

Laplacian Trajectory Vector Fields for Robotic Movement Imitation and Adaption

Thomas Nierhoff, Sandra Hirche, and Yoshihiko Nakamura

Abstract Vector field methods for trajectory following are generally computed offline before execution and thus only applicable to static trajectories. In contrast this paper introduces Laplacian trajectory vector fields (LTVF) as a computationally efficient method for creating convergent vector fields towards a discretized reference trajectory. In case of environmental changes both the vector field and the reference trajectory can be quickly recomputed. The conducted experiment uses a HRP-4 robot in order to display the applicability to daily life problems.

1 Introduction

Vector fields are a standard method for reactive trajectory adaption, allowing one to calculate a desired movement vector for any point in space while avoiding obstacles and maintaining a desired movement behavior. Probably the oldest and most used vector fields in robotics are potential fields [8, 1], focusing on collision avoidance in both static and non-static environments by superimposing repellent forces from each obstacle. Other approaches have the goal of a safe physical interaction with a human user [12], goal convergence in constrained environments [9] or global convergence towards closed curves [5]. A recent trend focusing on imitation learning incorporates the modulation of a dynamical system [6, 4], thus overcoming the problem of getting stuck in a local minimum and being able to encode more complex movements. An elaborate approach for convex-shaped obstacle avoidance while maintaining the shape of a goal-driven motion is presented in [7].

Thomas Nierhoff, Sandra Hirche
Institute for Information-Oriented Control (ITR), Faculty of Electrical Engineering, Technische Universität München, D-80290 München, Germany. e-mail: {tn, hirche}@tum.de

Thomas Nierhoff, Yoshihiko Nakamura
Department of Mechano-Informatics, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, 113-8656 Tokyo, Japan. e-mail: nakamura@ynl.t.u-tokyo.ac.jp

Most of the presented vector field methods for trajectory following are learnt offline and assume the underlying trajectory not to be changed during execution. Yet in the presence of large disturbances or environmental changes this can cause undesired and possibly diverging trajectories [3]. An alternative option presented in this paper is to recompute the trajectory and recalculate the resulting vector field for faster convergence.

The contribution of this paper is a method to construct converging vector fields for arbitrarily, discretized trajectories. Regarding computational complexity, the vector field is recalculated quickly whenever the trajectory has to be updated. Convergence of the integral curves along the trajectory towards an end point can be ensured by interpolating between multiple first-order dynamical systems. Continuity of the vector field within specified bounds is proven. Because the vector field depends only on a few parameters its calculation can be automatized fully and combined with a Programming by Demonstration approach as shown in the experiment using an HRP-4 robotic platform.

2 Approach

2.1 Laplacian Vector Fields

Let a trajectory be the combination of a path $\mathbf{P} = [\mathbf{p}(t_1), \mathbf{p}(t_2), \dots, \mathbf{p}(t_n)]^T \in \mathbb{R}^{n \times m}$ with n sampling points $\mathbf{p}(t_i)$ and associated temporal information $t_i \in \mathbb{R}$. For simplicity, $\mathbf{P} = [\mathbf{p}(t_1), \mathbf{p}(t_2), \dots, \mathbf{p}(t_n)]^T$ is rewritten as $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n]^T$. The trajectory consists of trajectory segments τ_i defined as the line segment between two subsequent sampling points $\mathbf{p}_i, \mathbf{p}_{i+1}$, $i \in \{1, \dots, n-1\}$, see Fig. 1. Every trajectory segment has a mid point $\mathbf{p}_{m,i} = \frac{\mathbf{p}_i + \mathbf{p}_{i+1}}{2}$, lying on the hyperplane h_i with corresponding normal vector $\mathbf{n}_i = \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{\|\mathbf{p}_{i+1} - \mathbf{p}_i\|}$. The vector fields for the remainder of this paper are of the general form

$$\dot{\mathbf{p}} = \mathbf{A}_i \mathbf{p} + \mathbf{b}_i, \quad (1)$$

with $\mathbf{A} \in \mathbb{R}^{m \times m}$, $\mathbf{b} \in \mathbb{R}^m$. For the i -th trajectory segment, they can be split up as

