Robot Navigation to approach people using G^2-Spline path planning and Extended Social Force Model

Marta Galvan*, Ely Repiso*, and Alberto Sanfeliu

Institut de Robòtica i Informàtica Industrial, CSIC-UPC
{mgalvan,erepiso,sanfeliu}@iri.upc.edu.

Abstract. When a robot has to interact with a person in a dynamic environment, it has to navigate to reach a close distance and to be in front of the person. This navigation has to be smooth and take care of the person's movements, the static obstacles and the motion of other people. In this paper, we present a new method to approach a person, that combines G^2-Splines (G^2S) paths with the Extended Social Force Model (ESFM) to allow the robot to move in dynamic environments avoiding static obstacles and other people. Moreover, we use the Bayesian human motion intentionally prediction (BMP) in combination with the Social Force Model (SFM) to be able to approach a moving person and also to avoid moving people in the environment. The method computes several paths using the G^2S and taking into account the person's position and orientation. Then, the method selects the best path using several costs that consider distance, orientation, and interaction forces with static obstacles and moving people. Finally, the robot is controlled with the ESFM to follow the best path. The method was validated by a set of simulations and also by real-life experiments with a humanoid robot in a dynamic environment.

Keywords: Human-Robot approaching, Robot Navigation, Human-Robot Interaction, Human-Robot Collaboration.

1 Introduction

Our society is evolving to include intelligent robots in daily live, which have to interact and collaborate with humans. These robots have to develop several skill and behaviors, among them social or collaborative navigation [8, 14], learning how to approach people and develop original ways of reaching them [4, 18], or understanding and predicting human intentions [22, 12, 6].

Work supported by the Spanish Ministry of Science and Innovation under EU project AI4EU (H2020-ICT-2018-2-825619), the national RD project ColRobTransp (DPI2016-78957-RAEI/FEDER EU), by the Spanish State Research Agency through the María de Maeztu Seal of Excellence to IRI (MDM-2016-0656). Ely Repiso is also supported by Spanish Ministry of Science and Innovation under a FPI-grant, BES-2014-067713. [* Means that both authors contributed equally to this work]

Fig. 1: The robot uses the implemented method to approach a static and moving person, while avoiding several static obstacles of the environment.

Social navigation and approach a person are common tasks for people. We tend to approach other people when: we need help to arrive at a specific place, we want to buy something in a shop, we meet someone somewhere or we know someone in the street, and so on.

For humans moving around people and obstacles while approaching one person socially and predictably is very natural, but for robots is a challenging behavior, that implies develop different skills to be able to do different things at the same time. In particular, robots have to navigate autonomously towards humans, predict people movement, recognize the person to be approached and know its position and orientation, deal with uncertainties like momentary occlusions of the approached person by other people, approach in a human-like way and initiate interactions following social rules and patterns. Furthermore, the difficulty increases if the person to be approached is moving. Fig. 1 shows two real experiments, firstly the robot approaches a static person (the two fist images at left) and secondly approaches a dynamic person (the two last images at right), while it has to avoid several static obstacles of the environment.

Some researches like Carton et al. [3, 4] try to use appropriate human-like features, such as smooth trajectory shapes, specified approach speed, appropriate human-robot distance, etc. With these characteristics the approaching of the robot appears significantly more natural, enhancing non-verbal interaction initiation with humans, mutual collision avoidance and reduction of interference. Also, Takayama et al. [20] do a user study to know the influence of several factors in the preferences of the personal spaces in the approaching behavior for Japanese people. Furthermore, Joosee et al. [13], study the appropriateness of the robot's approach behavior in different cultures, where they found that Chinese participants prefer closer approaches compared to participants from the U.S. or Argentina. In our work, we also try to use most of these human-like features.

Other methods to approach people are based on learning algorithms like in [1], where they use learning methods to find the personal comfort field, without invading the personal space. That work is based on an online learning algorithm where the robot learns the user's specific personal comfort space from the user's reactions, by exploring regions nearby the user to search for a more comfortable approaching trajectory. Another method is based on the collection of real data to obtain human-like approaching behaviors like in [2], where they collect navigation trajectories in which a person approaches another human, to create a model that can be used by a robot's path planner to get more socially acceptable paths.

