View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Archivio della ricerca- Università di Roma La Sapienza

World Academy of Science, Engineering and Technology
International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:10, No:6, 2016

Learning the Dynamics of Articulated Tracked Vehicles

Mario Gianni, Manuel A. Ruiz Garcia, Fiora Pirri

Abstract—In this work, we present a Bayesian non-parametric approach to model the motion control of ATVs. The motion control model is based on a Dirichlet Process-Gaussian Process (DP-GP) mixture model. The DP-GP mixture model provides a flexible representation of patterns of control manoeuvres along trajectories of different lengths and discretizations. The model also estimates the number of patterns, sufficient for modeling the dynamics of the ATV.

Keywords—Dirichlet processes, Gaussian processes, robot control learning, tracked vehicles.

International Science Index, Computer and Information Engineering Vol:10, No:6, 2016 waset.org/Publication/10004630

I. Introduction

DERIVING an explicit dynamic model for an Articulated Tracked Vehicles (ATVs) is a challenging task [1]. This task requires an accurate analysis of all the forces acting on the robot. Most of these forces are due to the interaction between the robot and the environment and can not be directly measurable due to the lack of suitable tactile sensors. Therefore, exact dynamics is not straightforward, since it is not possible to predict the exact motion of the vehicle only on the basis of the velocities of all the active mechanical components. Nevertheless, an effective dynamics approximation of the vehicle is crucial for both motion planning and control for real-time autonomous navigation.

In this work, we present a Bayesian non-parametric approach to model the motion control of ATVs (see Fig. 1). The main idea is to estimate the control manoeuvres to be applied to the robot to track a given trajectory from the previously observed trajectories of the robot, the terrain features associated with each trajectory and the control commands sent to the robot to follow such trajectories. The motion control model is based on a Dirichlet Process-Gaussian Process (DP-GP) mixture model [2], [3]. The DP-GP mixture model provides a flexible representation of patterns of control manoeuvres along trajectories of different lengths and discretizations. This representation allows us to group trajectories sharing either patterns of control manoeuvres or path segments. Finally, the model estimates the number of patterns, sufficient for modeling the dynamics of the ATV.

II. Related Work

Dynamic modeling is a key component of compliant and force control for complex robots, especially for actively articulated tracked robots [1]. However, due to unknown and hard to model non-linearity, analytic models of the dynamics for such systems are often only rough approximations. Nowadays, machine learning techniques are commonly

Mario Gianni is with the Sapienza University of Rome, Italy (e-mail: gianni@dis.uniroma1.it).

Fig. 1 Actively Articulated Tracked Vehicle Absolem, designed by ©Bluebotics [4] for USAR applications: This platform is equipped with the KINOVA Jarm Arm for pick and place opeartions

applied to significantly improve model-based control [5]. In this regard, a number of methods have been proposed combining contextual policy search (CPS) [6] with prior knowledge [7] and regression [8], [9]. CPS is a popular means for multi-task reinforcement learning in robotic control [6]. CPS learns a hierarchical policy, in which the lower-level policy is often a domain-specific behavior representation such as dynamical movement primitives (DMPs) [10]. Learning takes place on the upper-level policy that defines a distribution over the parameters of the lower-level policy for a given context. This context encodes properties of the environment or the task. CPS is typically based on local search based approaches such as regression. Locally Weighted Projection Regression (LWPR), introduced in [11], is a local model which approximates non-linear mappings in high-dimensional space. Its computational complexity depends linearly on the amount of the training instances. A drawback of this approach is the large number of free parameters which are hard to optimize. In [7], the authors introduced prior knowledge in order to increase the generalization properties of LWPR. A large portion of the literature is focused on employing kernel-based methods for the estimation of the inverse dynamics mapping by employing approaches, such as Gaussian Process Regression (GPR) and Support Vector Regression (SVR) [12]. Local Gaussian Process (LGP), introduced in [8], handles the problem of real-time learning by building local models on similar inputs, based on a distance metric and uses the Cholesky decomposition for incrementally updating the kernel matrix. In [13], the authors propose a real-time algorithm, dubbed SSGPR, which incrementally

