7

Navigation Among Humans

Mikael Svenstrup
Aalborg University
Denmark

1. Introduction

As robot are starting to emerge in human everyday environments, it becomes necessary to find ways, in which they can interact and engage seamlessly in the human environments. Open-ended human environments, such as pedestrian streets, shopping centres, hospital corridors, airports etc., are places where robots will start to emerge. Hence, being able to plan motion in these dynamic environments is an important skill for future generations of robots. To be accepted in our everyday human environments, the robots must be able to move naturally, and such that it is both safe, natural and comfortable for the humans in the environment.

Imagine a service robot driving around in an airport. The objective of the service robot is: to drive around among the people in the environment; identify people who need assistance; approach them in an appropriate way; interact with the person to help with the needs of the person. This could be showing the way to a gate, providing departure information, checking reservations, giving information about local transportation, etc. This chapter describes algorithms that handle the motion and navigation related problems of this scenario.

To enable robots with capabilities for safe and natural motion in human environments like in the above described scenario, there are a number of abilities that the robot must possess. First the robot must be able to sense and find the position of the people in the environment. Also, it is necessary to obtain information about orientation and velocity of each person. Then the robot must be able to find out if a person needs assistance, i.e. the robot needs to establish the intentions of the people. Knowing the motion state and the intention state of the person, this can be used to generate motion, which is adapted according to the person and the situation. But in a crowded human environment, it is not only necessary for the robot to be able to move safe and naturally around one person, but also able to navigate through the environment from one place to another. So in brief, the robot must be able to:

- estimate the pose and velocity of the people in the environment;
- estimate the intentions of the people;
- navigate and approach people appropriately according to the motion and intention state of the people;
- plan a safe and natural path through a densely populated human environment.

The overall system architecture is briefly presented in Section 3. Finding the position and orientation of people is done using a laser range finder based method, which is described in Section 4. For on-line estimation of human intentions a Case-Based Reasoning (CBR) approach

is used to learn the intentions of people from previous experiences. The navigation around a person is done using an adaptive potential field, that adjusts according to the person's intentions. The CBR method for estimating human intentions and human-aware motion is described in Sections 5 and 6 respectively. In Section 7 the potential field is extended to include many people to enable motion in a crowded environment. Finally an adapted Rapidly-exploring Random Tree (RRT) algorithm is, in Section 8, used to plan a safe and comfortable trajectory. But first, relevant related work is described in the following section.

2. Related work

Detection and tracking of people, that is, estimation of the position and orientation (which combined is denoted pose) has been discussed in much research, for example in Jenkins et al. (2007); Sisbot et al. (2006). Several sensors have been used, including 2D and 3D vision (Dornaika & Raducanu, 2008; Munoz-Salinas et al., 2005), thermal tracking (Cielniak et al., 2005) or range scans (Fod et al., 2002; Kirby et al., 2007; Rodgers et al., 2006; Xavier et al., 2005). Laser scans are typically used for person detection, whereas the combination with cameras also produces pose estimates Feil-Seifer & Mataric (2005); Michalowski et al. (2006). Using face detection requires the person to always face the robot, and that person to be close enough to be able to obtain a sufficiently high resolution image of the face Kleinehagenbrock et al. (2002), limiting the use in environments where people are moving and turning frequently. The possibility of using 2D laser range scanners provides extra long range and lower computational complexity. The extra range enables the robot to detect the motion of people further away and thus have enough time to react to people moving at a higher speed.

Interpreting another person's interest in engaging in interaction is an important component of the human cognitive system and social intelligence, but it is such a complex sensory task that even humans sometimes have difficulties with it. CBR allows recalling and interpreting past experiences, as well as generating new cases to represent knowledge from new experiences. CBR has previously proven successful in solving spatio-temporal problems in robotics in Jurisica & Glasgow (1995); Likhachev & Arkin (2001); Ram et al. (1997), but not to estimate intentions of humans. Other methods, like Hidden Markov Models, have been used to learn to identify the behaviour of humans Kelley et al. (2008). Bayesian inference algorithms and Hidden Markov Models have also successfully been applied to modelling and for predicting spatial user information Govea (2007).