In our institute, we started to develop methods to approach a person in [16], where the robot tries to approach a person while accompanying another person. The model includes a framework to calculate a moving goal taken into account the movement of the approached person, the movement of the group and the best path to go throw the obstacles of the environment. After that in [17] we enhanced the previous approach by computing the best encounter point using a gradient descent method, taking into account all people's predictions. In the encounter point, the robot performs a triangle formation to achieve an engagement with both people.

Now in this paper, we go a step further from the previous approaching methods realizing a path planning algorithm that allows us to have a more human-like robot navigation to approach a person. First, we obtain some smoothed paths using G^2-Splines computation; then, we formulate a new cost function, to select the best path to go to the goal while avoiding collisions with the entities in the environment, which uses the social forces presented in the ESFM [8]. Furthermore, we use the ESFM to control the robot to deal with real-time navigation in dynamic environments. The computation is done online and in real-time, to keep the environment constantly updated and to have always a feasible path for the robot.

In the remainder of the paper, we start by introducing the implemented approach, that combines the G^2-Splines (G^2S) with the Extended Social Force Model (ESFM) in Sec. 2. Then, we show the developed metrics of performance to evaluate the task in Sec 3. In Sec. 4 we include the simulation results. The real-life experiments with our robot are shown in Sec. 5. Finally, Sec. 6 presents the conclusions.

2 Extended Social Force G^2-Spline navigation method to approach a person.

In this section, we present a robot navigation method to approach a person that combines the G^2-Splines [15] (G^2S) path planning with the Extended Social Force Model (ESFM) [8]. The path planning algorithm for the robot behavior is summarized in Alg.1. In Fig. 1 you can see two different instants of time of two approaching situations and in Fig. 2, you can see a simulation example of the path planning generation. In this article, we have used a similar notation of [8].

The basic idea is to start approaching a person when the robot is close to he/she (around 15 m), using a combination of G^2S path planning and ESFM. The method assumes that the robot has to move from its location up to a surrounding circle around the person that has 1:5 m of radius, and we suggest a 5 good final goal locations over this circle to approach that person, covering and angle view of 80° around the point to face directly the person (their computation is better explained, next in the algorithm). We use only 5 final destinations around the person to allow the robot to do not approach directly facing the person if it is necessary, but neither allow the robot to approach the person very laterally using a difference of orientations between them smaller than 45°. You can see how these 5 final locations are distributed around the person in the upper images of Fig. 2.

Algorithm 1 AKP approaching(s_{ini}; Δt)

Initialize $T(V, E) \leftarrow \{\emptyset\}$
$V \leftarrow s_{ini}$
$\{D^{goal}_{p_j;n}\} = \texttt{scene\_prediction}$
$\{D^{goal}_{r;n}\} = \texttt{final\_destinations}(s_{ini}, N)$
for $n = 1$ to N **do**
  $\{x_1, ..., x_k\} = \texttt{splines\_generation}(s_{ini}, D^{goal}_{r;n})$ $\triangleright$ $x_1 = s_{ini}$ and $x_k = D^{goal}_{r;n}$
  for $i = 1$ to K **do**
    $u_{r_i} = \texttt{calculate\_edge}(x_i, x_{i+1}, i, dist_{col})$
  end for
  $J_n = \texttt{path\_cost\_computation}(\{x_1, ..., x_k\}, \{u_{r_1}, ..., u_{r_k}\})$
end for
$[D^{goal}_{r;best}, J_{best}] = \texttt{best\_path\_selection}(\{J_1, \ldots J_N\})$
$u_{r_f} = \texttt{calculate\_edge}(s_{ini}, step_{goal}, index, dist_{col})$
$s_{new} = \texttt{robot\_propagation}(s_{ini}, u_{r_f})$
$s_{new} = \texttt{orientation\_adjusting}(s_{new})$
$V \leftarrow \{s_{new}, J_{best}\}$
$E \leftarrow \{u_{r_f}\}$
return $\texttt{branch}(T)$

The distance between the person and the robot was based on a previous work of our institute [9]. At each cycle time, the robot computes G^2-Splines paths to go from its current position to each one of the 5 possible final destinations. For each one of the paths, the algorithm uses the ESFM to evaluate the paths (and selects the best one) and to control the robot approaching behavior, while avoiding static and moving obstacles (moving people). As the environment is dynamic (has other people around), the robot has to generate new paths in each iteration (every 0.2 seconds) in order to adapt to it.

