IEEE Control Systems Letters paper presented at
2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Rigid Motion Gaussian Processes with $SE(3)$ Kernel and Application to Visual Pursuit Control*

Marco Omainska[1] and Junya Yamauchi[1] and Armin Lederer[2] and Sandra Hirche[2] and Masayuki Fujita[1]

***Abstract*— We address the learning of unknown rigid body motions in the Special Euclidian Group $SE(3)$ based on Gaussian Processes. A new covariance kernel for $SE(3)$ is presented and proven to be a valid kernel for Gaussian Process Regression. The learning error of the proposed Gaussian Process model is extended to a high-probability statement on $SE(3)$. We employ it in a visual pursuit scenario of a moving target with unknown velocity in 3D space. Our approach is validated in a simulated 3D environment in Unity, and shows significant better prediction accuracy than the most commonly used Gaussian kernel. When compared to other covariance kernels proposed on $SE(3)$, its advantages are a natural extension of covering numbers on $SE(3)$, that it is computationally more efficient, and that stability of target pursuit can be guaranteed without limiting the target rotational space to $SO(2)$.**

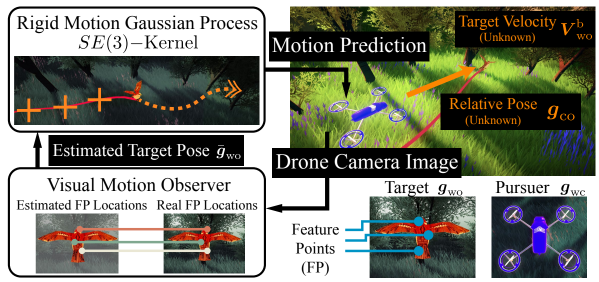

Fig. 1. Our kernel achieves computational efficiency and high prediction accuracy in a pursuit setting. `https://youtu.be/yf2JhwhPAoA`

I. Introduction

Data-driven modelling approaches gain popularity as for a rising number of problems for autonomous systems exact mathematical models become intractable. This especially holds for control tasks in robotics that have to rely on visual information of their environment [1], [2], for which data is readily available [3]. Scenarios include aerial swarm robotics [4], and visual tracking in traffic and animal ecology [5], [6].

Many of these scenarios require the tracking of objects (targets) on the Special Euclidian group $SE(3)$, that means the object position, rotation, and their respective velocities are crucial to the task. To that regard, a wide range of motion estimators [7]–[11] have been presented. While the *Visual Motion Observer* in [7] comes without the requirement of a target motion model, it suffers from an estimation error that eventually leads to target loss. One credible remedy for this risk is to adopt a data-driven mechanism [8]–[11].

A vast variety of data-driven modelling techniques is available such as Support Vector Machines and Gaussian Mixture Models [3], [12]. For modelling complex dynamics on $SE(3)$, Neural Networks [13] and Gaussian Process (GP) Regression [11], [14], [15] are a popular choice. GPs have the advantage that they provide a mean estimate and variance to measure model fidelity, but many works require prior knowledge of a bounded RKHS norm of the modelled function which is not realistic depending on the application [16]. As pointed out in [14], usual GP models are only defined in Euclidean space, and satisfactory models can be obtained only in limited situations on $SE(3)$. This is challenging since it imposes hard limitations on the kernel choice, but high fidelity models are required to minimize the risk of target loss. Under these requirements, generalizations of GPs to manifolds have also been attempted [17], [18].

A common choice to represent rotations is by Euler angles $\boldsymbol{g} = [x\ y\ z\ \alpha\ \beta\ \gamma]^\intercal \in \mathbb{R}^6$, or as an axis-angle vector $\boldsymbol{g} = [x\ y\ z\ \boldsymbol{\xi}^\intercal\theta]^\intercal \in \mathbb{R}^6$. While these have vector space structures and standard kernel choices (e.g. squared exponential) can be applied, they often lead to innacurate predictions at high angular speeds for sparse training data [14]. A new axis-angle kernel was proposed in [14], but it does not translate well in our pursuit scenario as performance guarantees depend on a worst-case rotational error [11]. Since it also holds issues with the uncertainty prediction, [19] proposed a new kernel based on a dual-quaternion representation of $\boldsymbol{g}$. However, it comes at the expense of an increased computational complexity and GP training failures as the topology of quaternions is sensitive to hyperparameter changes [14]. So far, there is no kernel available for the homogeneous form of $\boldsymbol{g}$ despite its wide usage in robotics [1], [2], [7].

The main contributions of this letter are as follows:

(i) Developing a kernel for the homogeneous form of $\boldsymbol{g}$ for GP Regression and proving its validity.
(ii) Extending the notion of covering numbers to $SE(3)$ to derive a new high-probability statement for the learning error based on Lipschitz continuity on $SE(3)$.
(iii) Deriving an online-computable performance bound, stability, and validation in a 3D simulation (Fig. 1).

Notation: Vectors/matrices are denoted as bold lower/upper case characters (except $\boldsymbol{V}^{\mathrm{b}}, \boldsymbol{g}$ to keep to literature [1], [2], [7]). $\wedge$ computes the cross product $\hat{\boldsymbol{a}}\boldsymbol{b} = \boldsymbol{a}\times\boldsymbol{b}$, $\boldsymbol{a},\boldsymbol{b} \in \mathbb{R}^3$, with $\vee$ the inverse-operation. $\mathrm{diag}(\cdot)$ is a diagonal matrix, $\|\cdot\|$ the Euclidean norm. $a^{\{i:j\}}$ are elements i to j from a series.

*This work was supported by JSPS KAKENHI Grant Number 20K14761, and the European Research Council (ERC) Consolidator Grant "Safe data-driven control for human-centric systems (CO-MAN)" under grant agreement number 864686.

[1]M. Omainska, J. Yamauchi and M. Fujita are with the Department of Information Physics & Computing, The University of Tokyo, Japan marcoomainska@g.ecc.u-tokyo.ac.jp

[2]A. Lederer and S. Hirche are with the Department of Electrical and Computer Engineering, Technical University of Munich, Germany

II. Problem Setting

Translational and rotational motion of rigid bodies form together the special Euclidian group $SE(3) := \mathbb{R}^3 \times SO(3)$. Motion dynamics on this space can take several forms [2, Ch. 2], however, in this letter we adopt the target object tracking technique from [7], which uses the homogeneuous (matrix) representation of $\boldsymbol{g} \in SE(3)$ and body velocity $\hat{\boldsymbol{V}}^{\mathrm{b}} \in se(3) := \left\{\left[\begin{smallmatrix} \hat{\boldsymbol{\omega}}^{\mathrm{b}} & \boldsymbol{v}^{\mathrm{b}} \\ \boldsymbol{0} & 0 \end{smallmatrix}\right] \middle| \hat{\boldsymbol{\omega}}^{\mathrm{b}} \in \mathbb{R}^{3\times 3}, (\hat{\boldsymbol{\omega}}^{\mathrm{b}})^{\intercal} = -\hat{\boldsymbol{\omega}}^{\mathrm{b}}, \boldsymbol{v}^{\mathrm{b}} \in \mathbb{R}^3\right\}$ as

$$\dot{\boldsymbol{g}} = \boldsymbol{g}\hat{\boldsymbol{V}}^{\mathrm{b}}, \quad \boldsymbol{g} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{p} \\ \boldsymbol{0} & 1 \end{bmatrix}, \tag{1}$$