International Science Index, Computer and Information Engineering Vol:10, No:6, 2016 waset.org/Publication/10004630

updates the model using GPR as learning method. The model is capable of learning non-linear mappings by using random features mapping for kernel approximation whose hyper-parameters are automatically updated. For the special case of relatively low-dimensional search spaces combined with an expensive cost function, which limits the number of evaluations of the cost functions, global search approaches, like Bayesian optimization are often superior, for instance for selecting hyper-parameters [14]. Bayesian optimization has been used for non-contextual policy search in robot grasping [15] and for locomotion tasks [16], [17]. The proposed approach for learning patterns of control manoeuvres for the tracked vehicle in Fig. 1 resorts to the main concepts underlying CPS. However, it differs from it by representing both the upper-level and the lower-level policies with a unified hierarchical model, defined by DP-GP mixture model where the number of upper-level policies sufficient for describing robot motions is also learned from data. Gibbs sampling [18] and a hybrid Monte Carlo technique [19] are applied to obtain estimates of the concentration upper-level policies and of the the hyper-parameters of the lower-level policies, respectively. A similar approach has been used for modeling non-linear dynamics of moving targets [2], [3].

III. The Motion Control Model of the ATV

Let us consider the ATV moving within a three-dimensional environment, negotiating rubbles, stairs and adapting each active sub-track to complex terrain surfaces. We assume that the robot is endowed with a 3D SLAM algorithm, which, over time, provides an estimation $\mathbf{q}(t) \in SE(3)$ of the robot pose within the map, with respect to a global reference frame. A trajectory $\boldsymbol{\tau}$ is a sequence of robot poses $\{\mathbf{q}(t)\}_{t=1}^{T}$, which denote a path between two different points of the environment (e.g., the sequence of robot poses along a staircase, which interconnects the basement of a building to its first floor). A descriptor $\boldsymbol{\varphi}(t) \in \mathbb{R}^d$ of the terrain features at a particular pose $\mathbf{q}(t)$ of a trajectory is a vector of real values specifying a measure of the heights z of the points of the 3D map, built by the SLAM algorithm. A 3D voxel grid is centered and oriented according to the pose $\mathbf{q}(t)$. The dimensions of the voxel grid is fixed according to the size of the ATV. We denote with $\boldsymbol{\phi}$ the set of all the terrain descriptors $\{\boldsymbol{\varphi}(t)\}_{t=1}^{T}$, obtained by sliding the voxel grid along the entire trajectory.

We denote with $\mathbf{c}$ a sequence of control manoeuvres $\mathbf{u}(t) \in \mathbb{R}^n$, for all $t = 1, \ldots, T$ (e.g., linear and angular velocity of the robot body, angular velocities of sub-tracks), applied to the ATV in order to suitably steer the robot along a trajectory and simultaneously adapt its morphology to the terrain. This said, let us assume that the robot has followed several paths $\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_N$ of different lengths, starting and ending to different locations in the map, possibly sharing either the same sequence of control manoeuvres or path segments. Let $\boldsymbol{\tau}_j$, $\boldsymbol{\phi}_j$ and $\mathbf{c}_j$ be the j-th tracked trajectory, the j-th set of terrain descriptors and the j-th sequence of control commands, respectively. The dynamics of the ATV can be modeled by the following non-linear system:

$$\mathbf{u}_j(t) = \mathbf{f}_i\left(\mathbf{q}_j(t), \boldsymbol{\varphi}_j(t)\right), \quad j = 1, \ldots, N. \tag{1}$$

Here $\mathbf{f}_i : \mathbb{R}^6 \times \mathbb{R}^d \mapsto \mathbb{R}^n$ are unknown continuous functions that code specific patterns of control manoeuvres such as falling down all the sub-tracks of the ATV for nose line climbing, lifting up the sub-tracks for riser climbing or for rotational motion within narrow passages. We refer to $\mathbf{f}_i(\cdot)$ as a the i-th *pattern of control manoeuvres*.

We assume that each pattern is drawn from a set $\mathcal{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_M\}$ of unknown continuous functions, whose number M is also unknown. Note that, there is no bijection between the set $\mathcal{F}$ of patterns and the set of all the tracked trajectories. In fact, one or more trajectories followed by the robot within the map of the environment may be described by the same pattern, while some pattern in $\mathcal{F}$ may not describe any tracked trajectory. Thus, in order to model the motion control of the ATV we have to determine the set $\mathcal{F}$ as well as the relation between the patterns $\mathbf{f}_i \in \mathcal{F}$, the trajectories $\boldsymbol{\tau}_j$, the set of terrain descriptors $\boldsymbol{\phi}_j$ and the sequences of $\mathbf{c}_j$ of control commands.

