Learning How to Increase the Chance of Human-Robot Engagement

Douglas G. Macharet[1] and Dinei A. Florencio[2]

Abstract— **The increasing use of mobile robots in social contexts makes it important to provide them with the ability to behave in the most socially acceptable way possible. In this paper we investigate the problem of making a robot learn how to approach a person in order to increase the chance of a successful engagement. We propose the use of Gaussian Process Regression (GPR), combined with ideas from reinforcement learning to make sure the space is properly and continuously explored. In the proposed example scenario, this is used by the robot to predict the best decisions in relation to its position in the environment and approach distance, each one accordingly to a certain time of the day. Numerical simulations show a significant performance improvement when compared with a random technique. The robot is able to improve performance after just one day of interaction (a few dozens of trials), and achieves the maximum expected value for the proposed approach within sixty days.**

I. INTRODUCTION

It is widely expected that the use of mobile robots in different parts of society will be commonplace in the near future. This change from controlled environments (e.g., factories) to unconstrained environments where people are constantly present (e.g., home, public places, hospitals, etc.) will require robots to behave in "socially acceptable" ways.

This need for socially acceptable behavior crosses many domains (e.g., can I make noise now? how fast can I move and people still feel safe? can I cross in front of someone? behind?). While behavior in a social space co-occupied by humans brings a number of issues, the direct interaction with people is particularly challenging, as the state of mind of the person is hard to estimate — and may change with the interaction itself. Thus, even a simple decision as whether to initiate interaction is challenging. The simplest approach is to be conservative and never initiate interaction. Depending on the role played by the robot this may be acceptable. For example, for a receptionist robot in a building, it may be acceptable to wait for humans to start the interaction. However, in many cases (e.g. a seller robot in a store) the robot is the one responsible for approaching and initiating the interaction.

Engagement is the process by which different parts have a perceived connection to each other during an interaction [1]. It is clear that many factors affect the likelihood of a person engaging with the robot. Some of these factors may be outside the robot's control (e.g., time of day, size/appearance of the robot, etc). Others may be clearly within the robot's control. For example, the robot can choose how close to get to the person before trying to engage; how loud to talk; whether to approach from the front or from the side; the best location to wait for people; whether to say *Hi*, or *Hello*, etc. Furthermore, these are so variable that we should not expect an operator to be able to accurately estimate the influence of any of them during the robot's setup. Therefore, our main question is whether it is possible (and how) to make the robot learn how to choose each of the variables under its control, in order to influence people and increase the chance of successfully initiate/maintain an interaction.

With that purpose, in this paper we describe a learning framework that will permit a robot to change its behavior accordingly to people's behavior in the current environment. The framework is based on applying principles from *reinforcement learning* on top of a *Gaussian Regression Process*. We will show that — if the variables under the robot control do influence people — learning will increase the chance of successfully initiate engagement in a human-robot interaction.

The problem we tackle is very useful, but it is also very generic. We could make it more specific. For example, we could answer what is the optimal distance for our robot to approach a person at a shopping mall. However, it is not clear that result would still hold if we change any of the parameters of the experiment. Changing the robot size, facial expression, the country, the type of mall, or the task, would likely affect the optimum distance. Thus, instead of making the problem more specific, we focus on the learning process itself. If we show *how* the robot can learn, the same learning process is much more likely to apply, even after any of the conditions of the experiment change.

The scenario we chose to simulate is that of a robot trying to distribute a flyer for a new store opening at a shopping mall (illustrated in Figure 1). The probability of success is influenced by a number of variables, in this work we consider three: time of day, approaching distance, and initial position. All other non-modeled variables are lumped together in the noise component. We also assume the robot will know the outcome of the trial immediately after each approach (success iff the customer took the flyer). We then measure the ability of the robot to learn the influence of variables under its control (robot's position, distance to person before engaging) by themselves and as a function of the variables that are observable, but outside its control (time of day, rate of arrival).

The robot's objective is to tune the variables under its control to maximize the chance of engagement. Results will, of course, vary widely with the scenario. Over our specific

This work was done while Douglas G. Macharet was with Microsoft Research.

[1]Douglas G. Macharet is with the Computer Vision and Robotic Laboratory (VeRLab), Computer Science Department, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil `doug@dcc.ufmg.br`

[2]Dinei A. Florencio is with Microsoft Research, One Microsoft Way, Redmond, WA, USA `dinei@microsoft.com`

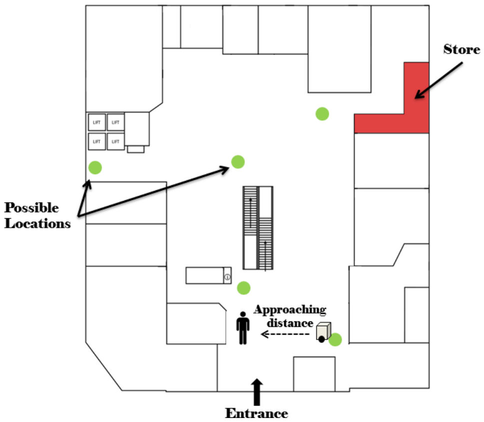


Fig. 1. Illustration of the simulated scenario. The robot can choose to stay in different pre-defined positions (green circles) along the day. After a person is detected the robot must execute an approach accordingly to a certain distance.

simulation parameters, the robot's improved its performance by 25% at the end of the first day, and was $3.5\times$ more effective after 20 days (compared to a random policy).