with position $\boldsymbol{p} \in \mathbb{R}^3$ and rotation $\boldsymbol{R} \in SO(3) := \{\boldsymbol{R} \in \mathbb{R}^{3\times 3} \mid \boldsymbol{R}\boldsymbol{R}^{\intercal} = \boldsymbol{I}_3, \det(\boldsymbol{R}) = 1\}$. For elements on $se(3)$, operator $\vee$ extracts the translational $\boldsymbol{v}^{\mathrm{b}}$ and angular velocity $\boldsymbol{\omega}^{\mathrm{b}}$, with $\wedge$ the inverse operation. Thus, left-transitioning $\boldsymbol{g}$ in (1),

$$\boldsymbol{V}^{\mathrm{b}} = (\boldsymbol{g}^{-1}\dot{\boldsymbol{g}})^{\vee} = \begin{bmatrix} \boldsymbol{v}^{\mathrm{b}\intercal} & \boldsymbol{\omega}^{\mathrm{b}\intercal} \end{bmatrix} \in \mathbb{R}^6, \quad \boldsymbol{v}^{\mathrm{b}}, \boldsymbol{\omega}^{\mathrm{b}} \in \mathbb{R}^3, \tag{2}$$

we have found a Euclidian vector space structure to represent velocity on $SE(3)$. We are interested in modelling (2) by Gaussian Proccesses in terms of a velocity field of the form

$$\boldsymbol{f} : SE(3) \to \mathbb{R}^6, \tag{3}$$

that means, the velocity (2) takes a mapping $\boldsymbol{g} \mapsto \boldsymbol{V}^{\mathrm{b}}(\boldsymbol{g})$. Targets following a velocity field is a frequent problem [8], [14], [19]. However, since the space $SE(3)$ is non-Euclidian, the challenge is to find in Sec. III a valid kernel function

$$\mathrm{k} : SE(3) \times SE(3) \to \mathbb{R} \tag{4}$$

in order to have a well-defined Gaussian distribution. Thereafter, we derive a high-probability statement for the learning error on a compact space $\mathbb{G} \subset SE(3)$ which only requires Lipschitz continuity of (2). Lastly, we apply the new findings to a visual pursuit scenario in Sec. IV, prove stability, and validate it in a virtual environment (Sec. V).

III. Rigid Motion Gaussian Process

A. Gaussian Process Regression

Let us consider (3) as a regression problem of this form:

Assumption 1 ([16])**.** The unknown dynamics $\boldsymbol{f}(\cdot)$ are samples from a Gaussian Process $\boldsymbol{f}(\boldsymbol{g}) \sim \mathcal{GP}(\boldsymbol{0}, \mathrm{k}(\boldsymbol{g}, \boldsymbol{g}'))$ and the observations $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{g}) + \boldsymbol{\epsilon}$ are perturbed by zero mean i.i.d. Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma_n^2 \boldsymbol{I}_6)$ with variance $\sigma_n^2 > 0$.

We stack the observations in a training dataset of N data points $\mathcal{D} = \{(\boldsymbol{g}^{\{i\}}, \boldsymbol{y}^{\{i\}})\}_{i=1}^N$. GP models are defined by a covariance ("kernel") function (4) and a prior mean. The latter is set to zero in this work, which is common to simplify calculation without loss of generality [15], [16]. Under these conditions, the posterior distribution $\boldsymbol{f}(\boldsymbol{g}^*)$ at a test point $\boldsymbol{g}^* \in SE(3)$ is jointly Gaussian distributed with the mean and covariance function being

$$\boldsymbol{\mu} = [\mu_1, \ldots, \mu_6]^{\intercal} \in \mathbb{R}^6, \tag{5a}$$

$$\mu_i(\boldsymbol{g}^*) = \mathbf{k}_{\boldsymbol{\varphi}_i}^{\intercal}(\boldsymbol{g}^*)\mathbf{A}_{\boldsymbol{\varphi}_i}\boldsymbol{y}_i^{\{1:N\}}, \tag{5b}$$

$$\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_6^2) \in \mathbb{R}^{6\times 6}, \tag{5c}$$

$$\sigma_i^2(\boldsymbol{g}^*) = \mathrm{k}(\boldsymbol{g}^*, \boldsymbol{g}^*) - \mathbf{k}^{\intercal}(\boldsymbol{g}^*)\mathbf{A}_{\boldsymbol{\varphi}_i}\mathbf{k}(\boldsymbol{g}^*). \tag{5d}$$

The remaining terms are defined as follows: the Gram matrix $[\mathbf{K}_{\boldsymbol{\varphi}_i}]_{j,j'} = \mathrm{k}_{\boldsymbol{\varphi}_i}(\boldsymbol{g}^{\{j\}}, \boldsymbol{g}^{\{j'\}})$ for $j, j' \in \{1, \ldots, N\}$ with $\mathbf{A}_{\boldsymbol{\varphi}_i} := (\mathbf{K}_{\boldsymbol{\varphi}_i} + \sigma_n^2 \boldsymbol{I}_N)^{-1}$ encodes the similarity between data points in $\mathcal{D}$, whereas the extended covariance function $[\mathbf{k}_{\boldsymbol{\varphi}_i}(\boldsymbol{g}^*)]_j = \mathrm{k}_{\boldsymbol{\varphi}_i}(\boldsymbol{g}^{\{j\}}, \boldsymbol{g}^*)$ calculates the similarity between a test point and the dataset. The index $\boldsymbol{\varphi}_i$ denotes the hyperparameters for output $i = 1, \ldots, 6$ which are used to tune the kernel (4) for a better model performance. To define a valid Gaussian Process distribution, (4) must be a *valid* kernel function. We will define what consitutes to the validity of a kernel function on $SE(3)$ next, and later in Sec. III-C we introduce our final kernel.

B. Distance Metric on $SE(3)$

To measure the similarity between two poses $\boldsymbol{g}$ and $\boldsymbol{g}'$ we need to define a distance measure on $SE(3)$. We know from [20] that this metric can be a trade-off between translations and orientations by choosing appropriate length scales. For two weights $\rho_{\boldsymbol{p}}, \rho_{\boldsymbol{R}} \geq 0$ satisfying $\rho_{\boldsymbol{p}} + \rho_{\boldsymbol{R}} = 1$, we define the distance as the root over the sum of squares

$$d_{SE(3)}(\boldsymbol{g}, \boldsymbol{g}') = \sqrt{\rho_{\boldsymbol{p}}\|\boldsymbol{p} - \boldsymbol{p}'\|^2 + \rho_{\boldsymbol{R}} d_{SO(3)}^2(\boldsymbol{R}, \boldsymbol{R}')} \tag{6}$$

with the rotational distance $d_{SO(3)} : SO(3) \times SO(3) \to \mathbb{R}^+$ yet to be designed. Further, let us introduce the following:

Definition 1 ([12])**.** Let $\mathcal{X}$ be a non-empty set. A real-valued symmetric function $\mathrm{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive definite (pd) kernel if and only if the Gram matrix $\mathbf{K} \in \mathbb{R}^{N\times N}$ satisfies $\boldsymbol{c}^{\intercal}\mathbf{K}\boldsymbol{c} \geq 0$ for any vector $\boldsymbol{c} \in \mathbb{R}^N$. If $\boldsymbol{c}^{\intercal}\mathbf{K}\boldsymbol{c} \geq 0$ only holds for $\boldsymbol{c} \in \mathbb{R}^N$ with $\sum_{i=1}^N c_i = 0$, then k is called a conditionally positive definite (cpd) kernel.