According to (1), each function $\mathbf{f}_i(\cdot)$ projects both the robot pose $\mathbf{q}_j(t)$ and the terrain descriptors $\boldsymbol{\varphi}_j(t)$ of the j-th trajectory to the control commands $\mathbf{u}_j(t)$, at time t. Therefore, each $\mathbf{f}_i(\cdot)$ can be viewed as a spatial phenomenon that can be modeled by a Gaussian Process GP_i, for any $i = 1, \ldots, M$, $\mathbf{q} \in SE(3)$ and $\boldsymbol{\varphi} \in \mathbb{R}^d$ [20]. Each GP_i is completely specified by a mean function $\mathbf{m}_i(\tilde{\mathbf{x}})$ and a positive semi-definite covariance function $\mathbf{k}_i(\tilde{\mathbf{x}}_p, \tilde{\mathbf{x}}_q)$, with $\tilde{\mathbf{x}} \coloneqq \begin{bmatrix}\mathbf{q}^\top & \boldsymbol{\varphi}^\top\end{bmatrix}^\top$. In this work, we assume that each GP_i has the standard squared exponential covariance function:

$$\mathbf{k}_i(\tilde{\mathbf{x}}_p, \tilde{\mathbf{x}}_q) = \sigma_i^2 \exp\left\{-\tfrac{1}{2}\left(\tilde{\mathbf{x}}_p - \tilde{\mathbf{x}}_q\right)^\top \boldsymbol{\Lambda}_i^{-1}\left(\tilde{\mathbf{x}}_p - \tilde{\mathbf{x}}_q\right)\right\} + \delta(\tilde{\mathbf{x}}_p, \tilde{\mathbf{x}}_q)\sigma_{w_i}^2$$

Here $\delta(\tilde{\mathbf{x}}_p, \tilde{\mathbf{x}}_q)$ is the Kronecker delta function. $\boldsymbol{\Lambda}_i \coloneqq \mathtt{diag}\left(\left[l_{i,1}^2, \ldots, l_{i,6+d}^2\right]\right)$ depends on the characteristic length-scales parameters $l_{i,h}$. The term σ_{w_i} represents within-pose variation (e.g., due to noisy measurements); the ratio of σ_{w_i} and σ_i weights the reflective effects of noise and influences from nearby poses. The above exponential covariance function encodes similarities between the tracked trajectories. If we denote with $\boldsymbol{\Sigma}_i$ the covariance matrix of each GP_i, with terms $\Sigma_{p,q} = \mathbf{k}_i(\tilde{\mathbf{x}}_p, \tilde{\mathbf{x}}_q)$, then, for each $\tilde{\mathbf{x}} \in \mathbb{R}^6 \times \mathbb{R}^d$, $\mathbf{f}_i(\tilde{\mathbf{x}}) \sim \mathrm{GP}_i(\mathbf{m}_i, \boldsymbol{\Sigma}_i)$, namely, GP_i is the distribution of the control commands over the workspace specified by the i-th function $\mathbf{f}_i(\cdot) \in \mathcal{F}$. Now, we have to build a relation that links a sequence $\mathbf{c}_j$ of robot control commands to a pattern $\mathbf{f}_i(\cdot)$. This relation has to be also able to capture differences resulting from trajectories $\boldsymbol{\tau}_j$ ending into different locations within the 3D map.

Let us consider a discrete random variable g_j ranging over a set $\mathcal{I} = \{1, \ldots, M\}$ of indices. The event $\{g_j = i\}$ represents the association of the sequence $\mathbf{c}_j$, applied to the robot to track the trajectory $\boldsymbol{\tau}_j$, having terrain descriptors set $\boldsymbol{\phi}_j$, with the i-th pattern $\mathbf{f}_i \in \mathcal{F}$, as shown in (1). We assume that $g_j \sim \mathrm{Categorical}(\boldsymbol{\pi})$, with $\boldsymbol{\pi} = \begin{bmatrix}\pi_1 & \ldots & \pi_M\end{bmatrix}$ prior probabilities of every possible outcome of g_j, for any $\mathbf{c}_j$, $\boldsymbol{\tau}_j$ and $\boldsymbol{\phi}_j$, with $j = 1, \ldots, N$. Given these prior probabilities, the probability of the j-th sequence of control commands, given