The rest of this paper is organized as follows: in Section II we summarize related work. Our methodology is presented in detail in Section III and validated by numerical simulations whose results are shown in Section IV. Finally, Section V concludes with a discussion of the results and future research directions.

II. RELATED WORK

In [2] a computational model is proposed for recognizing engagement between a human and a robot based on gestures and speech. In [3] a learning approach is proposed in order to predict engagement intentions.

The interaction however has a initial phase which occurs before the moment that the common parts establish the belief that they are interacting with each other. Therefore, the engagement can be seen as the consequence of a successful initial approach. In [4] a reaction model is proposed which takes into account the behavior of people moving toward the robot in order to encourage interaction. In [5] the methodology considers both participation state of a person and spatial formation, determining appropriate timing and position for initiating conversation. In [6] a behavior model is proposed focusing on approaching people who are walking, which permits the robot to plan an approaching path nonverbally indicating its intention to initiate a conversation. In [7] people's trajectories are analyzed in order to identify potential customers that can be approached by the robot, while [8] uses previous paths to learn a spatial affordance map, which is then used to plan paths that maximize the probability of meeting people.

The fact of people being approached by an autonomously moving robot makes the approach itself a challenging task. There are some implicit nonverbal social rules that must the respected in this case, especially those concerning people's personal spaces (called *proxemics*). As previously works have shown [9], [10], [11], [12], [13], the same proxemic zones that exists in human-human interaction can also be applied to human-robot interaction scenarios. Considering this, some works incorporate this personal space model in the path planning step [14], [15], [16]. However, this model is not static and can vary accordingly to different aspects, such as previously experience with the robot[17], or functional noise of the robot [18]. Therefore, it is important to adapt this model, increasing the chance of a successful approach.

A robot is a very dynamic system which demands for techniques that are capable of adapting along time, making learning methods a good choice in this scenario. However, it is also a high dimensional space, and data is very sparse. This creates learning needs which are very specific. As Morimoto et al. put it, "It would be difficult to naively apply the existing machine-learning methods to these robots"[19]. Thus, most of the work in this context has as the main focus the low-level kinematic control of the robot [20], [21], [22], [23], [24]. When dealing with low-level kinematics, data is not as sparse, and can often be further augmented by physical simulation (e.g., "envisioning"). Learning has also been used to train "human-like" trajectories in crowded environments [25], while [26] investigates the influence of the trajectories on pedestrians' comfort. Some of our own previous work [27], [28] addresses the interaction between people and the robot trajectories, focused on a telepresence scenario. Considering learning in the human-robot interaction scenario [29] presents a reinforcement learning approach which uses discomfort signals from the human as the reward function, making it possible to learn how to approach in socially acceptable distances. Two recent surveys are also relevant, with [30] (and the associated special issue [19]) focusing on several aspects of robot learning, and [31] focusing on the application of reinforcement learning in robotics.

III. METHODOLOGY

A. Problem Formalization

As stated in the previous sections, the machine learning aspect of our problem consists on making a robot learn how to approach a person in order to increase the chance of successful engagement.

The robot is able to choose a certain position p on the environment from a discrete set of allowed positions P. It can use the time of the day t to make this decision. When a person is within the range of perception of the robot, it has to decide a distance d at which it will approach (i.e., start talking to) the person. The result of each approach trial is either successful or unsuccessful. We assume no cost on approaching, thus the robot will approach every person possible. Therefore, we can define the problem at hand more formally as:

Problem 1: The result r_i of each approach trial $\mathcal{T}_i \in \boldsymbol{\mathcal{T}}$ is a binary random variable (success of failure), with the probability of success being a function of the parameter vector $\mathbf{x}$, which includes robot's position $p_i \in \mathbf{P}$, the approach distance $d_i \in \mathbf{D}$ and the time of the day $t_i \in \mathbf{T}$. Consider the subset of successful trials $\boldsymbol{\mathcal{T}}_s = \{\mathcal{T}_i \in \boldsymbol{\mathcal{T}} \mid r_i = success\}$

Then, we want to optimize the success rate of approaches over the variables under the robot's control, i.e., :

$$\underset{\mathbf{P},\ \mathbf{D}}{\text{maximize}} \quad \frac{|\mathcal{T}_{\text{s}}|}{|\mathcal{T}|},$$

where $|\cdot|$ represent the cardinality of a set.