Literature [18], [20] provides a vast variety of distance metrics on $SO(3)$, though, not all in the form of rotation matrices $\boldsymbol{R}$. We are in favor of the Frobenius-Norm, that, for a given matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, is defined as $\|\boldsymbol{A}\|_F = \sqrt{\mathrm{tr}(\boldsymbol{A}^{\intercal}\boldsymbol{A})}$. Thus, for the remainder of this work, let

$$d_{SO(3)}(\boldsymbol{R}, \boldsymbol{R}') = \tfrac{1}{2}\|\boldsymbol{R} - \boldsymbol{R}'\|_F \tag{7}$$

be the distance between two rotations $\boldsymbol{R}, \boldsymbol{R}' \in SO(3)$. [18] also provides other valid kernels, though, advantages of (7) are simultaneously satisfying a high regression performance and fast computability, interpretability of covering numbers on $SO(3)$ (Sec. III-E), and being able to calculate a pursuit performance bound (Sec. IV). Make (7) a cpd kernel:

Lemma 1. *The negative squared distance function* (7)*, i.e.* $-d_{SO(3)}^2(\boldsymbol{R}, \boldsymbol{R}') = -\frac{1}{4}\|\boldsymbol{R} - \boldsymbol{R}'\|_F^2$*, is a cpd kernel.*

Proof: Direct consequence of [18, Lem. 5.5], since $\|\cdot\|_F$ defines a matrix inner product space $\langle\cdot,\cdot\rangle_F$. ∎

C. Kernel on $SE(3)$

The class of cpd kernels generalize the feature space representation of pd kernels as it does not need to be a dot product [12, Ch. 2.4]. Still, for Gaussian Process Regression the kernel (4) must be pd to be a valid covariance function. Though, the distance (7) can still be used as follows:

Theorem 1. *Consider the $SE(3)$-distance metric* (6) *with the Frobenius-Distance* (7) *on $SO(3)$. Then, for all hyperparameters $\boldsymbol{\varphi}_i = [\sigma_{f_i}, l_i]$ satisfying $\sigma_{f_i} > 0$, $l_i > 0$, the kernel $\mathrm{k}_{\boldsymbol{\varphi}_i} : SE(3) \times SE(3) \to \mathbb{R}^+$,*

$$\mathrm{k}_{\boldsymbol{\varphi}_i}(\boldsymbol{g}, \boldsymbol{g}') = \sigma_{f_i}^2 \exp\Big(-\tfrac{d_{SE(3)}^2(\boldsymbol{g},\boldsymbol{g}')}{2l_i^2}\Big), \tag{8}$$

is a valid kernel for Gaussian Process Regression.

Proof: From (6) and $\exp(a+b)=\exp(a)\exp(b)$, $a,b\in\mathbb{R}$, we get $\mathrm{k}(\boldsymbol{g},\boldsymbol{g}')=\sigma_f^2\exp(-\frac{\rho_p}{2l^2}\|\boldsymbol{p}-\boldsymbol{p}'\|^2)\exp(-\frac{\rho_R}{2l^2}d_{SO(3)}^2(\boldsymbol{R},\boldsymbol{R}'))$. The first exponential is the well-known squared-exponential kernel, which has been already proven to be a valid kernel and therefore pd. From [12, Prop. 2.28] and Lemma 1 we also conclude that the second exponential is pd because the $SO(3)$ distance (7) is cpd. Since the finite product of pd kernels is also pd [3, p. 296], the Theorem is proven. ■

The hyperparameters in (8) adjust the probability of the mean of function samples in the Gaussian distribution, and optimal values are typically obtained by evidence maximization. Note that whereas the weights in (6) are typically set application-dependent, it is viable to include them in $\boldsymbol{\varphi}_i$.

Remark 1. It may be possible to extend (8) to the Matérn class, which requires the use of *Bochner's Theorem* [12].

D. Lipschitz Bounds

Let $\big(SE(3), d_{SE(3)}\big)$ be a metric space. To compute a uniform error bound between the real target and estimated motion later in Sec. IV, we require Lipschitz continuity of the unknown function (3), which is a weak assumption for many control systems. With the special vectorized form of $\boldsymbol{g}$

$$\mathrm{vec}(\boldsymbol{g}) \coloneqq \begin{bmatrix}\boldsymbol{p}\\ \mathrm{sk}(\boldsymbol{R})^\vee\end{bmatrix}, \quad \mathrm{sk}(\boldsymbol{R}) = \frac{1}{2}(\boldsymbol{R} - \boldsymbol{R}^\intercal), \tag{9}$$

we are ready to note the following:

Lemma 2. *Suppose Assumption 1 holds and that $\boldsymbol{f}(\cdot)$ is Lipschitz continuous $|f_i(\boldsymbol{g}) - f_i(\boldsymbol{g}')| \leq L_{f_i} \cdot d_{SE(3)}(\boldsymbol{g},\boldsymbol{g}')$, $\forall \boldsymbol{g},\boldsymbol{g}' \in SE(3)$, $\forall i \in \{1,\ldots,6\}$ with Lipschitz constants L_{f_i}. Further, let $\boldsymbol{R}$, $\boldsymbol{R}'$ be* close, *that means, $\boldsymbol{R}^\intercal\boldsymbol{R}' \succ 0$. Then, there exists a Lipschitz constant $L_{\boldsymbol{f}}$ such that*

$$\|\boldsymbol{f}(\boldsymbol{g}) - \boldsymbol{f}(\boldsymbol{g}')\| \leq L_{\boldsymbol{f}} \big\|\mathrm{vec}\big(\boldsymbol{g}^{-1}\boldsymbol{g}'\big)\big\|.$$

Proof: Assuming $\boldsymbol{R}^\intercal\boldsymbol{R}' \succ 0$, then from [7, Prop. 5.3 (ii)] we know that $\frac{1}{4}\|\boldsymbol{R} - \boldsymbol{R}'\|_F^2 \leq \|\,\mathrm{sk}(\boldsymbol{R}^\intercal\boldsymbol{R}')^\vee\|^2$. The statement follows then straightforward by inserting all terms. ■ Note that Lemma 2 only assumes the existance of a Lipschitz constant, but its value does not need to be known. In fact, [16, Thm. 3.2] provides a high-probability Lipschitz estimate based on the observation data in $\mathcal{D}$ satisfying Assumption 1.

E. Learning Error

To make qualitative statements in machine learning it is crucial to quantify a learning error. For Gaussian Processes this comes in the form of a probabilistic uniform error bound [16], [21]. The classical approach in [21] requires prior knowledge of a bounded RKHS norm of $\boldsymbol{f}$. However, in our visual pursuit scenario (Sec. IV) of targets with unknown motion $\boldsymbol{f}$ this is paradoxal, even if we restrict ourselves to universal kernels to at least assume the existance of such a bound [10], [11], [15]. Instead, we will derive a probabilistic uniform error bound based on our previous assumption of Lipschitz continuity on the metric space $\big(SE(3), d_{SE(3)}\big)$, which our kernel (8) already fulfills by design:

Lemma 3. *Consider the GP model* (5) *based on the covariance kernel* (8) *on a compact set $\mathbb{G} \subset SE(3)$. Furthermore, consider a continuous unknown dynamics $\boldsymbol{f}\colon \mathbb{G} \to \mathbb{R}^6$ with Lipschitz constants L_{f_i} on $\big(\mathbb{G}, d_{SE(3)}\big)$, and $N \in \mathbb{N}$ observations satisfying Assumption 1. Then, the posterior mean and variance of the Gaussian Process conditioned on the training data $\mathcal{D}$ are continuous with Lipschitz constants L_{μ_i} and $L_{\sigma_i^2}$ $\forall i \in \{1,\ldots,6\}$ on $\mathbb{G}$, respectively, where*

$$L_{\mu_i} \leq \frac{\sigma_{f_i}^2}{l_i^2}\bar{\rho}\sqrt{N}\Big\|\mathbf{A}_{\boldsymbol{\varphi}_i}\boldsymbol{y}_i^{\{1:N\}}\Big\|, \quad L_{\sigma_i^2} \leq 2\tau\bar{\rho}\frac{\sigma_{f_i}^2 + \sigma_{f_i}^4 N\|\mathbf{A}_{\boldsymbol{\varphi}_i}\|}{l_i^2}$$

and $\bar{\rho} \coloneqq \max\{\rho_{\boldsymbol{p}}, \rho_{\boldsymbol{R}}\}$. Also, pick $\delta \in (0,1)$, $\tau \in \mathbb{R}_+$ with

$$\begin{aligned}\beta(\tau) &= \sqrt{2\log\Big(\tfrac{M(\tau,\mathbb{G})}{\delta}\Big)}, \quad \boldsymbol{\gamma}(\tau) = [\gamma_1,\ldots,\gamma_6]^\intercal \\ \gamma_i(\tau) &= (L_{\mu_i} + L_{f_i})\tau + \beta(\tau)\sqrt{L_{\sigma_i^2}\tau}\end{aligned} \tag{10}$$

where $M(\tau,\mathbb{G})$ denotes the minimum number (τ-covering number of $\mathbb{G}$) such that there exists a set $\mathbb{G}_\tau$ satisfying $|\mathbb{G}_\tau| = M(\tau,\mathbb{G})$ and $\forall \boldsymbol{g} \in \mathbb{G}$ there exists $\boldsymbol{g}' \in \mathbb{G}_\tau$ with $d_{SE(3)}(\boldsymbol{g},\boldsymbol{g}') \leq \tau$. Then, the following probabilistic uniform error bound holds for $\Delta(\boldsymbol{g}) = \|\boldsymbol{f}(\boldsymbol{g}) - \boldsymbol{\mu}(\boldsymbol{g})\|$:

$$\Pr\{\forall \boldsymbol{g}\in\mathbb{G},\ \Delta(\boldsymbol{g}) \leq \beta(\tau)\|\boldsymbol{\Sigma}^{1/2}(\boldsymbol{g})\|_F + \|\boldsymbol{\gamma}(\tau)\|\} \geq (1-\delta)^6$$

Proof: The one dimensional case in [16, Thm. 3.1] considered a Euclidian metric space, but the proof can be easily modified with our metric space by straightforward replacing all Euclidian distances by our distance $d_{SE(3)}$, and kernel (8) Lipschitz constant $\bar{\rho}\sigma_{f_i}^2/l_i^2$. Since $\boldsymbol{\epsilon}$ is uncorrelated, by intersection and triangle inequality for the multi dimensional case

$$\begin{aligned}&\Pr\{\forall \boldsymbol{g}\in\mathbb{G}, |f_1(\boldsymbol{g}) - \mu_1(\boldsymbol{g})| \leq \beta(\tau)\sigma_1(\boldsymbol{g}) + \gamma_1(\tau)\ \cap\cdots\cap \\ &\qquad |f_6(\boldsymbol{g}) - \mu_6(\boldsymbol{g})| \leq \beta(\tau)\sigma_6(\boldsymbol{g}) + \gamma_6(\tau)\} \geq (1-\delta)^6 \\ \Leftrightarrow\quad &\Pr\{\forall \boldsymbol{g} \in \mathbb{G}, \|\boldsymbol{f}(\boldsymbol{g}) - \boldsymbol{\mu}(\boldsymbol{g})\| \leq \\ &\qquad \|\boldsymbol{\Sigma}^{1/2}(\boldsymbol{g})[\beta(\tau)\ \ldots\ \beta(\tau)]^\intercal + \boldsymbol{\gamma}(\tau)\|\} \geq (1-\delta)^6\end{aligned}$$

the uniform error bound from Lemma 3 is obtained. ■

The right-hand side of the probability-inequality stems from the regression problem due to measurement noise. Also, the covering number $M(\tau,\mathbb{G})$ represents the minimum number of points in a grid over $\mathbb{G}$ with grid constant τ to fully cover the space. However, its calculation on $\mathbb{G}$ is a non-trivial problem since $d_{SE(3)}(\boldsymbol{g},\boldsymbol{g}') \leq \tau$ forms hyperellipsoids

$$\sqrt{\rho_{\boldsymbol{p}}\Big(\tfrac{\|\boldsymbol{p}-\boldsymbol{p}'\|}{\tau}\Big)^2 + \rho_{\boldsymbol{R}}\Big(\tfrac{d_{SO(3)}(\boldsymbol{R},\boldsymbol{R}')}{\tau}\Big)^2} \leq 1. \tag{11}$$

Lemma 4. *Let the same conditions as in Lemma 3 hold. With the maximum extension $r_i \coloneqq \max|\boldsymbol{p}_i - \boldsymbol{p}_i'|$ in each dimension x, y, z, the covering number on $\mathbb{G}$ is upper-bounded by*

$$M(\tau,\mathbb{G}) \leq \Big(1 + \sqrt{\rho_{\boldsymbol{R}}}\tfrac{2\sqrt{2}}{\tau}\Big)^2 \prod_{i=\{x,y,z\}} \Big(1 + \sqrt{\rho_{\boldsymbol{p}}}\frac{r_i}{\tau}\Big). \tag{12}$$

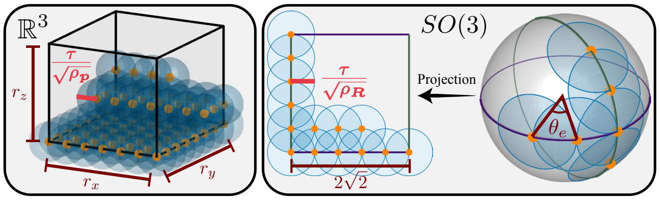

Fig. 2. Computing covering number $M(\tau, \mathbb{G})$ on compact $\mathbb{G} \subset SE(3)$.

Proof: From [7, p. 93], we can conclude $\frac{1}{2}\|\boldsymbol{R} - \boldsymbol{R}'\|_F = \frac{1}{2}\|\boldsymbol{I}_3 - \boldsymbol{R}^\intercal \boldsymbol{R}'\|_F = \sqrt{1-\cos\theta_e}$. It measures the distance between $\boldsymbol{R}, \boldsymbol{R}' \in SO(3)$ where θ_e is the rotation angle between both rotations. Since $0 \leq \sqrt{1-\cos\theta_e} \leq \sqrt{2}$ and because every rotation on the sphere can be viewed in terms of spherical coordinates (ϕ, ψ), we can construct a rectangle space $[-\sqrt{2}, \sqrt{2}] \times [-\sqrt{2}, \sqrt{2}]$ that includes every rotation (see Fig. 2). From (11), the grid points become circles with radius $\tau/\sqrt{\rho_{\boldsymbol{R}}}$ and $\tau/\sqrt{\rho_{\boldsymbol{p}}}$. Hence, the number of grid points in one dimension of the rotational rectangle space is $1+\sqrt{\rho_{\boldsymbol{R}}}\frac{2\sqrt{2}}{\tau}$, whereas for the position in one dimension it is $1+\sqrt{\rho_{\boldsymbol{p}}}\frac{r_i}{\tau}$. In this form, $\mathbb{G}$ can be over-approximated [16] by a hyperrectangle set $\tilde{\mathbb{G}}$ whose covering number $M(\tau, \tilde{\mathbb{G}})$ is in multiplicative relation to the number of grid points in each dimension. The statement then follows from $M(\tau, \mathbb{G}) \leq M(\tau, \tilde{\mathbb{G}})$. ■

Remark 2. Despite that Lemma 4 only calculates an upper-bound on $M(\tau, \mathbb{G})$, the logarithm and square root in (10) keep any effect of conservatism small. That means, this bound can be readily applied to real applications (see Sec. V).

