensitive to the choice of parameters, and the parameters are more intuitive in their effect. Ill-chosen parameters for a gradient descent may cause unexpected results and divergent, possibly chaotic behavior, making their use in real-time applications risky.

2) *Application to automotive trajectory planning*: In this section, the motivating problem of fully-automated driving is discussed on an example taken from [RZR⁺14] and explained in more detail there. The ego vehicle (cf. Fig. 4a) wants to turn left at an intersection, and has to avoid two cars of oncoming traffic. The front car takes a right turn, the rear car passes straight through the intersection. Depending on their spacing, either passing in between the two cars (Fig. 4b) or waiting for both cars to clear the intersection (Fig. 4c) before following the first at a safe distance represents desirable behavior. The decision is based on a field of occupancy

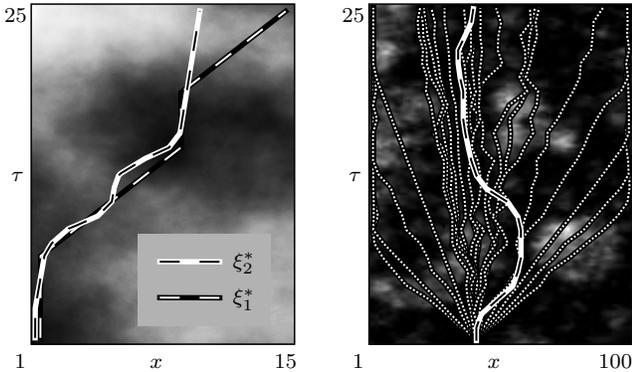
⁸Fewer if analytic gradients (as given by (4)) and Hessians are provided.



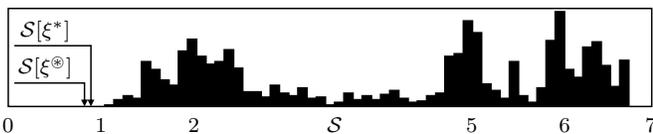
(a) Effect of state space resolution at large scales (state includes location and speed). The field is sampled at $x \in \{1, \dots, 100\}$, $\tau \in \{1, \dots, 200\}$. Computation time is ca. 1 minute. (b) The field of (a) is upsampled to $x \in \{0.125, \dots, 100\}$ using linear interpolation. Derivatives are finer but the trajectory is almost the same. Computation time is ca. 6 hours.



(c) Percentage of computation time used for (b). The largest share is taken by the computation of predecessor indices, and index-to-vector and vector-to-index conversions, which could be optimized by lookup tables.

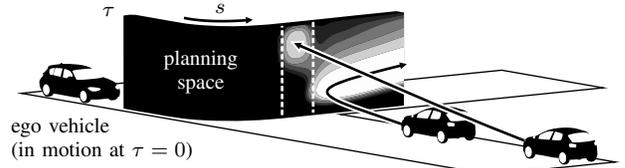


(d) Small-scale effect of state space resolution, in comparison to (a) and (b). The field is sampled at $x \in \{1, \dots, 25\}$ for ξ_1^* and $x \in \{0.125, 0.25, \dots, 25\}$ for ξ_2^* (figure axis cropped). At this small scale, the difference in resolution has more significant effects. (e) Global vs. local optimization. The global optimum ξ^* found by the HMM is shown dashed. With “initial guesses” radiating from x_{start} at 1,200 different speeds, the Euler-Lagrange descent gets stuck in local minima (examples shown dotted), none of which matches ξ^* . Cf. (f).

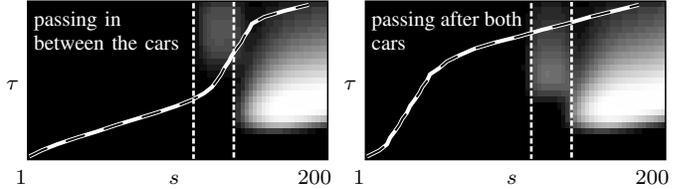


(f) Histogram of actions S for all initial guesses as in (e). The solution $S[\xi^*] = 0.903$ (found by the HMM) is never found from any such guess; the closest solution reaches $S = 1.148$ (average $S = 4.523$). Using ξ^* as an initial guess yields $S[\xi^\circ] = 0.833$ (possible due to the continuous X).

Fig. 3: Example IV-A.1: Parameter effects of variational models and HMMs. Φ as shown in the background (black=0, white=1), $a = 5 \cdot 10^{-3}$, $b = 0$, $\xi \in [-1, 1]$, $\xi \in [-6, 6]$, startpoint is fixed, endpoint is arbitrary by NBCs.