$$\begin{aligned} \mathbf{A}_i &= \mathbf{A}_{\parallel i} + \mathbf{A}_{\perp i}, \\ \mathbf{b}_i &= \mathbf{b}_{\parallel i} + \mathbf{b}_{\perp i}, \end{aligned} \quad (2)$$

consisting of a parallel vector field $V_{\parallel i} = \{\mathbf{A}_{\parallel i}, \mathbf{b}_{\parallel i}\}$ responsible for guiding an object along the trajectory and an orthogonal vector field $V_{\perp i} = \{\mathbf{A}_{\perp i}, \mathbf{b}_{\perp i}\}$ for convergence towards the trajectory in case of a sudden disturbance. In order to calculate the elements of the vector fields, we define an orthonormal basis in \mathbb{R}^m with basis vectors $\mathbf{a}_{j,i}$, $j \in \{1, \dots, m\}$ such that $\mathbf{a}_{1,i}$ coaligns with $\mathbf{p}_{i+1} - \mathbf{p}_i$. It is assumed that the eigenvectors $\mathbf{e}_{j,i}$ of \mathbf{A}_i coincide with $\mathbf{a}_{j,i}$. The scalar values $\lambda_{\parallel j,i}$ and $\lambda_{\perp j,i}$ denote the eigenvalues of $\mathbf{A}_{\parallel i}$ and $\mathbf{A}_{\perp i}$. Then the parallel vector field $V_{\parallel i}$ has to fulfill the conditions

$$\begin{aligned} \lambda_{\parallel 1,i} = \lambda_{\parallel 2,i} = \dots = \lambda_{\parallel m,i} &= 0, \\ \dot{\mathbf{p}}(\mathbf{p}_{m,i}) &= \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{t_{i+1} - t_i} > 0, \end{aligned} \quad (3)$$

that is a vector field with constant velocity parallel to the i -th trajectory segment. The orthogonal vector field $V_{\perp i}$ has to fulfil the condition

$$\begin{aligned} \lambda_{\perp 1,i} &= 0, \\ \lambda_{\perp 2,i} = \lambda_{\perp 3,i} = \dots = \lambda_{\perp m,i} &< 0, \\ \dot{\mathbf{p}}(\mathbf{p}_{m,i}) &= 0, \end{aligned} \quad (4)$$

i.e. the line through $\mathbf{p}_i, \mathbf{p}_{i+1}$ being the only stable set of equilibrium points. By defining $\mathbf{A}_{\parallel i}$ and $\mathbf{b}_{\parallel i}$ as

$$\begin{aligned} \mathbf{A}_{\parallel i} &= \mathbf{0}, \\ \mathbf{b}_{\parallel i} &= \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{t_{i+1} - t_i}, \end{aligned} \quad (5)$$

it suffices (3). A heuristic approach is used to find a solution that suffices (4). By remembering that

$$\dot{\mathbf{p}} = \mathbf{A}_{\perp i} \mathbf{p} + \mathbf{b}_{\perp i}, \quad (6)$$

one can create the equation system

$$\begin{aligned} \dot{\mathbf{p}}(\mathbf{p}_{m,i}) &= 0, \\ \dot{\mathbf{p}}(\mathbf{p}_{m,i} + \mathbf{a}_{1,i}) &= 0, \\ \dot{\mathbf{p}}(\mathbf{p}_{m,i} + \mathbf{a}_{q,i}) &= -\kappa_i \mathbf{a}_{q,i}, \quad \forall q \in \{2, \dots, m\} \end{aligned} \quad (7)$$

and solve it for the variables in $\mathbf{A}_{\perp i}$ and $\mathbf{b}_{\perp i}$ in order to construct the orthogonal vector field. The strength parameter $\kappa_i > 0$ adjusts the influence of the orthogonal vector field $V_{\perp i}$, accounting for the rate at which any object converges back to the trajectory after a disturbance. Note that the norm of any vector $\mathbf{v}_{\perp i}$ caused the vector field $V_{\perp i}$ is directly proportional to the minimal distance $d_{m,i}$ to the line through $\mathbf{p}_{i+1}, \mathbf{p}_i$ as