In the following, we will present the entire procedure to obtain the approaching behavior of the robot. At each iteration a robot plan is computed, and the linear and angular velocities of the robot state, included in s_{new}, are executed by the robot controller to move it. The input of the algorithm is the $s_{ini} \in S$, where $S = S_r \cup S_{p_i}$ is the state that contains the information of the robot state plus all people's states considered on the scene, included the approached person (see [8]). This state includes position, velocity and orientation for the robot and people. Moreover, the cycle time is set to $\Delta t = 0.2$ sec (It is the maximum time allowed by the controller to work in real time). Furthermore, all the laser scans are processed to obtain the static and dynamic obstacles of the environment.

In the function $\texttt{scene\_prediction}()$, the robot infers all the final destinations for all the people of the environment, $D^{goal}_{p_j;n}$, by using the people's direction of movement, and using these destinations and SFM [11] to predict the people paths inside a window of time of 5 seconds, using the BHMIP [6].

Then, the algorithm computes $N = 5$ candidate paths using the G^2-Splines function [15], in $\texttt{splines\_generation}()$. Moreover, the needed information to generate a path is only related to the initial and final states. For details we refer to [15]. Then, these paths start in the robot position and finish around the position of the approached person. To find the final destinations of all the paths we use the function in Alg. 1-*line 4*. This function computes 5 final destinations at 1.5 meters around the position of the approached person, taking into account the person orientation and computing the possible final states starting from the one facing directly the person in front of it and using an increment of 20° or 40° from the starting goal, where the robot face directly the person in front of it, to both sides (left and right) to compute the other four approaching positions. We use this type of geometric splines, because they present very nice properties: the quintic G^2-Splines offers flexibility and, since they are geometric polynomials of 5th order with second order geometric continuity (G^2), the curvature is continuous. Furthermore, to obtain the robot iterative and dynamic behavior we need to split these paths of the G^2-Splines in K steps with length of 0.2 m. This length is the maximum distance that can cover the robot during one iteration time of the algorithm, due to the maximum robot's velocity that is 1m/s and the maximum iteration time of the robot's controller that is 0.2 seconds (this not mean that the robot always cover this path distance, it depends of the resultant force of this step of the path, this distance is only the maximum one to allow the robot to navigate at its maximum speed if does not have any obstacles around). Then, we obtain the robot steps $\{x_1, \ldots, x_k\}$ inside the path.

After that for each path we need to know the forces in each step of the path, because the robot navigates using the ESFM. These forces follow the Helbing definition [11] and are computed in function `calculate_edge`(), where $dist_{col}$ defines the radius of the circular area were the interactions with obstacles and other people are considered. The ESFM to control the robot uses as local goals the steps of the path computed by the G^2-Splines and includes repulsive forces between the robot and static and dynamic obstacles, like people. According to the model, both humans and robots are free particles in a 2D space, following the laws of Newtonian mechanics, and the *resulting force* $\boldsymbol{F}_e$ that governs the trajectory of the movement of each entity ($e = \{r, p\}$, where the entity e can be robot (r) or person (p)) is described in Eq. 1. This force is used to control the robot in a dynamical environment where people or other robots are moving around, and also to predict the people movement.

$$\boldsymbol{F}_e = \boldsymbol{f}^{goal}_{e,d}(D^{goal}_{e,n}) + \sum_{j\in P} \boldsymbol{f}^{int}_{e,j} + \sum_{b\in R} \boldsymbol{f}^{int}_{e,b} + \sum_{o\in O} \boldsymbol{f}^{int}_{e,o}, \qquad (1)$$

where, P, O and R are the sets of people, obstacles and robots of the environment, respectively. The resultant force, $\boldsymbol{F}_e$, is composed by the attractive force until the destination and the repulsive forces respect other people, robots or obstacles. Furthermore, the $\{\ ,\ ,\ \}$ parameters were used in [7], to implement three different robot behaviors (aware, balanced and unaware) and here we select different robot behaviors respect to the final goal, and the people and obstacles interactions, that face better our approaching case. We set these parameters equal to $\{\ = 1.2,\ = 1.4,\ = 1.0\}$, to allow a balanced robot behavior respect obstacles, an aware behavior respect people and an intermediate behavior between unaware and balanced respect to the final goal. The aware robot's behavior respect people allows the robot to hinder less the people's path.