IV. Application to Visual Target Tracking

A. Relative Motion Estimation

In this section we explain how to perform target tracking. Let us denote three coordinate frames: a world frame Σ_{w}, camera frame Σ_{c}, and target frame Σ_{o}. The indices i, j on $\boldsymbol{g}_{ij}$ and $\boldsymbol{V}^{\mathrm{b}}_{ij}$ define the pose and velocity of a frame Σ_j as measured from another frame Σ_i. By this definition, the pose of the target as seen from the camera can be calculated as $\boldsymbol{g}_{\mathrm{co}} = \boldsymbol{g}_{\mathrm{wc}}^{-1}\boldsymbol{g}_{\mathrm{wo}}$. From the time derivative and (1), we obtain

$$\dot{\boldsymbol{g}}_{\mathrm{co}} = -\hat{\boldsymbol{V}}^{\mathrm{b}}_{\mathrm{wc}}\boldsymbol{g}_{\mathrm{co}} + \boldsymbol{g}_{\mathrm{co}}\hat{\boldsymbol{V}}^{\mathrm{b}}_{\mathrm{wo}} \tag{13}$$

the so-called *relative rigid body motion* model. Note that by our pursuit scenario neither $\boldsymbol{V}^{\mathrm{b}}_{\mathrm{wo}}, \boldsymbol{g}_{\mathrm{wo}}, \boldsymbol{g}_{\mathrm{co}}$ are measurable, but the camera can infer its own velocity $\boldsymbol{V}^{\mathrm{b}}_{\mathrm{wc}}$ and pose $\boldsymbol{g}_{\mathrm{wc}}$.

Our goal is to estimate the target velocity $\hat{\boldsymbol{V}}^{\mathrm{b}}_{\mathrm{wo}}$ and pose $\boldsymbol{g}_{\mathrm{co}}$ (and $\boldsymbol{g}_{\mathrm{wo}}$, consequently) since they are not directly measurable. Thus, based on (13) let us introduce the *Visual Motion Observer* (VMO) [7, Ch. 6]

$$\dot{\bar{\boldsymbol{g}}}_{\mathrm{co}} = -\hat{\boldsymbol{V}}^{\mathrm{b}}_{\mathrm{wc}}\bar{\boldsymbol{g}}_{\mathrm{co}} - \bar{\boldsymbol{g}}_{\mathrm{co}}\hat{\boldsymbol{u}}_e \tag{14}$$

with observer input $\boldsymbol{u}_e$ and motion estimate $\bar{\boldsymbol{g}}_{\mathrm{co}}$. Further, let $\boldsymbol{g}_{ee} := \bar{\boldsymbol{g}}_{\mathrm{co}}^{-1}\boldsymbol{g}_{\mathrm{co}}$ be the estimation error and its vectorized form $\boldsymbol{e}_e := \mathrm{vec}(\boldsymbol{g}_{ee})$. However, it cannot be measured since it is dependent on the target pose $\boldsymbol{g}_{\mathrm{co}}$ as seen from the camera.

To solve this issue, we want to infer $\boldsymbol{e}_e$ from the 2D images of the camera. More specifically, we assume knowledge of at least $n_f \geq 4$ target feature points (FPs) $\mathfrak{p}_{\mathrm{o}}^{\{i\}} \in \mathbb{R}^3$, $i = 1, \dots, n_f$ in frame Σ_{o} that are collected in a dataset $\mathbb{F}_{n_f} = \{\mathfrak{p}_{\mathrm{o}}^{\{i\}}\}_{i=1}^{n_f}$. A subset $\tilde{\mathbb{F}} \subseteq \mathbb{F}_{n_f}$ of these FPs are projected onto the image plane with $\lambda > 0$ the camera focal length at

$$\mathfrak{f}_i = \frac{\lambda}{\mathfrak{p}_{\mathrm{c},2}^{\{i\}}}\begin{bmatrix}\mathfrak{p}_{\mathrm{c},1}^{\{i\}}\\ \mathfrak{p}_{\mathrm{c},3}^{\{i\}}\end{bmatrix} \in \mathbb{R}^2, \qquad \begin{bmatrix}\mathfrak{p}_{\mathrm{c}}^{\{i\}}\\ 1\end{bmatrix} = \boldsymbol{g}_{\mathrm{co}}\begin{bmatrix}\mathfrak{p}_{\mathrm{o}}^{\{i\}}\\ 1\end{bmatrix}. \tag{15}$$

These FPs are collected in a visual measurement vector $\mathfrak{f} = [\mathfrak{f}_1^\intercal \ \dots \ \mathfrak{f}_{|\tilde{\mathbb{F}}|}^\intercal]^\intercal \in \mathbb{R}^{2|\tilde{\mathbb{F}}|}$ and can be detected by real time Computer Vision techniques such as classical methods described in [2, Ch. 4 & 11] or Neural Networks [22]. Based on the projection model (15), if we replace $\boldsymbol{g}_{\mathrm{co}}$ with the estimate $\bar{\boldsymbol{g}}_{\mathrm{co}}$, we can also obtain the *estimated* FP locations $\bar{\mathfrak{f}}_i \in \mathbb{R}^{2|\tilde{\mathbb{F}}|}$ on the image plane. Suppose that at all times at least 4 FPs are detected (i.e. $|\tilde{\mathbb{F}}| \geq 4$) and the following holds:

Assumption 2. For the estimated rotation error $\boldsymbol{R}_{ee} \succ 0$ holds. That means, $|\theta_{ee}(t)| \leq \pi/2, \forall t \geq 0$ of $\boldsymbol{R}_{ee} = e^{\hat{\boldsymbol{\xi}}\theta_{ee}}$.

Then, the estimation error is in multiplicative relation to the displacement of the detected and estimated FP locations [1], $\boldsymbol{e}_e = \boldsymbol{J}^\dagger(\mathfrak{f} - \bar{\mathfrak{f}})$, where $\boldsymbol{J}^\dagger$ denotes the pseudo-inverse of the image jacobian $\boldsymbol{J}$ which can be calculated from [7, p. 108].

B. Data-driven Visual Pursuit

Our goal is to bring the drone closer to the target. Mathematically speaking, we want the estimation error $\boldsymbol{e}_e$ and control error $\boldsymbol{e}_c := \mathrm{vec}(\boldsymbol{g}_{ce})$ with $\boldsymbol{g}_{ce} := \boldsymbol{g}_{\mathrm{d}}^{-1}\bar{\boldsymbol{g}}_{\mathrm{co}}$ to be *small*, where $\boldsymbol{g}_{\mathrm{d}}$ is a *desired* constant relative pose. Note that the control error is based on the estimation $\bar{\boldsymbol{g}}_{\mathrm{co}}$ since the real relative pose $\boldsymbol{g}_{\mathrm{co}}$ is not measureable. In summary, for a given nonnegative constant b, we seek a control law that achieves $\lim_{t\to\infty}\|\boldsymbol{e}(t)\| < b$ with $\boldsymbol{e} = [\boldsymbol{e}_c^\intercal \ \boldsymbol{e}_e^\intercal]^\intercal$.