$$\|\mathbf{v}_{\perp i}\| = d_{m,i} \kappa_i. \quad (8)$$

When moving along a trajectory, an interpolation scheme according to (9) with the weighting factor $w_i \in [0, 1]$ blending over the vector fields of subsequent trajectory segments is proposed in order to avoid discontinuities

$$\dot{\mathbf{p}} = w_i(\mathbf{A}_i + \mathbf{b}_i) + (1 - w_i)(\mathbf{A}_{i+1} + \mathbf{b}_{i+1}). \quad (9)$$

The trajectory segment τ_i is defined by the two points $\mathbf{p}_i, \mathbf{p}_{i+1}$, the trajectory segment τ_{i+1} by $\mathbf{p}_{i+1}, \mathbf{p}_{i+2}$. Let the projection of an object position onto the τ_i segment be $\mathbf{p}_{p,i}$ and its relative length $r_i = \frac{\|\mathbf{p}_{p,i} - \mathbf{p}_i\|}{\|\mathbf{p}_{i+1} - \mathbf{p}_i\|}$. The interpolation scheme weight w_i is given as

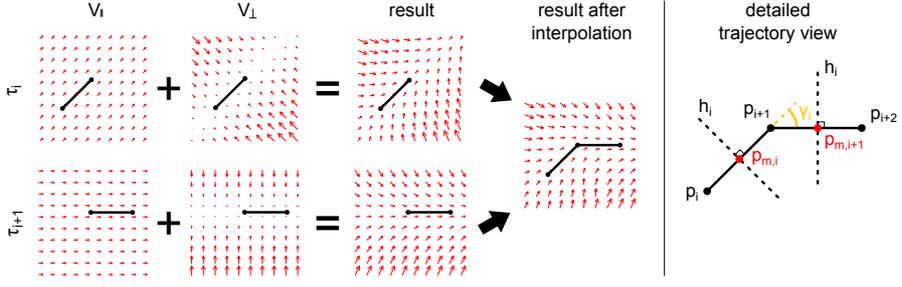


Fig. 1 Vector field example for a trajectory consisting of two segments τ_i and τ_{i+1} . Left side: Calculated vector field both after superposition and split up into individual parallel and orthogonal vector fields. Right side: Schematic view.

$$\begin{aligned}
 r'_i &= |r_i - 0.5|, \\
 r'_{i+1} &= |r_{i+1} - 0.5|, \\
 w_i &= \frac{r'_{i+1}}{r'_i + r'_{i+1}},
 \end{aligned} \tag{10}$$

with the offset of 0.5 moving the moment of switching to the hyperplane h_i . Note that this interpolation scheme is only well defined for intersection angles γ_i

$$\gamma_i = \arccos \left(\frac{(\mathbf{p}_{i+2} - \mathbf{p}_{i+1}) \cdot (\mathbf{p}_{i+1} - \mathbf{p}_i)}{\|\mathbf{p}_{i+2} - \mathbf{p}_{i+1}\| \|\mathbf{p}_{i+1} - \mathbf{p}_i\|} \right) < \frac{\pi}{2}, \tag{11}$$

The approach can be readily combined with Laplacian Trajectory Editing [11], allowing one to deform the underlying trajectory proactively while still being able to recompute the new vector field quickly. For collision avoidance, one can use an arbitrary superimposing method, for example potential fields or the method in [7].

2.2 Convergence

Two aspects of convergence are investigated: Transition convergence (any point s either on the hyperplane h_i or in between h_i and h_{i+1} eventually converges to h_{i+1}) and point converge (any point on the last hyperplane h_{n-1} or beyond convergence to the last point \mathbf{p}_n). By combining both convergence properties one can thus ensure that a point moving along the trajectory eventually converges to \mathbf{p}_n .