B. Learning Framework

It is natural to consider a reinforcement learning approach to tackle the proposed problem. However, most standard reinforcement learning techniques rely on discrete states [32]. The common approach of discretizing the variables quickly brings in an explosion of the learning space, making the number samples required for training impractical. The approach of fitting a parametric function requires a prior model, and, (on our early experiments) a huge amount of training data as well.

Our problem can, however, be recast as a prediction problem (given the outcome of all previously visited sates, what is the expected outcome of other states). We thus decided to address it as a regression problem. One of the drawbacks of this approach is the right choice of a model. Therefore, our methodology is based on the Gaussian Process Regression (GPR) technique, which consists of a less 'parametric' tool, since it assumes that data can be represented as a sample from a multivariate Gaussian distribution [33]. We then combine GPR with a exploration policy based on principles from reinforcement learning.

Given a vector of input variables $\mathbf{x}$, a Gaussian process is completely specified by its mean function ($m(\mathbf{x})$) and covariance function ($k(\mathbf{x},\mathbf{x}')$)[33]. Thus, we approximate our random process $f(\mathbf{x})$, as a GP:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x},\mathbf{x}')). \tag{1}$$

Additionally, we model the covariance function as a linear combination of the covariance function of each one of the state parameters, time of day (K_{t}) and approach distance (K_{d}), i.e.:

$$K = \alpha_1 K_{\text{t}} + \alpha_2 K_{\text{d}}. \tag{2}$$

Note that this does not imply the variables are linearly related, but simply that the covariance of one does not change as a function of the other.

The position variable will be treated separately, therefore this variable will not be incorporated in the covariance matrix.

GPR involves two steps. Initially, we have to fit a covariance matrix related to the process, that best explain the observed datapoints. This can be done by maximizing the likelihood of the observed data as a function of the parameters in the covariance function. More specifically, each of the covariance matrices K_{t} and K_{d} corresponding to the specific vectors x_i in the experiments

$$K(\mathbf{x},\mathbf{x}) = \begin{bmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & \cdots & k(\mathbf{x}_1,\mathbf{x}_{\text{n}}) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_{\text{n}},\mathbf{x}_1) & \cdots & k(\mathbf{x}_{\text{n}},\mathbf{x}_{\text{n}}) \end{bmatrix} \tag{3}$$

is obtained by modeling $k(\mathbf{x},\mathbf{x})$ as

$$k(\mathbf{x}_{\text{i}},\mathbf{x}_{\text{j}}) = \sigma_f^2 \exp\left(\frac{-(\mathbf{x}_{\text{i}}-\mathbf{x}_{\text{j}})^2}{2l^2}\right) + \sigma_n^2 \delta_{\text{ij}}(\mathbf{x}_{\text{i}},\mathbf{x}_{\text{j}}), \tag{4}$$

where $\delta_{\text{ij}}(\mathbf{x}_{\text{i}},\mathbf{x}_{\text{j}})$ is a Kronecker delta which is one iff i = j and zero otherwise. The squared exponential function has some characteristic parameters such as the maximum allowable covariance (σ_f^2), a length parameter related to the separation of the observations (l) and a parameter related to the process noise (σ_n^2).

The next step consists in predicting the mean and variance of the process accordingly to the observations vector $\mathbf{y}$ as:

$$\bar{\mathbf{y}}_* = K_* K^{-1}\mathbf{y} \tag{5}$$
$$\text{var}(\mathbf{y}_*) = K_{**} - K_* K^{-1} K_*^{\text{T}} \tag{6}$$

where $\mathbf{y}_*$ is the Gaussian process prediction (random variable) for the test input vector $\mathbf{x}_*$, $\bar{\mathbf{y}}_*$ is its mean and var($\mathbf{y}_*$) its variance. The covariance matrices are $K = K(\mathbf{x},\mathbf{x})$, $K_* = K(\mathbf{x}_*,\mathbf{x})$ and $K_{**} = K(\mathbf{x}_*,\mathbf{x}_*)$.

Since the method is executed iteratively, it is necessary to use an efficient policy to choose the next state to visit in order to improve the model learned until the moment. Here we borrow from the reinforcement learning theory, and establish a balance between exploration and exploitation.

Considering the mean (Equation 5) and variance (Equation 6) previously presented, we first introduce in Equation 9 the policy regarding the position. Among all the possible positions, we choose the value with the maximum predicted mean with a probability P^p_{mean}, the value with the highest value on a 95% C.I. with a probability P^p_{ci} or a random position otherwise.

$$p_{\text{mean}} = \underset{\forall i \in |\mathbf{P}|}{\arg\max} \max(\bar{\mathbf{y}}_*)^{[i]} \tag{7}$$
$$p_{\text{std}} = \underset{\forall i \in |\mathbf{P}|}{\arg\max} \max(\bar{\mathbf{y}}_* + 1.96\sqrt{\text{var}(\mathbf{y}_*)})^{[i]} \tag{8}$$
$$p \sim \begin{cases} p_{\text{mean}} & \text{, if } r \leq P^p_{\text{mean}} \\ p_{\text{std}} & \text{, if } P^p_{\text{mean}} < r \leq P^p_{\text{ci}} \\ \mathcal{U}(1,|\mathbf{P}|) & \text{, otherwise} \end{cases} \tag{9}$$

where $r \in [0,1]$ is a random real number with uniform distribution.