Identically to the target motion model (1), let the drone motion model be given by $\dot{\boldsymbol{g}}_{\mathrm{wc}} = \boldsymbol{g}_{\mathrm{wc}}\hat{\boldsymbol{V}}^{\mathrm{b}}_{\mathrm{wc}}$, and $\boldsymbol{u}_c$ be the input to the drone velocity $\boldsymbol{V}^{\mathrm{b}}_{\mathrm{wc}} := -\mathbf{Ad}_{(\boldsymbol{g}_{\mathrm{d}})}\boldsymbol{u}_c$. We are modelling the target motion in terms of $\boldsymbol{f}(\boldsymbol{g}_{\mathrm{wo}}) = \boldsymbol{V}^{\mathrm{b}}_{\mathrm{wo}}(\boldsymbol{g}_{\mathrm{wo}})$, that means, our data takes the form $\mathcal{D} = \{(\boldsymbol{g}_{\mathrm{wo}}^{\{i\}}, \boldsymbol{y}^{\{i\}})\}_{i=1}^N$, $\boldsymbol{y} = \boldsymbol{V}^{\mathrm{b}}_{\mathrm{wo}}(\boldsymbol{g}_{\mathrm{wo}}) + \boldsymbol{\epsilon}$, as given by Assumption 1. However, we design our controller with the GP mean prediction $\boldsymbol{\mu}(\bar{\boldsymbol{g}}_{\mathrm{wo}})$ based on the *estimated* target pose $\bar{\boldsymbol{g}}_{\mathrm{wo}} := \boldsymbol{g}_{\mathrm{wc}}\bar{\boldsymbol{g}}_{\mathrm{co}}$ as

$$\boldsymbol{u} = \begin{bmatrix}\boldsymbol{u}_c\\ \boldsymbol{u}_e\end{bmatrix} = -\boldsymbol{K}\mathbf{N}\boldsymbol{e}_e - \begin{bmatrix}\mathbf{Ad}_{(\boldsymbol{R}_{ce})}\mathbf{Ad}_{(\boldsymbol{R}_{ee})}\\ \mathbf{Ad}_{(\boldsymbol{R}_{ee})}\end{bmatrix}\boldsymbol{\mu}(\bar{\boldsymbol{g}}_{\mathrm{wo}}) \tag{16}$$

with controller gains $\boldsymbol{K} = \mathrm{diag}(k_c\boldsymbol{I}_6, k_e\boldsymbol{I}_6)$, $k_c > 0$, $k_e > 0$. The other terms are given as follows ($\mathbf{Ad}_{(\boldsymbol{R})} := \mathbf{Ad}_{(\boldsymbol{p}=\boldsymbol{0},\boldsymbol{R})}$):

$$\mathbf{Ad}_{(\boldsymbol{g})} := \begin{bmatrix}\boldsymbol{R} & \hat{\boldsymbol{p}}\boldsymbol{R}\\ \boldsymbol{0} & \boldsymbol{R}\end{bmatrix}, \quad \mathbf{N} := \begin{bmatrix}\boldsymbol{I}_6 & \boldsymbol{0}\\ -\mathbf{Ad}_{(\boldsymbol{R}_{ce}^\intercal)} & \boldsymbol{I}_6\end{bmatrix}.$$

Before we prove that the controller (16) indeed results in stable target tracking, we require one more assumption:

Assumption 3. For the control rotation error $\boldsymbol{R}_{ce} \succ 0$ holds. That means, $|\theta_{ce}(t)| \leq \pi/2, \forall t \geq 0$ of $\boldsymbol{R}_{ce} = e^{\hat{\boldsymbol{\xi}}\theta_{ce}}$.

This assumption is in general satisfied for the given target pursuit scenario since the drone must be able to move faster than the target. Finally, let us state the following theorem:

Theorem 2. *Assume $\boldsymbol{V}^{\mathrm{b}}_{\mathrm{wo}}(\cdot)$ admits a Lipschitz constant L on $(\mathbb{G}, d_{SE(3)})$ with compact field $\mathbb{G} \subset SE(3)$ and let $N \in \mathbb{N}$ observations be given that satisfy Assumption 1. Suppose that Assumptions 2 and 3 hold, and that $\kappa := \min\{k_c, k_e\} - L_{\boldsymbol{f}} > 0$. Then, the controller (16) guarantees with $\tau > 0$ and current estimate $\bar{\boldsymbol{g}}_{\mathrm{wo}}$ that the error $\boldsymbol{e} = [\boldsymbol{e}_c^\intercal\ \boldsymbol{e}_e^\intercal]^\intercal$ converges with a probability higher than $(1-\delta)^6$ to $\Pr\{\|\boldsymbol{e}\| \leq b_{\mathrm{var}}(\bar{\boldsymbol{g}}_{\mathrm{wo}}, \delta, \tau)\} \geq (1-\delta)^6$ where*

$$b_{\mathrm{var}}(\bar{\boldsymbol{g}}_{\mathrm{wo}}, \delta, \tau) := \frac{\beta(\tau)\|\boldsymbol{\Sigma}^{1/2}(\bar{\boldsymbol{g}}_{\mathrm{wo}})\|_F + \|\boldsymbol{\gamma}(\tau)\|}{\kappa}. \tag{17}$$

Proof: We reuse from [7], [11] the storage function $S := \frac{1}{2}\sum_{j=\{c,e\}}\left(\|\boldsymbol{p}_{je}\|^2 + \mathrm{tr}(\boldsymbol{I}_3 - \boldsymbol{R}_{je})\right)$ whose time derivative is known as $\dot{S} = \boldsymbol{e}^\intercal \mathbf{N}^\intercal \boldsymbol{u} + \boldsymbol{e}^\intercal [\mathbf{0}\ \ \mathbf{Ad}^\intercal_{(\boldsymbol{R}_{ee})}]^\intercal \boldsymbol{V}^{\mathrm{b}}_{\mathrm{wo}}(\boldsymbol{g}_{\mathrm{wo}})$. Inserting the controller (16), and since $-\boldsymbol{e}^\intercal \mathbf{N}^\intercal \boldsymbol{K} \mathbf{N} \boldsymbol{e} \leq -\lambda_{\boldsymbol{K}}\|\boldsymbol{e}\|^2$ for $\lambda_{\boldsymbol{K}} := \min\{k_c, k_e\}$, we obtain:

$$\begin{aligned}\dot{S} &= -\lambda_{\boldsymbol{K}}\|\boldsymbol{e}\|^2 + \boldsymbol{e}^\intercal[\mathbf{0}\ \mathbf{Ad}^\intercal_{(\boldsymbol{R}_{ee})}]^\intercal\left(\boldsymbol{V}^{\mathrm{b}}_{\mathrm{wo}}(\boldsymbol{g}_{\mathrm{wo}}) - \boldsymbol{\mu}(\bar{\boldsymbol{g}}_{\mathrm{wo}})\right)\\ &\leq -\lambda_{\boldsymbol{K}}\|\boldsymbol{e}\|^2 + \|\boldsymbol{e}\|\left(\|\boldsymbol{V}^{\mathrm{b}}_{\mathrm{wo}}(\boldsymbol{g}_{\mathrm{wo}}) - \boldsymbol{V}^{\mathrm{b}}_{\mathrm{wo}}(\bar{\boldsymbol{g}}_{\mathrm{wo}})\| + \right.\\ &\qquad\qquad \left.\|\boldsymbol{V}^{\mathrm{b}}_{\mathrm{wo}}(\bar{\boldsymbol{g}}_{\mathrm{wo}}) - \boldsymbol{\mu}(\bar{\boldsymbol{g}}_{\mathrm{wo}})\|\right)\\ &\leq -\|\boldsymbol{e}\|\left((\lambda_{\boldsymbol{K}} - L_{\boldsymbol{f}})\|\boldsymbol{e}\| - \|\boldsymbol{V}^{\mathrm{b}}_{\mathrm{wo}}(\bar{\boldsymbol{g}}_{\mathrm{wo}}) - \boldsymbol{\mu}(\bar{\boldsymbol{g}}_{\mathrm{wo}})\|\right)\end{aligned} \tag{18}$$

where we used Lemma 2 (since Assumption 2 holds). From Lemma 3 we see that for a probability higher than $(1-\delta)^6$ that $\dot{S} < 0, \forall \|\boldsymbol{e}\| \leq b_{\mathrm{var}}(\bar{\boldsymbol{g}}_{\mathrm{wo}}, \delta, \tau)$. ■