The first force of Eq. 1 is the *attraction force* to reach the goal Eq. 2. This force assumes that the entity tries to adapt its velocity within a relaxation time k^{-1} to arrive to the destination and is given by:

$$f^{goal}_{e,d}(D^{goal}_{e,n}) = k(\mathbf{v}^0_e(D^{goal}_{e,n}) - \mathbf{v}_e) \tag{2}$$

where $k = 1/\tau$, and τ is the time for a human to take a step, 0.5 seconds approximately. $\mathbf{v}_e$ is the current velocity of the entity, $D^{goal}_{e,n}$ is any of the possible final destinations for the robot or people and $\mathbf{v}^0_e(D^{goal}_{e,n})$ is the desired velocity to reach the goal. Moreover, each repulsive interaction forces are modeled using the Helbing [11] social force model, as:

$$f^{int}_{e,z} = A_{ez} e^{(d_{ez} - d_{e,z})/B_{ez}} \hat{d}_{e,z} w(\varphi_{e,z}, \lambda_{ez}) \tag{3}$$

where $z \in P \cup O \cup R$ is either a person or a static obstacle or a robot, of the environment. A_{ez} and B_{ez} denote respectively the strength and range of the repulsive interaction force, between e and z. $d_{e,z}$ is the distance between the centers of the two entities, and $d_{ez} \geq r_e + r_z$, is a parameter that depends only from the interaction between each type of entities. Moreover, given the limited field of view, influences might not be isotropic, requiring then a scaling anisotropic factor, $w(\varphi_{e,z}, \lambda_{ez})$, further details in [5]. Finally, all the parameters $\{k, A_{ez}, B_{ez}, \lambda_{ez}, d_{ez}\}$ are defined depending on the interaction type, where we use the parameters learned in [21] for the interaction force between robot-obstacle, and we use the parameters learned in [5] for the interaction force between human-robot.

Now, we need to calculate the final cost of each path, Alg. 1-*line 10*, to be able to finally select the best path, J_{best}, until the best final destination, $D^{goal}_{r,best}$, using the cost, Alg. 1-*line 12*. The best path has the minimum cost and the cost for each path requires three steps calculation. First, each individual cost function of $\boldsymbol{J}(S, U) = [J_d(S), J_{or}(S), J_p(U), J_o(U)]$ is computed. There are multiple objectives to be minimized in dynamic planning, and we use different and independent criteria, where each single cost is related to the considered criteria: distance of all the steps of the path (path length), change of the robot orientation during all the steps of the path (path curvature) and repulsive forces respect to people and obstacles during all the steps of the path according to the ESFM (interactions between people and obstacles). Each different cost computed for all the steps along the path is defined by:

$$J_d(S) = \sum_{t_{ini}}^{t_{end}} ||\mathbf{x}_r(t+1) - \mathbf{x}_r(t)||^2, \qquad J_{or}(S) = \sum_{t_{ini}}^{t_{end}} ||\theta_r(t+1) - \theta_r(t)||^2,$$

$$J_p(U) = \sum_{t_{ini}}^{t_{end}} \sum_{i=1}^{P} ||u_{p_i}(t)||^2, \qquad J_o(U) = \sum_{t_{ini}}^{t_{end}} \sum_{i=1}^{O} ||u_{o_i}(t)||^2$$

where, $\mathbf{x}_r = (x_r, y_r)$ is the robot position, θ_r is the robot orientation $u_{p_i}(t)$ and $u_{o_i}(t)$ are the repulsive forces respect to people and obstacles. Second, in order to avoid the scaling effect of a weighted-sum method, each cost function

Algorithm 2 Robot propagation

function Robot propagation($s_{ini}, u_{r_f}, \Delta t$)
  $s_{ini} = [x_{t_i}, y_{t_i}, \theta_{t_i}, v_{t_i}, \omega_{t_i}, t_i]^T$
  $a_{v_{t_i}} = f_x \cos(\theta_{t_i}) + f_y \sin(\theta_{t_i})$
  $a_{\omega_{t_i}} = -f_x \sin(\theta_{t_i}) + f_y \cos(\theta_{t_i})$
  $v_{t_{i+1}} = v_{t_i} + a_{v_{t_i}} \cdot \Delta t$
  $\omega_{t_{i+1}} = \omega_{t_i} + a_{\omega_{t_i}} \cdot \Delta t$
  $x_{t_{i+1}} = x_{t_i} + v_{t_i} \cos(\theta_{t_i}) \cdot \Delta t + a_{v_{t_i}} \cos(\theta_{t_i}) \cdot \Delta t^2/2$
  $y_{t_{i+1}} = y_{t_i} + v_{t_i} \sin(\theta_{t_i}) \cdot \Delta t + a_{v_{t_i}} \sin(\theta_{t_i}) \cdot \Delta t^2/2$
  $\theta_{t_{i+1}} = \theta_{t_i} + \omega_{t_i} \cdot \Delta t + a_{\omega_{t_i}} \cdot \Delta t^2/2$
  return $s_{new} = [x_{t_n}, y_{t_n}, \theta_{t_n}, v_{t_n}, \omega_{t_n}, t_n]^T$
end function