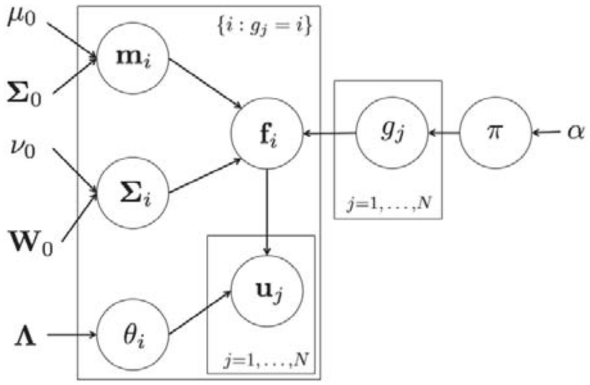

Fig. 2 Bayesian network of the motion contol model of the ATV

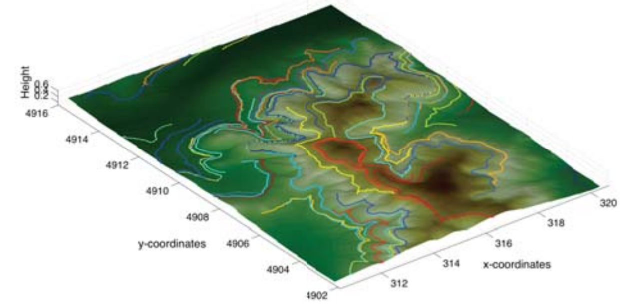

Fig. 3 Trajectories generated for training the motion control model over a simulated harsh terrain. Terrain surface is represented through a mesh

International Science Index, Computer and Information Engineering Vol:10, No:6, 2016 waset.org/Publication/10004630

the j-th tracked trajectory and of the observed j-th terrain descriptors set $\boldsymbol{\phi}_j$ is given by the following mixture model

$$\begin{aligned}&\Pr(\mathbf{u}_j^{1:T_j}|\mathbf{q}_j^{1:T_j}, \boldsymbol{\varphi}_j^{1:T_j}) = \\ &= \textstyle\sum_{i=1}^{M} \Pr(g_j = i|\boldsymbol{\pi})\Pr(\mathbf{u}_j^{1:T_j}|\mathbf{q}_j^{1:T_j}, \boldsymbol{\varphi}_j^{1:T_j}, \{\mathbf{m}_i, \boldsymbol{\Sigma}_i\})\end{aligned}$$

However, we can not know *a priori* how many patterns of control manoeuvres are sufficient for modeling the dynamics of the ATV. To cope with this issue we resort to a Dirichlet process (DP) mixture model to create an infinite mixture of patterns of control manoeuvres and to place a prior over the number of patterns. In this work, we choose a Gaussian process $\mathrm{GP}_0(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ as the base distribution of the DP. On the other hand, the support set of the process is chosen to be the set of all the admissible sequences of control commands for the robot. According to this model, the prior probability that a pair $\boldsymbol{\tau}_j$, $\boldsymbol{\phi}_j$ generates a sequence $\mathbf{c}_j$ of control commands, through an existing pattern $\mathbf{f}_i(\cdot)$ is $\Pr(g_j = i|g_{-j}, \alpha) = \frac{n_i}{N-1+\alpha}$. On the other hand, the probability that a pair $\boldsymbol{\tau}_j$, $\boldsymbol{\phi}_j$ is projected to a $\mathbf{c}_j$, through a new pattern $\mathbf{f}_i(\cdot)$ is $\Pr(g_j = M+1|g_{-j}, \alpha) = \frac{\alpha}{N-1+\alpha}$. Here, g_{-j} refers to the pattern assignments for the remaining trajectories, α is the concentration parameter of the DP, n_i is the number of pairs $\boldsymbol{\tau}_j$, $\boldsymbol{\phi}_j$ assigned to $\mathbf{f}_i(\cdot)$, N is the total number of observations and M is the number of the observed $\mathbf{f}_i(\cdot)$. To close, the dynamics of the articulated tracked robot can be represented by the following probabilistic motion control model