We present in Equation 12 the policy regarding the distance. Assuming a position was chosen, we select some test values ($\mathbf{x}_*$) and predict the possible values. Similarly to the previous step, with a probability P^d_{mean} we select the value with the maximum mean, but then we choose a distance with a Normal distribution considering this value. The value with the highest value on a 95% C.I. with a probability P^d_{ci} is selected, and distance is chosen again with a Normal distribution. Otherwise, a value is uniformly randomly chosen in the domain of the set of distances.

$$d_{\text{mean}} = \underset{\forall x \in \mathbf{x}_*}{\arg\max} \max(\bar{\mathbf{y}}_*) \tag{10}$$
$$d_{\text{std}} = \underset{\forall x \in \mathbf{x}_*}{\arg\max} \max(\bar{\mathbf{y}}_* + 1.96\sqrt{\text{var}(\mathbf{y}_*)}) \tag{11}$$

$$d \sim \begin{cases} \mathcal{N}(d_{\text{mean}}, \sigma_{\text{d}}^2) & , \text{ if } r \leq P_{\text{mean}}^d \\ \mathcal{N}(d_{\text{std}}, \sigma_{\text{d}}^2), & , \text{ if } P_{\text{mean}}^d < r \leq P_{\text{ci}}^d \\ \mathcal{U}(\min(\mathbf{D}), \max(\mathbf{D})) & , \text{ otherwise} \end{cases} \tag{12}$$

where $r \in [0, 1]$ is a random real number with uniform distribution.

Figure 2 is presented is other to clarify the main idea behind the policies. The first case on the policies is responsible for choosing the best know value. The second case is used to increase the confidence in an area not yet explored. Finally, the third case is responsible for executing a global search. It is important to notice that choosing the distance based on a Normal distribution it is also executing a local search.

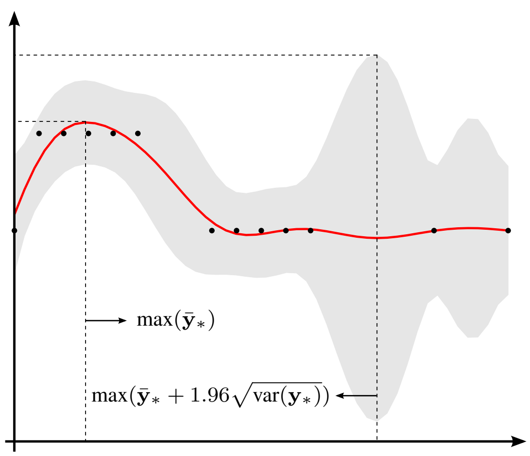


Fig. 2. Example of the exploration policy. The red line represents the Gaussian process posterior mean ($\bar{\mathbf{y}}_*$) and the gray area represents a 95% C.I. around the mean.

The Gaussian process prediction method has a basic complexity of $O(n^3)$ due to the inversion of K (considering standard techniques), which can be prohibitive for large datasets. However, for the specific problem under consideration, datasets are typically small, varying from a few dozens to a few thousands. At this levels, computational complexity is not a problem. For situations where the robot approaches say, over 10,000 people, data consolidation or other methods for controlling the complexity may need to be used.

C. Simulation

The proposed framework was evaluated in a simulation context in order to measure the learning potential, and evaluate the amount of data needed to learn certain characteristics of the interaction.

Our arrival method is based on a non-homogeneous Poisson process, since we consider that it can vary along the day. The frequency of inclusion of new people in the environment is defined by a rate (intensity) function $\lambda(t)$, which represents an estimate of the amount of new persons that must be created in a given unit of time.

The random length of time that will pass before the next region is inserted into the environment is obtained by generating random numbers based on a sample in the inverse transform in accordance with a cumulative distribution function [34]. The cumulative distribution function of a homogeneous Poisson process rate λ can be represented by an exponential distribution:

$$F(x) = 1 - e^{-\lambda x}, \qquad x \geq 0, \tag{13}$$

with the inverse transform given by

$$T = \frac{-\ln U}{\lambda}, \tag{14}$$

where $U \in [0, 1]$ is a random real number with uniform distribution.

The event schedule of the non-homogeneous Poisson Process for each time slot t_i are then generated according to a homogeneous Poisson process with rate $\lambda(t_i)$. Each event has a probability p_i of being added to the schedule given by

$$p_i = \frac{\lambda(t_i)}{\max(\lambda(t))}. \tag{15}$$

Considering an environment with $\mathbf{P} = \{1, 2, 3\}$, we propose for each position a model that represents the probability of successful engagement. The models are based on the time of the day ($\mathbf{T} = [0\text{h}, 12\text{h}]$) and approach distance ($\mathbf{D} = [0\text{m}, 5\text{m}]$). Figure 3 presents the models, the maximum probability (red areas) in all positions is 90%.