is normalized to $(-1, 1)$, with Eq. 4-*left*. Third, a projection via weighted sum is obtained with Eq. 4-*right* to convert the multi-cost function in a single cost for each path.

$$\bar{J}_i(X) = erf\left(\frac{X - \ _x}{\ _x}\right) ; \quad J(S,U) = \sum_i w_i \bar{J}_i(S,U) \tag{4}$$

Finally, the algorithm selects the best path that has the minimum cost, which represents minimum distance and minimum orientation changes, as well as avoiding interactions between people and obstacles. Then, we use the function calculate_edge() to obtain the resultant force Eq. 1, $u_{r_f} = (a_x, a_y) = (F_{x_r}, F_{y_r})/m_r$, to move the robot following the best path, where $m_r = 1$. Now, the entity means robot. Next, with function robot_propagation() of Alg. 2, we convert the forces into linear and angular accelerations to propagate the robot position and to obtain the angular and linear velocities that needs our robot controller to move the robot. It is to say that we obtain s_{new} for the robot. Finally, the function orientation_adjusting() is used to adjust the robot orientation until obtain a small difference, less than 20°, between its orientation and the desired orientation of the best path, only if the robot reaches the final goal with a big difference of orientation.

3 Metrics of Performance for Positioning the robot respect to the approached person

In this section, the performance metrics used to evaluate the robot behavior are described. These metrics are based on previous studies on humans [19] and the proxemic rules, proposed by Hall [10]. Furthermore, the limits of the interaction distances used were based on a previous work of our institute [9], and a similar version of these performances is included in [17].

The final position of the robot respect to the approached person was evaluated using three types of performance metrics. One related with proxemics, based on several areas of performance, to obtain if the robot arrives to the best approached area to face the person. Other two metrics that serves to differentiate if the robot arrives inside a desired margin of distances with respect to the approached person and if also the robot is oriented to face the person.

The first performance takes into account the spatial relationship in 2D between the robot and the approached person and it is defined by three areas: *(i)* Human's personal space C, is the area where the robot can not be in order to avoid invading any human's personal space and it takes into account the space delimited by the radius of the person, that includes some free space, correspondent to R_i; *(ii)* Social distance area A, is the area where the robot should be to be socially accepted; and *(iii)* Human's best approaching area B, is the area where the robot must be placed so that the person perceives a socially accepted approach behavior. This area is a dynamic area inside the social distance that depends on the final path destinations and the best path selection from the 5 possible planned paths of the robot, Alg. 1-*line 4 and 12*. The description of these areas was included in [17]. Here only change in the A the P_c correspondent to the companion person to P_a for the approaching person, and also the physical position of the area B changes because here we have only the approached person and this area is defined by the final goal of the best path. Furthermore, the formulation of how the performance was extracted from these areas and the description of the robot area is detailed in [17], next we only explain the general idea to understand it. The Area metric has the maximum performance of 1 when the robot is in the area described by B, since it is the best position to approach the human. Additionally, if the robot is in the area A, but not in area B, is a partial success, since the robot is inside the social distance of the human, but not in the best approaching position. Then the performance has a value of 0.5. Finally, if the robot is further than 3 meters from the human's position, then we consider that there is not approaching interaction between robot and person, and therefore its performance is 0. Also, if the robot invades any human's personal space is penalized with 0 performance.