$$\begin{aligned}\boldsymbol{\pi} &\sim \mathrm{GEM}(\alpha)\\ \mathbf{m}_i &\sim \mathrm{GP}_0(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\\ \boldsymbol{\Sigma}_i &\sim \mathcal{W}^{-1}(\mathbf{W}_0, \nu_0)\\ g_j|\boldsymbol{\pi} &\sim \mathrm{Categorical}(\boldsymbol{\pi})\\ \mathbf{f}_{g_j}(\tilde{\mathbf{x}})|g_j, \{\mathbf{m}_{g_j}, \boldsymbol{\Sigma}_{g_j}\} &\sim \mathrm{GP}_{g_j}(\mathbf{m}_{g_j}, \boldsymbol{\Sigma}_{g_j}), \quad \forall \tilde{\mathbf{x}} \in \mathbb{R}^6 \times \mathbb{R}^d\end{aligned}$$

with $i = 1, \dots, \infty$ and $j = 1, \dots, N$. The associated Bayesian network is illustrated in Fig. 2. In this model, an Inverse-Wishart distribution of parameters $\mathbf{W}_0$, ν_0 is given to the covariance $\boldsymbol{\Sigma}_i$. Both the parameters σ_{g_j} and $\sigma_{w_{g_j}}$ are given inverse gamma priors with hyper-hypers a and b (separately for the two variances). We also give independent log normal prior to the length-scales $\boldsymbol{\Lambda}_{g_j}$ [18].

IV. Model Parameters Learning

The algorithm for learning the parameters of the Dirichlet Process-Gaussian Process (DP-GP) mixture model of the robot motion control is similar to the approach proposed in [18], [21]. Let us consider a set $\mathcal{D} = \{\mathbf{u}_j^t, (\mathbf{q}_j^t, \boldsymbol{\varphi}_j^t)\}_{t=1,j=1}^{T_j,N}$ of observed trajectories, terrain features and control commands (see Fig. 3). These trajectories have been generated by manually steering the robot within different simulated disaster scenarios (e.g., climbing stairs, overcoming obstacles, traversing harsh terrains).

We resort to our knowledge about the robot as well as about its locomotion capabilities to fix in advance an initial value of the number M of patterns of control manoeuvres. Thus, we initialize the mixture model with M components. At each Gibbs sweep, estimates of the assignments of patterns of control manoeuvres to sequences of control commands, as well as the GP parameters are obtained by sampling from the following distributions:

$$\begin{aligned}&\Pr(g_j = i|\mathbf{u}_j^{1:T_j}, \alpha, \{\mathbf{m}_i, \boldsymbol{\Sigma}_i\}) \propto \\ &\propto \begin{cases}\frac{n_i}{N-1+\alpha}\mathcal{L}(g_j = i; \mathbf{u}_j^{1:T_j}) & \text{if } i \leqslant M\\ \frac{\alpha}{N-1+\alpha}\int \mathcal{L}(g_j = i; \mathbf{u}_j^{1:T_j})\mathcal{B}(\mathbf{m}_i, \boldsymbol{\Sigma}_i)\mathrm{d}\mathbf{m}_i\mathrm{d}\boldsymbol{\Sigma}_i & \text{if } i = M+1\end{cases}\end{aligned}$$

and

$$\Pr(\mathbf{m}_i, \boldsymbol{\Sigma}_i|\mathbf{u}_j^{1:T_j}, g_j, \mathbf{m}_{-i}, \boldsymbol{\Sigma}_{-i}) \propto \textstyle\prod_{j|g_j=i} \quad \mathcal{L}(g_j = i; \mathbf{u}_j^{1:T_j})\,\mathcal{B}(\mathbf{m}_i, \boldsymbol{\Sigma}_i)$$