To prove transition convergence, the vector field $\dot{\mathbf{p}}(s)$ always has to point in direction of \mathbf{n}_{i+1} as

$$\dot{\mathbf{p}}(s)^T \mathbf{n}_{i+1} > 0. \tag{12}$$

By decomposing $\dot{\mathbf{p}}(s)$ as

$$\dot{\mathbf{p}}(\mathbf{s}) = w_1(\mathbf{v}_{\parallel i} + \mathbf{v}_{\perp i}) + w_2(\mathbf{v}_{\parallel i+1} + \mathbf{v}_{\perp i+1}), \quad (13)$$

it accounts both for the orthogonal/parallel components of the vector fields V_i, V_{i+1} and a interpolation scheme with weights w_1, w_2 . Looking at the geometric properties, (12) can be rewritten as

$$\begin{aligned} w_1(\|\mathbf{v}_{\parallel i}\| \cos(\gamma_i) - \|\mathbf{v}_{\perp i}\| \sin(\gamma_i)) + w_2\|\mathbf{v}_{\parallel i+1}\| &> 0, \\ w_1(\|\mathbf{v}_{\parallel i}\| \cos(\gamma_i) - d_{m,i}\kappa_i \sin(\gamma_i)) + w_2\|\mathbf{v}_{\parallel i+1}\| &> 0. \end{aligned} \quad (14)$$

The minimal values of (14) are given for $w_1 = 1, w_2 = 0$, resulting in

$$\|\mathbf{v}_{\parallel i}\| \cos(\gamma_i) > d_{m,i}\kappa_i \sin(\gamma_i). \quad (15)$$

(15) is a necessary condition for transition convergence. Depending on the intersection angle γ_i it defines an admissible region around the trajectory segment whose extent can approach ∞ for $\gamma_i \rightarrow 0$ in the limiting case or 0 for $\gamma_i \rightarrow \frac{\pi}{2}$.

Point convergence for \mathbf{p}_n can be achieved by creating a vector field V for the last trajectory segment such that it fulfills

$$\begin{aligned} \lambda_{1,n} = \lambda_{2,n} = \dots = \lambda_{m,n} &< 0, \\ \dot{\mathbf{p}}(\mathbf{p}_n) &= 0, \end{aligned} \quad (16)$$

that is a point attractor for \mathbf{p}_n which can be created following the idea in (7). The proof is analogous to the one for transition convergence.

2.3 Spatial Bounds Approximation

Whereas conservative spatial bounds of the deviation of the integral curve C from the reference path can be given which hold for any $\kappa_i > 0$, a good approximation can be made under the assumption of $\gamma_i \approx 0$ and $\|\mathbf{v}_{\parallel i+1}\| \approx \|\mathbf{v}_{\parallel i}\|$. Then for an object starting from the hyperplane h_1 with an initial distance $d_{m,1}$ to the trajectory - see (8) - the resulting distance $d_{m,s}$ when passing the hyperplane h_s can be calculated as

$$\begin{aligned} d_{m,s} &= d_{m,1} - \int_C V_{\perp} dx, \\ &\approx d_{m,1} \prod_{i=1}^s \exp(-\kappa_i \Delta t_i), \\ &\approx d_{m,1} \prod_{i=1}^s \exp\left(-\kappa_i \frac{\|\mathbf{p}_{m,i+1} - \mathbf{p}_{m,i}\|}{\|\mathbf{v}_{\parallel i}\|}\right), \end{aligned} \quad (17)$$

with \int_C as the integral over the integral curve C and Δt_i as the time required to traverse from the hyperplane h_i to h_{i+1} .

3 Experimental Evaluation

The experiment consists of a movement imitation task where a button has to be pressed while avoiding an obstacle. To adapt to environmental changes, LTVF and Laplacian trajectory editing [11] are combined with a Programming by Demonstration (PbD) framework [10, 2]: Whereas the PbD framework produces a prototypic (i.e. regressed) trajectory, Laplacian trajectory editing is used to deform the prototypic trajectory in case of large environmental changes. Subsequently, LTVF are used to create movements converging to the prototypic trajectory in case of small disturbances (e.g. a slightly varied starting position).