Regarding the distance and angle performances, we consider that the robot achieves a good distance performance if it keeps its central position inside the interval of distances $[1.25 \quad 2]$ m, respect to the position of the approached person. This margin is around the ideal value of 1.5 m. Then, between 2 meters until 3 meters the performance decreases from 1 until 0, and also between 0.75 until 1.25 the performance increases from 0 to 1. The reader is referred to [17] for further explanation and see the equation of the metric of performance in distance. In terms of angle, the best angle performance is, at most, a difference of 20° from the ideal orientation to face the approached person (performance value of 1). Then, if the difference of the angle increases, we penalized it until we obtain a 0 value of performance, when the robot has an error of 90° with respect to the ideal value of orientation. The equation of the angle performance metric is shown in Eq. 5, where $P_{\ diff}$ means angle performance and is the difference of orientations between the robot and the approached person, $_{diff} = (180° \quad _r) \quad _t$, and $_t$ means the orientation of the approached person and $_r$ is the real orientation of the robot. For a better understanding of the performance metrics, the reader can find an image that graphically shows the performances in this link: `http://www.iri.upc.edu/people/erepiso/ROBOT2019.html`

$$P_{diff} = \begin{cases} 1 & if\ 0^\circ \quad {}_{diff} \quad 20^\circ \\ \frac{1}{70}(\ {}_{diff}) + \frac{9}{7} & if\ 20^\circ < \ {}_{diff} \quad 90^\circ \\ 0 & otherwise \end{cases} \quad (5)$$

4 Simulation experiments

The actual section describes our simulation environment and all the possible situations that we used to test and validate our implemented approach. To be able to test these situations we used a complex simulation environment, used in all our previous works [16, 17] and also a Gazebo simulator. All is coded using C++ and ROS. This environment includes random people moving trough different destinations, represented by green cylinders, and several obstacles, represented by dark grey cylinders, that we can put in any place. The people movement follows the Social Force Model [11] and their desired velocities were randomly selected inside the interval of [0-1] m/s. Furthermore, our simulated robot follows the laws of a non-holonomic vehicle and uses the presented algorithm to approach a person. The robot has a maximum velocity of 1 m/s. Between the robot and the target person we have used a final social distance of 1.5 m as minimum distance to approximate. We have selected this distance to prevent the robot to go too close to the person and we are based in one previous work [9].

To test and validate a large field of situations we perform more than 7200 simulations. These simulations include different approaching situations, where the robot has to approach one person (static or moving towards any possible destination) while avoiding several static and dynamic obstacles. A representative number of the simulation cases is shown in Fig. 2, where the simulation environment contains the robot model, the approached person in red. The possible paths for the robot are drawn in orange and in red the best path. The time window to compute the paths and take into account the people and obstacle interactions, which is a black dashed circle around the robot. The resultant force for the robot is represented with a red arrow over the robot, the blue arrow is the force until the goal, the green arrows are repulsive forces respect to people, the black arrows are repulsive forces respect to static obstacles and the purple arrows are repulsive forces for people with respect to the robot.

The first group of simulations were carried out in an empty environment, where we tested the approaching behavior of the robot with a static person that has 4 different approaching orientations (0°, 90°, 180° and 270°). We used only 4 cases because with these 4 selected orientations we covered a huge range of approaching behaviors of the robot. The second group of simulations were also in the empty space, but now the approached person was moving using three different approaching directions (right, center and left), that also represent well most of the possible real interactions with a moving approaching person. The third ones included the before approaching behaviors of the person, but with other people moving around. In the fourth ones the robot had to avoid