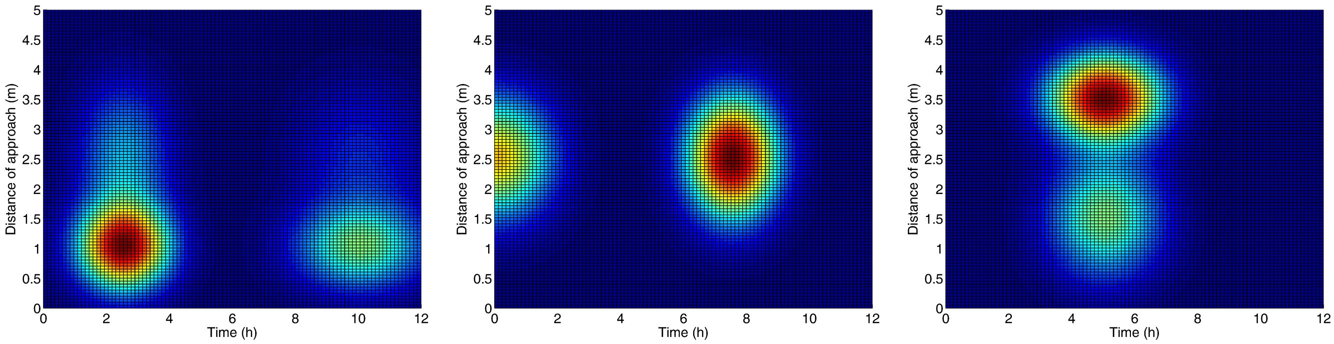
Algorithm 1 presents a simplified overview of the execution of the framework. In line 2 an schedule of the arrival times are created using the Poisson process previously described. On each iteration of the loop the robot chooses a new position (line 4) based on the current time (t) using Equation 9. Line 5 verifies if it is time for a new arrival. When a new arrival happens the robot must initially choose an approach distance (line 6) using Equation 12. The approach tentative is then realized performed considering the position, distance and current time (line 7). The result is obtained selecting an uniform random number and comparing to the corresponded probability of the model. Based on the return the model is updated (line 8) based on Equations 5 and 6.

Algorithm 1 SimulationLoop()

```
1:  t ← 0
2:  s ← createArrivalSchedule();
3:  while stopping criteria not met do
4:      p ← choosePosition(t);
5:      if isTimeOfNextArrival(s, t) then
6:          d ← chooseDistanceToApproach(p, t);
7:          r ← tryToApproach(p, d, t);
8:          updateModel(r);
9:      end if
10:     t ← t + Δt
11: end while
```

IV. NUMERICAL EXPERIMENTS

In this section we describe our experiments and the corresponding statistical analysis, showing the improvement obtained in the success engagement rate with the proposed



(a) Position 1 (b) Position 2 (c) Position 3

Fig. 3. Simulated approachability model proposed. The success of a tentative engagement is represented as a probability based on the position, time of day and approach distance. The maximum probability (red areas) in all positions is 90%.

methodology. The simulator was implemented using Matlab and all experiments were executed in a PC with an Intel Xeon 3.60 GHz processor, 8 Gb of RAM, and a 64-bit Windows OS. Table I presents an overview of the specific parameters related to the simulation execution.

TABLE I
PARAMETERS USED IN THE SIMULATION.

	Parameter	Value
GPR	σ_f^2	0.5
	l	1
	σ_n^2	1
	$\alpha_1 = \alpha_2$	1
Policies	$P^p_{\text{mean}} = P^d_{\text{mean}}$	0.8
	$P^p_{\text{ci}} = P^d_{\text{ci}}$	0.9
	σ_d^2	0.25
Simulation	Number of days	60
	Minimum arrival rate	$\frac{1}{30}$
	Maximal arrival rate	$\frac{1}{10}$

Figure 4 presents the behavior of the engagement success rate along the days. The red line are the results given by a random policy (the position and distance are randomly chosen) which has a mean value of $\approx 13\%$. The blue line represents the results of an optimal policy, with a mean success rate of $\approx 61\%$. The black line are the results given by our proposed methodology. After a period of circa 20 days it is possible to observe a convergence of the results to a mean value of $\approx 46\%$, a significant improvement over the random policy. Furthermore, as we will explain later, mostly all the gap to the optimum policy is explained by our decision to keep the exploration component active.

Figure 5 presents the model learned by the end of the simulated period. Despite the considerable size of the search space, the methodology was able to satisfactorily estimate the underlying model (especially the peaks). It is important to notice that some parts of the space may never be fully explored since we give priority to the use of known good estimates, instead of looking in parts of the space already proven not to lead to better solutions.