The human hand movement is recorded at a frame rate of 200Hz using a Vortex motion capture system. Five demonstration rollouts are performed in which the hand is moved from various starting positions through a narrow tunnel to press a button, see Fig 2a. All demonstrations are aligned in the temporal domain using Dynamic Time Warping. Then Gaussian Mixture Regression gives a prototypic trajectory with corresponding spatial covariance matrix Σ_i for the i -th sampling point. The spatial variance is used to determine the strength parameter $\kappa_i \propto \frac{1}{\sqrt{\det \Sigma_i}}$ of the vector field. This ensures quick convergence towards the trajectory wherever the spatial variance is low, corresponding to constrained sections of the trajectory. On the other hand a high variance implicitly assumes free space, resulting in low values of κ_i . Shown in Fig 2b are the five rollouts (orange trajectories). The colorful tube is centered around the prototypic trajectory. The color of the tube represents the vector field strength κ_i whereas the tube diameter is directly proportional to the distance $d_{m,s}$ in (17). The result is a funnel in which convergence according to the spatial bounds approximation can be guaranteed. Adaption to a varied environment (varied obstacle position, different button to press) can be achieved by retargeting the trajectory using Laplacian trajectory editing, see Fig 2c. Fig 2d shows the reproduction of only two trajectories from different starting positions using a HRP4 robotic platform.

4 Discussion

Experiments show how the presented approach can be readily combined with other approaches and applied to robotic movement imitation problems. Differing from other approaches based on Gaussian mixture models which have to learn the mixture model for every new trajectory, no learning is necessary when using LTVF. This makes the approach applicable for real-time applications where online trajectory modifications are necessary. Because at any time instance the vector field has to be computed based only on two adjacent trajectory segments, its computational complexity is independent of the shape or length of the trajectory. Moreover, tight spatial boundaries can be specified which are helpful for collision prediction and avoidance. A drawback is that methods based on Gaussian mixture models like Dynamic Movement Primitives have useful theoretical properties in terms of global