(a) Problem setup; vertical axis is time



(b) Solution for wide spacing (c) Solution for narrow spacing

Fig. 4: Example IV-A.2 as in [RZR⁺14]: A left turn through two oncoming cars. $a = 5 \cdot 10^{-3}$, $b = 2 \cdot 10^{-4}$, $x_{\text{end}} = 200$, $\xi \in [0, v_{\text{max}}]$. The dashed interval of s shows where the ego vehicle’s path crosses the oncoming lane.

predictions, generated by means described in [RZR⁺14] and [RZW⁺14]. This field is used to plan a trajectory along the arc length s of the given path with maximum safety and comfort, by minimizing collision risks as well as longitudinal and lateral accelerations, while obeying traffic rules. While [RZR⁺14] notes that the obtained solution depends on the initial guess in an Euler-Lagrange descent, HMMs always determine the optimal solution (provided in Fig. 4). The relevant distinction between the two cases can, for the given parameters, be made at a field resolution of 8×8 in below $0.2s$ with unoptimized, serial MATLAB code on a Core i7@2.90 GHz (see also Fig. 3c).

B. Theoretical Comparison

This section compares the two approaches in terms of their general computational properties, namely how effort scales with problem size, how they can be parallelized and what kind of results can be expected.

1) *Computational Complexity*: While a firm and complete upper bound for the Viterbi algorithm exists, given by

$$\text{VITERBI} \in \text{TIME}(T \cdot |X| \cdot |\dot{X}|) \cap \text{SPACE}(T \cdot |X|), \quad (21)$$

the only upper bounds for the Euler-Lagrange approach are

$$\text{EL}^1 \in \text{SPACE}(T) \quad \text{and} \quad \text{EL}^2 \in \text{SPACE}(T^2), \quad (22)$$

where EL^1 refers to, e.g., a simple gradient descent or L-BFGS, while EL^2 refers to algorithms using a full Hessian (e.g. Newton’s method or dense BFGS). The TIME required to reach convergence cannot be given; theoretically, infinitely many steps are possible, depending on the given problem. In general, however, the Euler-Lagrange optimization does *not* scale with the number of states X , assuming that X is not always monotonous. Therefore, this approach scales benignly with $|X|$, as opposed to the Viterbi algorithm.

Both algorithms can benefit from parallelization in terms of scaled speedup (cf. [Gus88]) but each maintains a significant serial fraction. The Viterbi algorithm can compute the entirety of actions for all $x \in X$ in parallel at every τ , but the accumulation over τ has to be performed serially. The Euler-Lagrange descent instead can compute all $\tau \in \{1, \dots, T\}$ components of ∇S in parallel, but the descent through the state space must be serial. Thus, a choice of models also determines which space, time or state, can be parallelized.

COMPARED CRITERION	CALCULUS OF VARIATIONS	HIDDEN MARKOV MODEL
usual optimization method	iterative linear or quadratic programming; e.g. gradient descent or dense Hessian SQP	dynamic programming; Viterbi algorithm
usual domain choices for above optimization method	times discrete and bounded, state space continuous and unbounded: $T \in \mathbb{N}$, $X = \mathbb{R}$	times <i>and</i> state space discrete and bounded: $T \in \mathbb{N}$, $ X \in \mathbb{N}$.
optimization goal	stationary solution or local optimum	global optimum
other constraints	enforcement possible but in general difficult to handle	local constraints straightforward
optimization complexity	SPACE(T) or SPACE(T^2), TIME to convergence unbounded	SPACE($T \cdot X $), TIME($T \cdot X \cdot \dot{X} $)
parallelization parameters	times parallel, iterations serial, $ X $ is not a scale parameter	state space parallel, times serial

TABLE II: COMPARISON OF SEVERAL KEY FEATURES OF THE TWO DISCUSSED MODELS.

2) *Quality of Results*: For the given computational requirements, it must be minded that the Viterbi algorithm is a global optimizer, while the Euler–Lagrange descent is only a local optimizer—thus, the two methods generally do *not* seek the same type of solution. Yet, as seen in Fig. 3d, the discretization necessary to render the HMM approach tractable can considerably affect the solution. Thus, not every variational problem may lend itself to the transformation to an HMM, as the global optimum of the discretized model may not sufficiently reflect the global optimum of the continuous model. Conversely, not every HMM can be reasonably turned into a variational model; namely, if the set of states X cannot be continuously approximated, the resulting action S will have a multitude of scattered local extrema, and if convergence is reached, the result may be near-random.