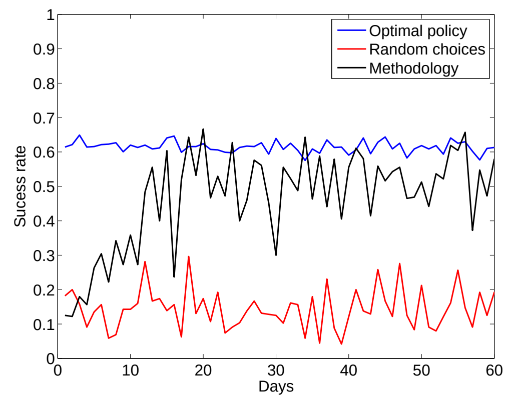


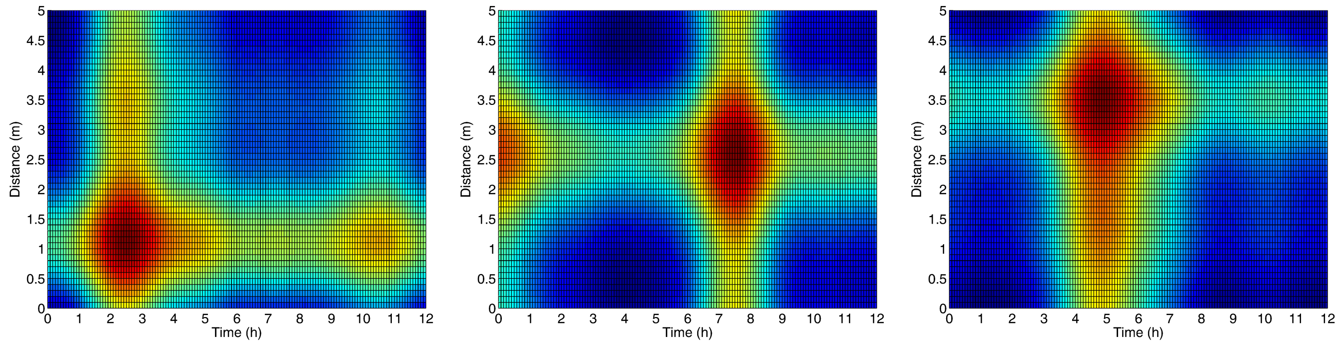
Fig. 4. Engagements success rate considering three different policies. The optimal policy accordingly to the proposed model (blue), a random policy (red) and the proposed methodology (black).

Figures 6 and 7 show the possible decisions that could be made by the policy considering the model learned by the end of the simulation. Figure 6 presents the decision related to the position of the robot accordingly to the time of the day. The blue lines are the optimal values and the black lines the values given by the methodology. Most of the time the robot is at the optimal position but still has some freedom to explore other positions.

Figure 7 presents the decision related to the approach distance accordingly to the time of the day. The blue lines are the optimal values and the black dots the values given by the methodology. The values are spread around the optimal values, which was expected considering Equation 12.

As the proposed technique is a probabilistic method, in order to perform a thorough statistical analysis, we present next an overall analysis with a significant number of experiments. We run 300 experiments considering the same parameters used in the previous experiment.

Figure 8 presents the average success rate along the simulated period. As can be observed, the methodology achieves a success rate of 50% in less than 30 days, about $4\times$ more effective than the random policy. If we fit an exponential to the learning curve, we conclude that the methodology



(a) Position 1 (b) Position 2 (c) Position 3

Fig. 5. Approachability model learned after a simulated period of 60 days.

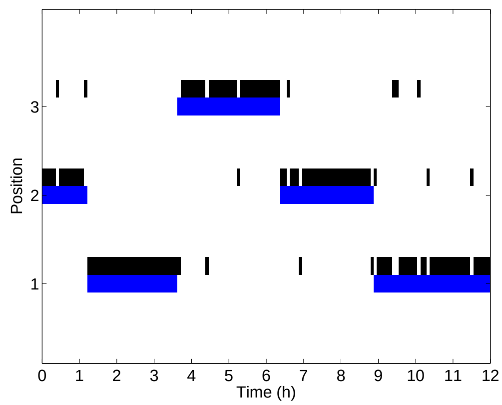


Fig. 6. Example of possible decisions taken during a day by the policy responsible for choosing a new position. This example considers the model previously learned after a 60 days simulation.

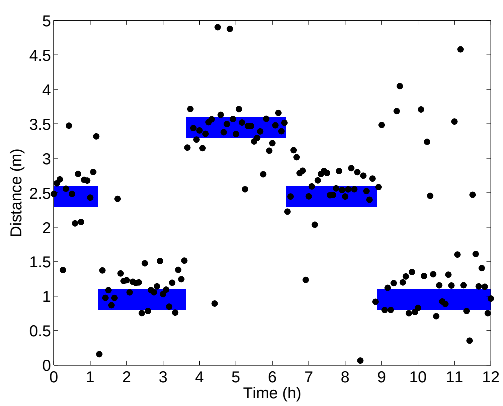


Fig. 7. Example of possible decisions taken during a day by the policy responsible for choosing the approaching distance. This example considers the model previously learned after a 60 days simulation.

has a "learning time constant" of 7 days. It is important to observe that the mean value obtained by the methodology will be lower than the maximum expected mean using the optimal policy due to the random values that are chosen accordingly to each policy (position and distance). Indeed, in 10% of the time it chooses a random position, and 10% of the time uses a random distance. Thus we estimate that this would reduce the success rate to approximately 52% (i.e., $(.81)61\% + (.19)13\%)$). The additional 2% sub-optimality is most likely also derived from the random noise added to the distance. In any case, both could easily be removed if we allow the policy to decrease the rate of exploration as it learns.