The computability of (17) is crucial for online learning scenarios to decide if data shall be added to the GP model and/or forgotten. It can be made arbitrarily small by either increasing k_c and k_e, or by increasing the number of data points in $\mathcal{D}$ to decrease $\|\boldsymbol{\Sigma}^{1/2}(\cdot)\|_F$. We can also prove that it is upper-bounded in terms of an ultimate bound, i.e. $b_{\mathrm{var}} \leq b$:

Corollary 1. *Let the same conditions as in Theorem 2 hold. Then, there exists $\zeta(\delta, \tau) > 0$, $T(\delta) > 0$ and $\tau > 0$ such that $\Pr\{\|\boldsymbol{e}\| \leq b(\delta, \tau), \forall t \geq T(\delta)\} \geq (1-\delta)^6$ with the ultimate bound being for any $\eta \in (0, 1)$*

$$b := \frac{\sqrt{2}\Delta_{\boldsymbol{f}}}{\kappa\eta}, \quad \Delta_{\boldsymbol{f}}(\tau) := \beta(\tau)\max_{\boldsymbol{g}\in\mathbb{G}}\|\boldsymbol{\Sigma}^{1/2}(\boldsymbol{g})\|_F + \|\boldsymbol{\gamma}(\tau)\| \tag{19}$$

Proof: From (18), Lemma 3, and constant η we obtain $\dot{S} \leq -\kappa(1-\eta)\|\boldsymbol{e}\|^2 - \kappa\eta\|\boldsymbol{e}\|^2 + \Delta_{\boldsymbol{f}}\|\boldsymbol{e}\|$ which holds for a probability higher than $(1-\delta)^6$. Therefore, if we define a set $\mathbb{E} := \left\{\boldsymbol{e} \in \mathbb{R}^{12} \,|\, \|\boldsymbol{e}\| \geq \zeta, \boldsymbol{R}_{ee} \succ 0, \boldsymbol{R}_{ce} \succ 0\right\}$ for $\zeta(\delta, \tau) := \Delta_{\boldsymbol{f}}/\eta\kappa$, it holds that $\Pr\{\dot{S} < 0, \forall \boldsymbol{e} \in \mathbb{E}\} \geq (1-\delta)^6$. We conclude from [23] that the error is uniformly ultimately bounded in probability with the ultimate bound following from $\alpha_1^{-1}(\alpha_2(\zeta(\delta, \tau))) = \sqrt{2}\zeta(\delta, \tau)$, where $\alpha_1(\|\boldsymbol{e}\|) := \frac{1}{2}\|e\|^2$ and $\alpha_2(\|\boldsymbol{e}\|) := \|e\|^2$ are class $\mathcal{K}$ functions such that $\alpha_1 \leq S \leq \alpha_2$. This completes the proof. ■

When $\boldsymbol{V}^{\mathrm{b}}_{\mathrm{wo}}$ is perfectly predictable, (19) approaches zero, significantly different from the results obtained in [7]. This is due to the construction of a GP model on $SE(3)$.

V. Simulation Results

In this section, we investigate the computational demand and prediction accuracy of these kernels: The popular Squared Exponential kernel which takes the poses in translation and axis-angle form ("$SE(3)$-RBF"), $SE(3)$-Kernel from [14] ("$SE(3)$-Axang"), and our $SE(3)$-Kernel (8) ("$SE(3)$-Hom"). The tests run all on a MacBook Pro, 2.3 GHz 8-Core Intel Core i9 with 32GB RAM.

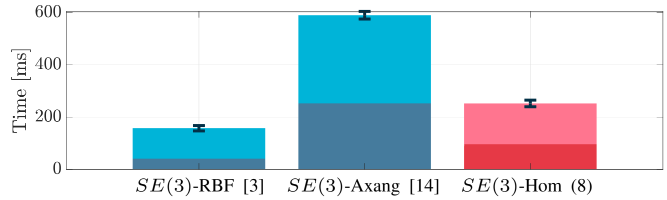

Fig. 3. Computation times of Gram matrix $\mathbf{K}_{\boldsymbol{\varphi}_i}$ for $N = 1000$ random data points (bottom) and prediction of 100 random data points (top). Averaged over 100 runs, with standard deviation given as black handle.

A. Kernel Runtime Comparison

We generate $N = 1000$ random poses $\boldsymbol{g}$ that are presented once in the translation and axis-angle form, and the homogeneous form (1). The former will be used for the $SE(3)$-RBF and $SE(3)$-Axang kernels, whereas the latter will be parsed to our homogeneous form $SE(3)$-Hom kernel. We compute the Gram matrix $\mathbf{K}_{\boldsymbol{\varphi}_i}$ for all three kernels, and then do a prediction of 100 randomly selected points. It is repeated for 100 times to get reliable results. Figure 3 depicts the average computation time and standard deviation between all runs. We observe that $SE(3)$-Hom is 60% faster than $SE(3)$-Axang. This result however does not take into account the prediction quality, which we will now investigate next.

B. Digital Twin Simulation

We will now evaluate our theoretical result in a simulated 3D forest environment[1] using Unity, whilst the control logic resides in MATLAB. Both sides communicate over a ROS layer with a message frequency of 50 Hz. The target is represented by a *bird* whose dynamics are given by a modified quartic oscillator (see Fig. 4) with $v = 1.5$, $\epsilon = 0.25$ as

$$\begin{aligned}\boldsymbol{V}^{\mathrm{b}}_{\mathrm{wo}} &= \begin{bmatrix}\boldsymbol{R}^\intercal_{\mathrm{wo}}\boldsymbol{v}^{\mathrm{b}}_{\mathrm{wo}}\\ \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{wo}}\end{bmatrix}, \quad \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{wo}} = \begin{bmatrix}0\\ 0\\ \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{atan2}(\boldsymbol{v}^{\mathrm{b}}_{\mathrm{wo}y}, \boldsymbol{v}^{\mathrm{b}}_{\mathrm{wo}x})\end{bmatrix}\\ \boldsymbol{v}^{\mathrm{b}}_{\mathrm{wo}} &= v\begin{bmatrix}\boldsymbol{p}_{\mathrm{wo}y}\\ \epsilon(-\boldsymbol{p}^3_{\mathrm{wo}x} + \boldsymbol{p}_{\mathrm{wo}x})\\ \cos\left(\mathrm{atan2}(\boldsymbol{p}_{\mathrm{wo}y}, \boldsymbol{p}_{\mathrm{wo}x}) - \frac{\pi}{4}\right)\end{bmatrix}.\end{aligned} \tag{20}$$

The angular velocity $\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{wo}}$ results in the bird always heading towards the direction of movement. Also, let $\rho_{\boldsymbol{p}} = \rho_{\boldsymbol{R}} = 0.5$.

1) Setup: We select $N = 6$ data points on the bird trajectory Fig. 4 and obtain optimal GP hyperparameters by evidence maximization [3]. Then, with $k_e = k_c = 12$, and approximated $L_{\boldsymbol{f}} \leq 4$, Corollary 1 guarantees from $\kappa > 0$ stability of our pursuit control scheme. We run the simulation for $T = 15$ s with the initial positions $\boldsymbol{p}_{\mathrm{wo}} = [-2, 0, 0]^\intercal$, $\boldsymbol{p}_{\mathrm{wc}} = [-2, -3, -3]^\intercal$, $\boldsymbol{p}_{\mathrm{co}} = [0, 2, 0]^\intercal$, $\boldsymbol{p}_{\mathrm{d}} = [0, 3, 0]^\intercal$, and initial rotations $\boldsymbol{\xi}\theta_{\mathrm{wo}} = [0, 0, 1]^\intercal \cdot 0$, $\boldsymbol{\xi}\theta_{\mathrm{wc}} = [1, 0, 0]^\intercal \cdot (-\frac{\pi}{4})$, $\boldsymbol{\xi}\theta_{\mathrm{co}} = [1, 0, 0]^\intercal \cdot \frac{\pi}{4}$, $\boldsymbol{\xi}\theta_{\mathrm{d}} = [1, 0, 0]^\intercal \cdot \frac{\pi}{4}$, with $\boldsymbol{R}(\boldsymbol{\xi}\theta) = e^{\hat{\boldsymbol{\xi}}\theta}$.