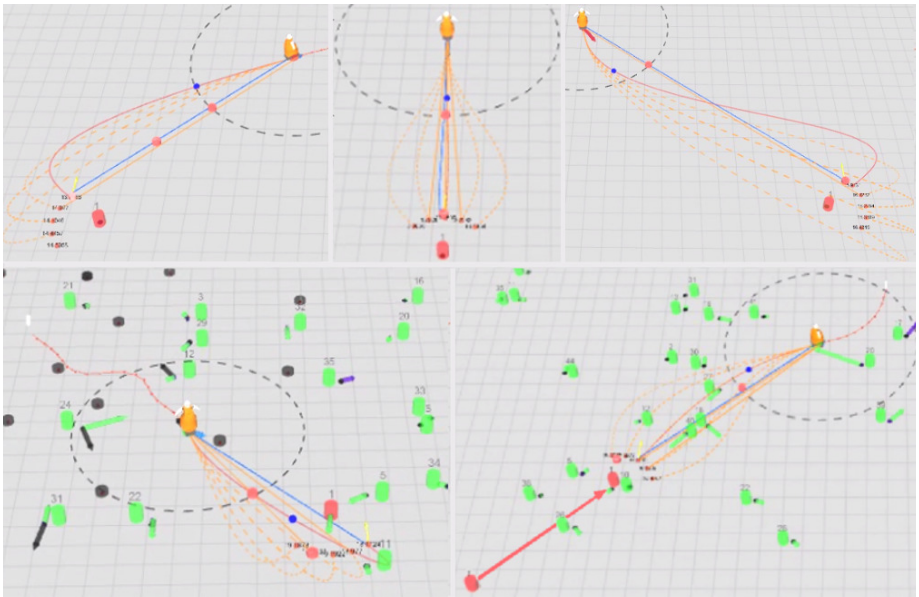

Fig. 2: **Synthetic experiments:** Simulation environment to test and evaluate the new method to approach a person. *Up*: Three images that show different approaching angles respect to a static person. *Down-left*: Shows the robot approaching one person, using the fourth approximation angle, while avoids static and dynamic obstacles. *Down-right*: Shows the robot approaching a moving person while avoids dynamic obstacles.

several static obstacles while approaching to the person with the same behaviors described before (static and dynamic). The fifth group of simulations included people and obstacles at the same time.

The performance of the robot was evaluated in all of these simulations using the metrics described in Sec. 3. Where the performance is inside the interval of $[0 \quad 1]$ and the best value of performance corresponds to 1. Table 1 shows all the performances for all the cases.

5 Real-life experiments

The implemented method was tested also in a real-life environment. The experiments were developed in the FME (Facultat de Matemàtiques i Estadística) lab, an outdoor urban environment located at the South Campus of the Universitat Politècnica de Catalunya (UPC). There, we have a controlled environment where we can test the algorithm without any obstacles and with an approached person situated at different points of the environment and with different orientations (static or moving), with static or dynamic obstacles and the same configurations for the approached person as the case without obstacles. The mean and standard deviation of the approaching performance of the 82 real-life experiments are shown in Table 2 and Fig. 1 shows several interesting moments of these experiments. In addition, the reader can find some videos of the real experiments in this link: `http://www.iri.upc.edu/people/erepiso/ROBOT2019.html`

Approaching Performance simulations	mean(P_{2R_i})	mean(P_{diff})	mean(P(r, p_a))
Person stop, without obst	0.98 (0.04)	1 (0)	0.98 (0.07)
Person stop, with dynamic obst	0.97 (0.07)	0.66 (0.19)	0.99 (0.05)
Person stop, with static obst	0.96 (0.06)	0.85 (0.13)	0.97 (0.08)
Person moving, without obst	0.96 (0.05)	1 (0)	0.99 (0.02)
Person moving, with dynamic obst	0.97 (0.1)	0.73 (0.03)	0.99 (0.05)
Person moving, with static obst	0.84 (0.13)	0.92 (0.16)	0.94 (0.13)
Person stop, static & dynamic obst	0.86 (0.12)	0.55 (0.21)	0.96 (0.095)
Person moving, static & dynamic obst	0.84 (0.13)	0.79 (0.15)	0.96 (0.08)

Table 1: Performance results for the approaching task of all the simulations. The performance value equal to 1 is considered the best value and the values between brackets are the standard errors of each mean value.

Approaching Performance real-life	mean(P_{2R_i})	mean(P_{diff})	mean(P(r, p_a))
Person stop, without obst	0.86 (0.14)	0.77 (0.2)	0.98 (0.02)
Person stop, with dynamic obst	0.8 (0.13)	0.82 (0.23)	0.97 (0.04)
Person stop, with static obst	0.96 (0.07)	0.62 (0.25)	0.97 (0.04)
Person moving, without obst	0.87 (0.14)	0.8 (0.23)	0.93 (0.11)
Person moving, with dynamic obst	0.86 (0.19)	0.67 (0.37)	0.98 (0.03)
Person moving, with static obst	0.93 (0.12)	0.64 (0.3)	0.92 (0.14)

Table 2: Performance results for the approaching task of all the real-life experiments. The performance value equal to 1 is considered the best value and the values between brackets are the standard errors of each mean value.

6 Conclusions

We have presented a method that combines the G^2-Splines with the Extended Social Force Model to allow the robot to approach a person as much natural human-like as possible. The major contribution of this work is how we combine both methods, G^2S and ESFM, to obtain a more smooth and natural approaching behavior for the robot, while the robot navigates well inside dynamic environments and at the same time facilitate the walking behavior of other people. Furthermore, the robot's final path will be free of obstacles by using the ESFM applied to the G^2S path generation. The computation is done in real time. The new method has been tested and validated over simulations and good results have been obtained. Furthermore, we validated the algorithm in real-life experiments on the FME, where the robot achieved a good approaching behavior.