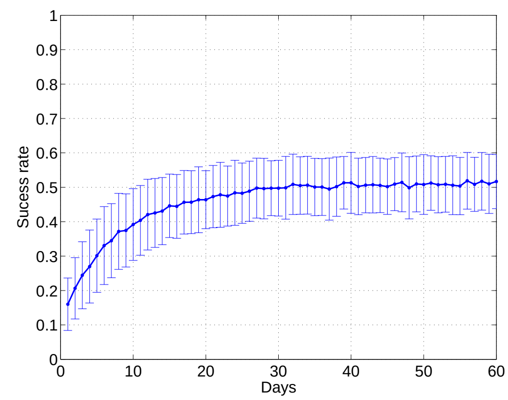


Fig. 8. Average success rate of the proposed methodology considering a simulated period of 60 days and 300 experiments.

Note also on Figure 8 the early improvement. We can estimate the success rate at the end of first day at around 18%, an improvement of almost 40% over the random policy after just a day.

Besides number of days, it is also important to evaluate the number of approaches necessary for convergence. Although we consider different arrival rates during the day, we assume they remain unchanged among days. Considering the values used for the minimum and maximum rate we obtained an average of $\approx$ 80 new approaches each day (Figure 9). Therefore, it has a "learning time constant" of $\approx$ 560 approaches.

V. CONCLUSIONS AND FUTURE WORK

This work proposed a learning framework for mobile robots in order to increase the chance of successfully initiate engagement in a human-robot interaction.

The proposed methodology is based on Gaussian Process Regression, combined with principles from Reinforcement learning. The use of GPR allows a rather simple model, while still being powerful enough to adequately represent

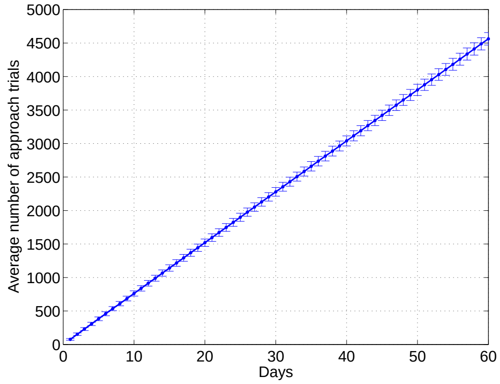


Fig. 9. Average number of approach trials considering a minimum and maximum rate of $\frac{1}{30}$ and $\frac{1}{10}$, respectively. Simulated period of 60 days and 300 experiments.

the the underlying phenomenon. In the simulated experiment, improvements of around 40% (over a random policy) were obtained after just one day ($\approx$ 80 engagement trials), and nearly optimum results after 60 days. Although suitable for problems with a reduced number of training data (our case), the main drawback of the technique is its computational cost, which can be a problem for long-term executions.

Future research directions include the deployment of a robot in a real world scenario, a better study of the parameters used in the model (probably the use of an optimization process on every learning iteration of the algorithm), and the analysis of different learning policies, particularly regarding the trade-off between exploration and exploitation. The study of techniques with a lower (or even constant) computational complexity is also of interest, particularly for other scenarios, where the number of data points may become excessively large.

ACKNOWLEDGMENTS

The authors would like to thank Dan Bohus, Harsha Kikkeri and Ashish Kapoor for insightful discussions during earlier phases of the project.

REFERENCES

[1] C. L. Sidner, C. Lee, C. D. Kidd, N. Lesh, and C. Rich, "Explorations in engagement for humans and robots," *Artificial Intelligence*, vol. 166, no. 1-2, pp. 140–164, Aug. 2005.

[2] C. Rich, B. Ponsler, A. Holroyd, and C. Sidner, "Recognizing engagement in human-robot interaction," in *HRI'10*, 2010.

[3] D. Bohus and E. Horvitz, "Learning to predict engagement with a spoken dialog system in open-world settings," in *SIGDIAL'09*, 2009.

[4] N. Bergstrom, T. Kanda, T. Miyashita, H. Ishiguro, and N. Hagita, "Modeling of natural human-robot encounters," in *IROS'08*, 2008.

[5] C. Shi, M. Shimada, T. Kanda, H. Ishiguro, and N. Hagita, "Spatial formation model for initiating conversation," in *RSS'11*, 2011.