Here $\mathcal{L}(g_j = i; \mathbf{u}_j^{1:T_j}) = \Pr(\mathbf{u}_j^{1:T_j}|\mathbf{q}_j^{1:T_j}, \boldsymbol{\varphi}_j^{1:T_j}, \{g_j = i\}, \{\mathbf{m}_i, \boldsymbol{\Sigma}_i\})$ is the likelihood of the i-th pattern under the sequence of control commands $\mathbf{u}_j^{1:T_j}$. $\mathcal{B}(\mathbf{m}_i, \boldsymbol{\Sigma}_i) = \Pr(\mathbf{m}_i, \boldsymbol{\Sigma}_i|\mathbf{u}_j^{1:T_j}, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \mathbf{W}_0, \nu_0)$. $\mathbf{m}_{-i}$ and $\boldsymbol{\Sigma}_{-i}$ refer to the parameters of the GP_i associated to the remaining trajectories. Then, the DP concentration α is sampled using standard Gibbs sampling techniques [18]. Finally, an hybrid Monte Carlo technique is applied to obtain estimates of the hyper-parameters σ_i, σ_{w_i} and $\boldsymbol{\Lambda}_i$ of each GP_i [19]. Fig. 4 shows the number of control manoeuvres along trajectories which have been estimated by the learning algorithm with respect the number of Gibbs sweeps. Here the initial guess M on the number of components has been set equal to 12.

Fig. 5 shows the relative Root Mean Square error (RMS) of the DP-GP model in imitating the motion patterns. This error is computed comparing the estimated target model to the real underlying motion patterns [3].

International Science Index, Computer and Information Engineering Vol:10, No:6, 2016 waset.org/Publication/10004630

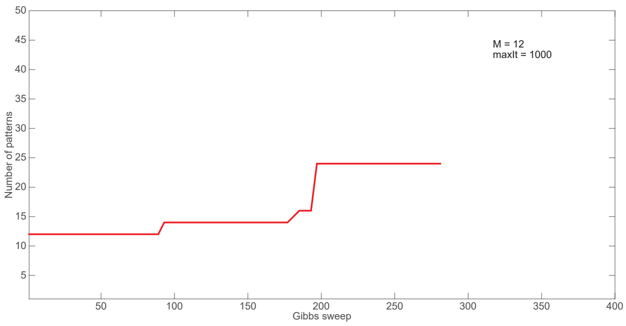

Fig. 4 Number of control manoeuvres along trajectories which have been estimated by the learning algorithm with respect the number of Gibbs sweeps

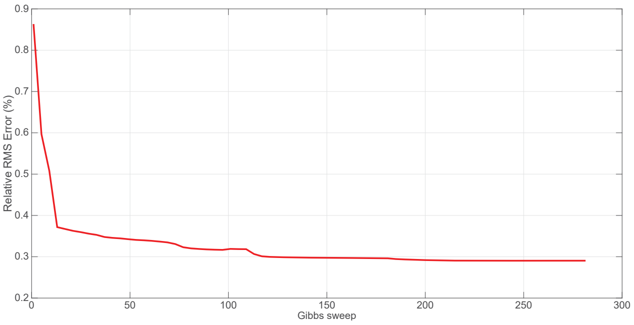

Fig. 5 Relative Root Mean Square error (RMS) of the DP-GP model in imitating the motion patterns over the number of Gibbs sweeps

The learned robot motion control model can be used to predict a pattern of control manoeuvres, given both a trajectory τ_j and the associated terrain features descriptors ϕ_j [2].

V. Future Work

In this work, we described the probabilistic model of the motion control of ATVs based on a Dirichlet Process-Gaussian Process (DP-GP) mixtures. This model is prevailing because it permits the number of patterns of control manoeuvres to be inferred directly from data and thus bypass the difficult model selection problem on the pattern number. However, inference with this model can be computationally inefficient because it requires the inversion of the covariance matrices, though this cost has been greatly alleviated by making use of the mixtures. In order to increase the scalability of the inference as well as to reduce the computational cost of the learning phase, an approach based on Multi-Task learning could be employed [22]. More precisely, the robot motion control task could be separated into two different sub-tasks, namely, the trajectory tracking task and the sub-tracks reconfiguration task. Both these tasks are correlated and share the same representation. Therefore a multi-task learning approach could be suitably applied leading to both a better model for the main task and an improvement of the performance of the learning process.

VI. Acknowledgment

This research is supported by the EU-FP7-ICT-Project TRADR 609763.