References

1. Ahn, H., Oh, Y., Choi, S., Tomlin, C.J., Oh, S.: Online learning to approach a person with no regret. IEEE Robotics and Automation Letters, vol. 3, no. 1, January 2018
2. Avrunin, E., Simmons, R.: Using human approach paths to improve social navigation. Human-Robot Interaction (HRI), ACM/IEEE International Conference on Human-Robot Interaction, March 2013
3. Carton, D., Olszowy, W., Wollherr, D., Buss, M.: Socio-contextual constraints for human approach with a mobile robot. Springer Science+Business Media Dordrecht, January 2017

4. Carton, D., Turnwald, A., Wollherr, D., Buss, M.: Proactively approaching pedestrians with an autonomous mobile robot in urban environments. In: Experimental Robotics. pp. 199–214. Springer (2013)
5. Ferrer, G., Garrell, A., Sanfeliu, A.: Robot companion: A social-force based approach with human awareness-navigation in crowded environments. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1688–1694. IEEE (2013)
6. Ferrer, G., Sanfeliu, A.: Bayesian human motion intentionality prediction in urban environments. Pattern Recognition Letters 44, 134–140 (2014)
7. Ferrer, G., Sanfeliu, A.: Multi-objective cost-to-go functions on robot navigation in dynamic environments. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3824–3829. IEEE (2015)
8. Ferrer, G., Sanfeliu, A.: Anticipative kinodynamic planning: multi-objective robot navigation in urban and dynamic environments. Autonomous Robots pp. 1–16 (2018)
9. Garrell, A., Sanfeliu, A.: Cooperative social robots to accompany groups of people. The International Journal of Robotics Research 31(13), 1675–1701 (2012)
10. Hall, E.T., Edward, T.: Hall. the hidden dimension. Anchor Books New York 20, 71 (1969)
11. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Physical review E 51(5), 4282 (1995)
12. Ikeda, T., Chigodo, Y., Rea, D., Zanlungo, F., Shiomi, M., Kanda, T.: Modeling and prediction of pedestrian behavior based on the sub-goal concept. Robotics: Science and Systems VIII p. 137 (2013)
13. Joosse, M., Poppe, R., Lohse, M., Evers, V.: Cultural differences in how an engagement-seeking robot should approach a group of people. CABS'14, August 2022, 2014, Kyoto, Japan
14. Khambhaita, H., Alami, R.: Viewing robot navigation in human environment as a cooperative activity. arXiv preprint arXiv:1708.01267 (2017)
15. LoBianco, C.G., Piazzi, A.: Optimal trajectory planning with quintic g2-splines. in Proceedings of the IEEE Intelligent Vehicles Symposium, 2000, pp. 620-625 (Dearborn, Michigan, USA, October 3-5, 2000)
16. Repiso, E., Garrell, A., Sanfeliu, A.: On-line adaptive side-by-side human robot companion to approach a moving person to interact. Iberian Robotics conference. Springer, pp. 113125, 2017
17. Repiso, E., Garrell, A., Sanfeliu, A.: Adaptive side-by-side social robot navigation to approach and interact with people. International Journal of Social Robotics pp. 1–22 (2019)
18. Satake, S., Kanda, T., Glas, D.F., Imai, M., Ishiguro, H., Hagita, N.: How to approach humans?-strategies for social robots to initiate interaction. In: Human-Robot Interaction (HRI), 2009 4th ACM/IEEE International Conference on. pp. 109–116. IEEE (2009)
19. Stubbs, K., Hinds, P.J., Wettergreen, D.: Autonomy and common ground in human-robot interaction: A field study. IEEE Intelligent Systems 22(2) (2007)
20. Takayama, L., Pantofaru, C.: Influences on proxemic behaviors in human-robot interaction. In: Intelligent robots and systems, 2009. IROS 2009. IEEE/RSJ international conference on. pp. 5495–5502. IEEE (2009)
21. Zanlungo, F., Ikeda, T., Kanda, T.: Social force model with explicit collision prediction. EPL (Europhysics Letters) 93(6), 68005 (2011)
22. Zanlungo, F., Ikeda, T., Kanda, T.: Potential for the dynamics of pedestrians in a socially interacting group. Physical Review E 89(1), 012811 (2014)