[6] S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, and N. Hagita, "How to approach humans?: strategies for social robots to initiate interaction," in *HRI'09*, 2009.

[7] T. Kanda, D. Glas, M. Shiomi, and N. Hagita, "Abstracting People's Trajectories for Social Robots to Proactively Approach Customers," *IEEE Trans. on Robotics*, vol. 25, no. 6, pp. 1382–1396, 2009.

[8] D. Tipaldi and K. Arras, "I want my coffee hot! Learning to find people under spatio-temporal constraints," in *ICRA'11*, 2011.

[9] M. Walters, K. Dautenhahn, R. Boekhorst, K. Koay, D. Syrdal, and C. Nehaniv, "An empirical framework for human-robot proxemics," in *Proc. of. New Frontiers in Human-Robot Interaction Symposium at the AISB09 Convention*, 2009.

[10] J. Mumm and B. Mutlu, "Human-robot proxemics: physical and psychological distancing in human-robot interaction," in *HRI'11*, 2011.

[11] M. Walters, M. Oskoei, D. Syrdal, and K. Dautenhahn, "A long-term Human-Robot Proxemic study," in *RO-MAN'11*, 2011.

[12] R. Mead and M. J. Mataric, "A probabilistic framework for autonomous proxemic control in situated and mobile human-robot interaction," in *HRI '12*, 2012.

[13] G. Ferrer, A. Garrell, M. Villamizar, I. Huerta, and A. Sanfeliu, "Robot interactive learning through human assistance," in *Multimodal Interaction in Image and Video Applications*. Springer Berlin Heidelberg, 2013, vol. 48, pp. 185–203.

[14] M. Svenstrup, S. Tranberg, H. Andersen, and T. Bak, "Pose estimation and adaptive robot behaviour for human-robot interaction," in *ICRA'09*, 2009.

[15] J. Kessler, C. Schroeter, and H.-M. Gross, "Approaching a person in a socially acceptable manner using a fast marching planner," in *ICIRA'11*, 2011.

[16] E. Avrunin and R. Simmons, "Using human approach paths to improve social navigation," in *HRI'13*, 2013.

[17] L. Takayama and C. Pantofaru, "Influences on proxemic behaviors in human-robot interaction," in *IROS'09*, 2009.

[18] N. van Berkel, "How adjustments to the velocity and functional noise of a robot can enhance the approach experience," in *Proc. of TSConIT*, 2013.

[19] J. Morimoto, O. C. Jenkins, and M. Toussaint, "Learning control in robotics," *IEEE Robotics Automation Mag.*, vol. 17, no. 2, pp. 17–18, 2010.

[20] T. Hester, M. Quinlan, and P. Stone, "RTMBA: A Real-Time Model-Based Reinforcement Learning Architecture for robot control," in *ICRA'12*, 2012.

[21] P. Q. Vidal, R. I. Rodriguez, M. A. R. Gonzalez, C. V. Regueiro, and F. V. Villarrubia, "Learning in real robots from environment interaction," *Journal of Physical Agents*, vol. 6, no. 1, 2012.

[22] T. Hester, M. Quinlan, and P. Stone, "Generalized model learning for reinforcement learning on a humanoid robot," in *ICRA'10*, 2010.

[23] C. Strauss and F. Sahin, "Autonomous navigation based on a q-learning algorithm for a robot in a real environment," in *SoSE'08*, 2008.

[24] D. Forte, A. Ude, and A. Kos, "Robot learning by gaussian process regression," in *RAAD'10*, 2010.

[25] P. Henry, C. Vollmer, B. Ferris, and D. Fox, "Learning to navigate through crowded environments," in *ICRA'10*, 2010.

[26] H. Kidokoro, T. Kanda, D. Brscic, and M. Shiomi, "Will I bother here? - a robot anticipating its influence on pedestrian walking comfort," in *HRI'13*, 2013.

[27] D. Macharet and D. Florencio, "A collaborative control system for telepresence robots," in *IROS'12*, 2012.

[28] A. Cosgun, D. Florencio, and C. Henrik, "Autonomous person following for telepresence robots," in *ICRA'13*, 2013.

[29] N. Mitsunaga, C. Smith, T. Kanda, H. Ishiguro, and N. Hagita, "Robot behavior adaptation for human-robot interaction based on policy gradient reinforcement learning," in *IROS'05*, 2005.

[30] S. Schaal and C. Atkeson, "Learning control in robotics," *IEEE Robotics Automation Mag.*, vol. 17, no. 2, pp. 20–29, June.

[31] J. Kober and J. Peters, "Reinforcement learning in robotics: A survey," in *Reinforcement Learning*, ser. Adaptation, Learning, and Optimization, M. Wiering and M. Otterlo, Eds. Springer Berlin Heidelberg, 2012, vol. 12, pp. 579–610.

[32] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: The MIT Press, 1998, A Bradford Book.

[33] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, 2006.

[34] D. E. Knuth, *Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 3rd ed. Addison-Wesley Professional, 1997.