References

[1] M. Gianni, F. Ferri, M. Menna, and F. Pirri, “Adaptive robust three-dimensional trajectory tracking for actively articulated tracked vehicles,” *Journal of Field Robotics*, pp. n/a–n/a, 2015. [Online]. Available: http://dx.doi.org/10.1002/rob.21584

[2] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy, “A bayesian nonparametric approach to modeling motion patterns,” *Autonomous Robots*, vol. 31, no. 4, pp. 383–400, 2011. [Online]. Available: http://dx.doi.org/10.1007/s10514-011-9248-x

[3] W. Wei, H.and Lu, P. Zhu, S. Ferrari, R. H. Klein, S. Omidshafiei, and J. P. How, “Camera control for learning nonlinear target dynamics via bayesian nonparametric dirichlet-process gaussian-process (dp-gp) models,” in *IROS*, 2014, pp. 95–102.

[4] Bluebotics, “Absolem surveillance & rescue,” http://www.bluebotics.com/mobile-robotics/absolem/, 2011.

[5] D. Nguyen-Tuong and J. Peters, “Model learning for robot control: a survey,” *Cognitive Processing*, vol. 12, no. 4, pp. 319–340, 2011.

[6] M. P. Deisenroth, G. Neumann, and J. Peters, “A survey on policy search for robotics,” *Foundations and Trends in Robotics*, vol. 2, no. 12, pp. 1–142, 2011.

[7] J. S. Cruz, D. Kulić, and W. Owen, *Proceedings of the Second International Conference on Autonomous and Intelligent Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ch. Online Incremental Learning of Inverse Dynamics Incorporating Prior Knowledge, pp. 167–176.

[8] D. Nguyen-Tuong, B. Scholkopf, and J. Peters, “Sparse online model learning for robot control with support vector regression,” in *IROS*, 2009, pp. 3121–3126.

[9] J. de la Cruz, W. Owen, and D. Kulic, “Online learning of inverse dynamics via gaussian process regression,” in *IROS*, 2012, pp. 3583–3590.

[10] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, “Dynamical movement primitives: Learning attractor models for motor behaviors,” *Neural Comput.*, vol. 25, no. 2, pp. 328–373, 2013.

[11] S. Vijayakumar and S. Schaal, “Locally weighted projection regression: An o (n) algorithm for incremental real time learning in high dimensional space,” in *ICML*, 2000.

[12] J. Kober, A. Wilhelm, E. Oztop, and J. Peters, “Reinforcement learning to adjust parametrized motor primitives to new situations,” *Autonomous Robots*, vol. 33, no. 4, pp. 361–379, 2012.

[13] A. Gijsberts and G. Metta, “Real-time model learning using incremental sparse spectrum gaussian process regression,” *Neural Networks*, vol. 41, pp. 59–69, 2013.

[14] J. Snoek, H. Larochelle, and R. P. Adams, “Practical bayesian optimization of machine learning algorithms,” in *NIPS*, 2012.

[15] O. Kroemer, R. Detry, J. Piater, and J. Peters, “Combining active learning and reactive control for robot grasping,” *Robotics and Autonomous Systems*, vol. 58, no. 9, pp. 1105–1116, 2010.

[16] D. J. Lizotte, T. Wang, M. H. Bowling, and D. Schuurmans, “Automatic gait optimization with gaussian process regression,” in *IJCAI*, 2007.

[17] R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, and M. P. Deisenroth, *Proceedings of the 8th International Conference on Learning and Intelligent Optimization*. Cham: Springer International Publishing, 2014, ch. Bayesian Gait Optimization for Bipedal Locomotion, pp. 274–290.

[18] C. E. Rasmussen and Z. Ghahramani, “Infinite mixtures of gaussian process experts,” in *NIPS*. MIT Press, 2002, pp. 881–888.

[19] S. Duane, A. D. Kennedy, B. Pendleton, and D. Roweth, “Hybrid monte carlo,” *Physics Letters B*, vol. 195, pp. 216–222, 1987.

[20] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, “Gaussian processes for data-efficient learning in robotics and control,” *PAMI*, vol. 37, no. 2, pp. 408–423, 2015.

[21] S. Sun, “Infinite mixtures of multivariate gaussian processes,” in *ICMLC*, 2013, pp. 1011–1016.

[22] M. P. Deisenroth, P. Englert, J. Peters, and D. Fox, “Multi-task policy search for robotics,” in *ICRA*, 2014.