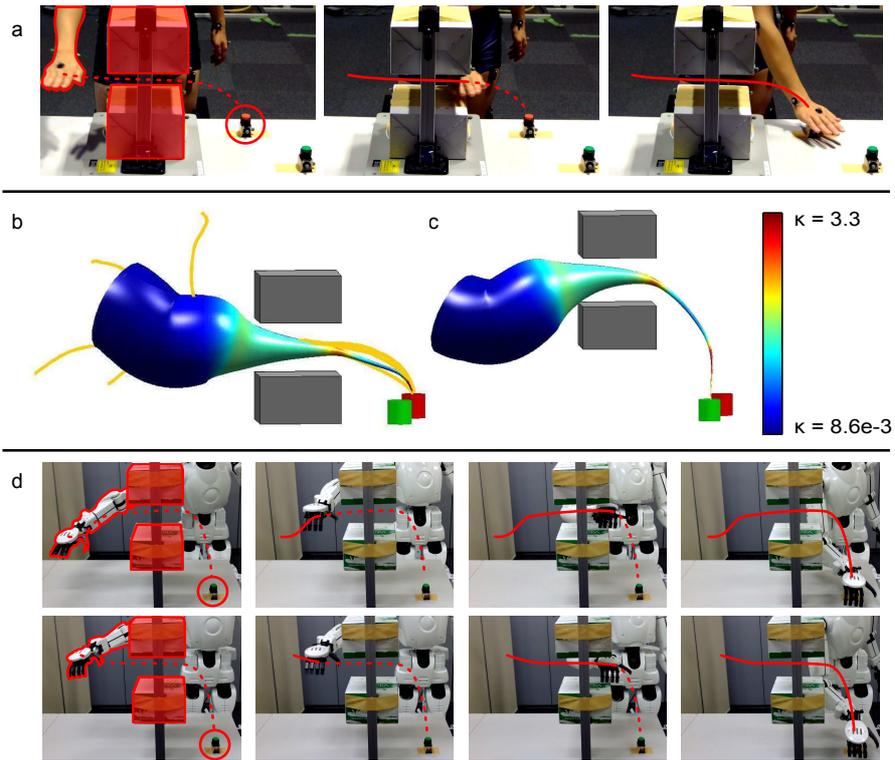


Fig. 2 Experimental results. Human demonstration (a), processed results (b), adaption to a changed environment (c) and adapted movement using a HRP-4 robot (d). Highlighted in (a) and (d) in red are the human/robot arm, two boxes representing the obstacle, the button to be pressed and the hand trajectory

stability and robustness to disturbances. Convergence for LTVF on the other hand can be shown only on a local scale. Hence any disturbance large enough may lead to divergent behavior, making it necessary to recompute the reference trajectory. Differing from Gaussian mixture models which are generally time-independent, the presented approach is time-variant as one has to keep track of the object's current trajectory segment. This can be used to resolve issues regarding self-intersecting paths.

5 Conclusion and Future Work

This paper presents a generic method to construct convergent vector fields towards a discretized reference trajectory. In case the trajectory is retargeted, the vector field can be reconstructed quickly without the need of an explicit optimization routine.

The guaranteed convergence, low computational complexity and large spatial adaption range makes it predestined to be used with other approaches for collision avoidance and vector field adaption.

Future work will be focused on an improved version extending the presented approach to continuous trajectories.

Acknowledgement

This work is supported in part within the DFG excellence research cluster *Cognition for Technical Systems - CoTeSys* (www.cotesys.org) and by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (S), 2008-2012, 20220001, “Establishing Human-Machine Communication through Kinestiology and Linguistics Integration” (PI: Y. Nakamura)

References

1. J. Barraquand, B. Langlas, and J. C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22:224–241, 1992.
2. S. Calinon, F. D’Halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard. Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression. *IEEE Robotics and Automation Magazine*, 17(2):44–54, 2010.
3. Sylvain Calinon, Florent D’halluin, Eric L. Sauser, Darwin G. Caldwell, and Aude Billard. Learning and reproduction of gestures by imitation. *IEEE Robotics and Automation Magazine*, 17(2):44–54, 2010.
4. J. Ernesti, L. Righetti, M. Do, T. Asfour, and S. Schaal. Encoding of periodic and their transient motions by a single dynamic movement primitive. In *IEEE-RAS Humanoids*, pages 57–64, 2012.
5. Vinicius Mariano Gonçalves, Luciano C. A. Pimenta, Carlos Andrey Maia, Bruno C. O. Dutra, and Guilherme A. S. Pereira. Vector fields for robot navigation along time-varying curves in n -dimensions. *IEEE Transactions on Robotics*, 26(4):647–659, 2010.
6. Micha Hersch, Florent Guenter, Sylvain Calinon, and Aude Billard. Dynamical system modulation for robot learning via kinesthetic demonstrations. *IEEE Transactions on Robotics*, 24(6):1463–1467, 2008.
7. Seyed Mohammad Khansari-Zadeh and Aude Billard. A dynamical system approach to real-time obstacle avoidance. *Auton. Robots*, 32(4):433–454, 2012.
8. Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.
9. S. Lindemann and S. La Valle. Smoothly blending vector fields for global robot navigation. In *IEEE CDC*, pages 3353–3559, 2005.
10. Manuel Muehlig, Michael Gienger, Sven Hellbach, Jochen J. Steil, and Christian Goerick. Task-level imitation learning using variance-based movement optimization. In *IEEE ICRA*, pages 1177–1184, 2009.
11. Thomas Nierhoff and Sandra Hirche. Fast trajectory replanning using Laplacian mesh optimization. In *ICARCV*, pages 154–159, 2012.
12. Antonio Pistillo, Sylvain Calinon, and Darwin G. Caldwell. Bilateral physical interaction with a robot manipulator through a weighted combination of flow fields. In *IEEE IROS*, pages 3047–3052, 2011.