Table II summarizes the key features of the two models.

V. CONCLUSION

This paper has shown that the HMM and the calculus of variations share the same general optimization goal, one in a discrete, the other in a continuous form. The transformations between the models have been given, and their characteristics have been demonstrated on example problems. Whether a particular problem benefits from this transformation, or if it should rather be solved within its natural formulation, can only be decided based on the problem’s individual characteristics; several factors to consider have been given.

It appears that the strengths of the HMM approach provide several valuable advantages for real-time trajectory planning (as in automated driving). The Viterbi algorithm is less sensitive to the choice of parameters and inputs; it terminates after a known number of steps; it can be efficiently implemented in parallel hardware; and it is able to compute more than one possible trajectory at almost no further cost—this is relevant for fail-safe emergency stop applications in case of a system failure, such as discussed in [RZW⁺14].

Future works must extend the evaluation, in particular to address the transformation of real-world HMMs to variational approaches. The provided correspondence can further be generalized to variational problems involving functions of several variables; here, a special case of matching probabilistic graphical models would be Markov Random Fields (MRFs) and Conditional Random Fields (CRFs).

REFERENCES

[Dem10] W. Demtröder. *Atoms, Molecules and Photons*. Graduate Texts in Physics. Springer, 2010.

[Gil60] A. Gill. Comparison of finite-state models. *IRE Transactions on Circuit Theory*, 7(2):178–179, June 1960.

[Gus88] J. L. Gustafson. Reevaluating Amdahl’s Law. *Communications of the ACM*, 31(5):532–533, May 1988.

[HBT96] Jianying Hu, M. K. Brown, and W. Turin. HMM based online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1039–1045, 1996.

[Kom08] L. Komzsik. *Applied Calculus of Variations for Engineers*. CRC Press, 2008.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[KZW⁺08] S. Kammel, J. Ziegler, M. Werling, T. Gindele, Ch. Frese, Ch. Stiller, et al. Team annieWAY’s autonomous system for the DARPA urban challenge. In *International Journal of Field Robotics Research*, 2008.

[LCO⁺04] Weiguo Lu, Ming-Li Chen, G. H. Olivera, K. J. Ruchala, and Th. R. Mackie. Fast free-form deformable registration via calculus of variations. *Physics in Medicine and Biology*, 49(14):3067–3087, 2004.

[LGB94] G. Leech, R. Garside, and M. Bryant. CLAWS4: The tagging of the British National Corpus. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING 94)*, Kyoto, Japan, pages 622–628, 1994.

[LTHRR11] L. Leal-Taixé, M. Heydt, A. Rosenhahn, and B. Rosenhahn. *Understanding what we cannot see: automatic analysis of 4D digital in-line holographic microscopy data*. Springer, 2011.

[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.

[MZ97] I. L. MacDonald and W. Zucchini. *Hidden Markov and Other Models for Discrete-valued Time Series*. Monographs on Statistics & Applied Probability. Taylor & Francis, 1997.

[NW06] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2006.

[PEF13] J. Petereit, T. Emter, and C. W. Frey. Safe mobile robot motion planning for waypoint sequences in a dynamic environment. In *IEEE International Conference on Industrial Technology (ICIT)*, pages 181–186, 2013.

[RZR⁺14] M. Ruf, J. Ziehn, B. Rosenhahn, J. Beyerer, D. Willersinn, and H. Gotzig. Situation Prediction And Reaction Control (SPARC). In B. Färber et al., editors, *9. Workshop Fahrerassistenzsysteme – FAS2014*, pages 55–66, March 2014.

[RZW⁺14] M. Ruf, J. Ziehn, D. Willersinn, B. Rosenhahn, J. Beyerer, and H. Gotzig. A Continuous Approach to Autonomous Driving. In *Vehicle and Infrastructure Safety Improvement in Adverse Conditions and Night Driving (VISION)*, Versailles, Oct. 2014.

[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.

[Wei74] R. Weinstock. *Calculus of Variations*. Dover Books on Mathematics. Dover Publications, 1974.

[WRG12] M. Werling, Ph. Reimisch, and K. Gresser. Kombinierte Brems-Ausweich-Assistenz mittels nichtlinearer modellprädiktiver Trajektorienplanung für den aktiven Fußgängerschutz. In K. Dietmayer et al., editors, *8. Workshop Fahrerassistenzsysteme – FAS2012*, pages 77–86, September 2012.

[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. Ziehn. Energy-based collision avoidance for autonomous vehicles. Master’s thesis, Leibniz Universität Hannover, 2012.