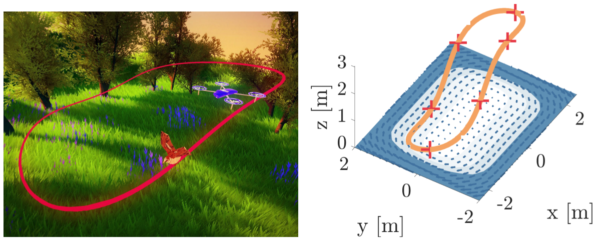

Fig. 4. Left: forest simulation environment with "bird" target and "drone" pursuer. Right: Trajectory with $N=6$ datapoints "+".

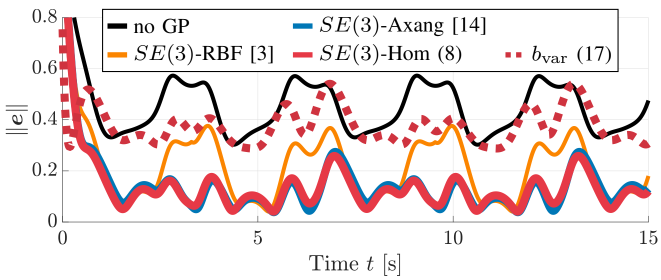

Fig. 5. Target tracking performance for different kernels. The dotted line indicates the bound (17) for $\delta = 0.001$, $\tau = 0.001$ when using $SE(3)$-Hom.

2) Results: The results are shown in Fig. 5. Clearly, using a GP model outperforms the conventional target tracking technique from [7], and both $SE(3)$-Axang and $SE(3)$-Hom perform better than $SE(3)$-RBF. The latter is due to wrong predictions at high angular speeds due to our sparse data (Fig. 4), resulting from the erroneous rotational distance [11], [14], [19]. In contrast, $SE(3)$-Hom does not show any significant increase in pursuit performance when compared to $SE(3)$-Axang. The advantage of $SE(3)$-Hom is the availability of the online-computable performance bound (17), stability guarantee without limiting the rotational space to $SO(2)$ [11], the natural extension for covering numbers on $SE(3)$ to obtain the in our setting less restrictive Bayesian-based high-probability statement Lemma 3, and its fast computability. The bound (17) clearly demonstrates the advantage of our method in a worst-case sense, since once entered, the error for $SE(3)$-Hom stays under it at all times, and shrinks with a higher GP model quality. This makes our method well suited for an online evasion learning and pursuit scheme that depends on the availability of such measures. Our simulation indicated that $SE(3)$-Hom is not prone to GP training failures (low sensitivity on hyperparameter changes in contrast to [19]), but they greatly influence the bound (17).

VI. Conclusion

In this letter, a Gaussian Process model for modelling rigid motions on $SE(3)$ is developed. A new $SE(3)$-kernel is proposed and proven valid that generalizes the GP input space to the homogeneous form $\boldsymbol{g} = \left[\begin{smallmatrix} \boldsymbol{R} & \boldsymbol{p} \\ \boldsymbol{0} & 1 \end{smallmatrix}\right]$. Further, we derive a high-probability statement on the GP learning error by extending the notion of covering numbers onto $SE(3)$. Our proposed data-driven controller is employed in a visual pursuit scenario of a moving target in 3D and outperforms alternative kernels on $SE(3)$ as it maintains both computational efficiency, prediction accuracy, and a computable worst-case performance.

[1]The code is made available here:
`https://github.com/marciska/vpc-rmgp-se3hom`

References

[1] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, 1st ed. Wiley, 2005.
[2] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry, *An Invitation to 3-D Vision*. Springer New York, 2004.
[3] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
[4] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 837–855, 2018.
[5] K. Shah, G. Ballard, A. Schmidt, and M. Schwager, "Multidrone aerial surveys of penguin colonies in antarctica," *Sci. Robot.*, vol. 5, no. 47, p. eabc3000, 2020.
[6] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, "A survey of vision-based traffic monitoring of road intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, pp. 2681–2698, 2016.
[7] T. Hatanaka, N. Chopra, M. Fujita, and M. W. Spong, *Passivity-Based Control and Estimation in Networked Robotics*. Springer, 2015.
[8] H. Wei, P. Zhu, M. Liu, J. P. How, and S. Ferrari, "Automatic pan–tilt camera control for learning Dirichlet-Process Gaussian-Process (DPGP) mixture models of multiple moving targets," *IEEE Trans. Autom. Control*, vol. 64, no. 1, pp. 159–173, 2019.
[9] G. Rotithor, D. Trombetta, R. Kamalapurkar, and A. P. Dani, "Full- and reduced-order observers for image-based depth estimation using concurrent learning," *IEEE Trans. Control Syst. Techno.*, vol. 29, no. 6, pp. 2647–2653, 2021.
[10] T. Adachi, N. Hayashi, and S. Takai, "Cooperative target tracking by multiagent camera sensor networks via Gaussian Process," *IEEE Access*, vol. 10, pp. 71 717–71 727, 2022.
[11] M. Omainska, J. Yamauchi, T. Fujinami, and M. Fujita, "Visual pursuit with switched motion estimation and rigid body Gaussian Processes," *Trans. Inst. Syst. Control Inf. Eng.*, vol. 36, no. 9, 2022, to appear.
[12] B. Schölkopf and A. J. Smola, *Learning with Kernels*. MIT Press, 2002.
[13] A. Byravan and D. Fox, "SE3-nets: Learning rigid body motion using deep neural networks," in *Proc. Int. Conf. Robot. Autom.*, 2017, pp. 173–180.
[14] M. Lang and S. Hirche, "Computationally efficient rigid-body Gaussian Process for motion dynamics," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1601–1608, 2017.
[15] T. Beckers, L. J. Colombo, M. Morari, and G. J. Pappas, "Learning-based balancing of model-based and feedback control for second-order mechanical systems," in *Proc. IEEE Conf. Decis. Control*, 2022, pp. 4667–4673.
[16] A. Lederer, J. Umlauft, and S. Hirche, "Uniform error bounds for Gaussian Process regression with application to safe control," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–11.
[17] R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth, "Manifold gaussian processes for regression," in *Proc. Int. Joint Conf. Neural Netw.*, 2016, pp. 3338–3345.
[18] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Kernel methods on riemannian manifolds with gaussian rbf kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2464–2477, 2015.
[19] M. Lang, M. Kleinsteuber, and S. Hirche, "Gaussian Process for 6-DoF rigid motions," *Auton. Robots*, vol. 42, no. 6, pp. 1151–1167, 2018.
[20] F. C. Park, "Distance metrics on the rigid-body motions with applications to mechanism design," *J. Mech. Des.*, vol. 117, no. 1, pp. 48–54, 1995.
[21] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for Gaussian Process optimization in the bandit setting," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.
[22] G. A. Kane, G. Lopes, J. L. Saunders, A. Mathis, and M. W. Mathis, "Real-time, low-latency closed-loop feedback using markerless posture tracking," *eLife*, vol. 9, p. e61909, 2020.
[23] H. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.