To approach a person in a way, that is perceived as natural and comfortable requires *human-aware navigation*. Human-aware navigation respects the person's social spaces as discussed in Althaus et al. (2004); Dautenhahn et al. (2006); Kirby et al. (2009); Sisbot et al. (2005); Takayama & Pantofaru (2009); Walters, Dautenhahn, te Boekhorst, Koay, Kaouri, Woods, Nehaniv, Lee & Werry (2005). Several authors have investigated the interest of people to interact with robots that exhibit different expressions or follow different spatial behaviour schemes Bruce et al. (2001); Christensen & Pacchierotti (2005); Dautenhahn et al. (2006); Hanajima et al. (2005). In Michalowski et al. (2006) models are reviewed that describe social engagement based on the spatial relationships between a robot and a person, with emphasis on the movement of the person. Although the robot is not perceived as a human when encountering people, the hypothesis is that robot behavioural reactions with respect to motion should resemble human-human scenarios. This is supported by Dautenhahn et al. (2006); Walters, Dautenhahn, Koay, Kaouri, te Boekhorst, Nehaniv, Werry & Lee (2005). Hall has investigated the spatial relationship between humans (proxemics) as outlined in Hall

(1966; 1963), which can be used for Human-Robot encounters. This was also studied by Walters, et al.Walters, Dautenhahn, Koay, Kaouri, te Boekhorst, Nehaniv, Werry & Lee (2005), whose research supports the use of Hall's proxemics in relation to robotics.
One way to view the problem of planning a trajectory through a crowded human environment, is to see humans as dynamic obstacles. The navigation problem can thus be addressed as a trajectory planning problem for dynamic environments. Given the fast dynamic nature of the problem, robotic kinodynamic and nonholonomic constraints must also be considered.
In the recent decade sampling based planning methods have proved successful for trajectory planning LaValle (2006). They do not guarantee an optimal solution, but are often good at finding solutions in complex and high dimensional problems. Specifically for kinodynamic systems Rapidly-exploring Random Trees (RRT's), where a tree with nodes correspond to connected configurations (vertices) of the robot trajectory, has received attention LaValle & Kuffner Jr (2001).
Various approaches improving the basic RRT algorithm have been investigated. In Brooks et al. (2009), a dynamic model of the robot and a cost function is used to expand and prune the nodes of the tree. Methods for incorporating dynamic environments, have also been investigated. Solutions include extending the configuration space with a time dimension ($\mathcal{C} - \mathcal{T}$ space), in which the obstacles are static van den Berg (2007), as well as pruning and rebuilding the tree when changes occur Ferguson et al. (2006); Zucker et al. (2007). To be able to run the algorithm on-line, the anytime concept Ferguson & Stentz (2006) can be used to quickly generate a possible trajectory. The RRT keeps being improved until a new trajectory is required by the robot, which can happen at any time.
These result only focus on avoiding collisions with obstacles. However, there have been no attempts to navigate through a human crowd taking into account the dynamics of the environment and at the same time generate comfortable and natural trajectories around the humans.

3. System architecture

The structure of the proposed system set-up is outlined in Fig. 1. The three main developed components are encaged by the dashed squares. First, the position of the people in the area are measured. The positions are then sent to the pose estimation algorithm, which estimates the person state using a Kalman filter and successively filters the state to obtain a pose estimate. This information about the peoples interest in interaction is sent to a person evaluation algorithm, which provides information to the navigation subsystem. The navigation subsystem first derives a potential field corresponding to the people's pose and interest in interaction. Then a robot motion subsystem finds out, how to move. This can either be by adapting the motion to interact with one specific person, or generating a trajectory through the environment. The loop is closed by the robot, which executes the path in the real world and thus has an effect on how humans move in the environment. The reactions of the people in the environment are then fed back to the person evaluation algorithm, to be able to learn from experience.

4. Person pose estimation

In this work we use a standard method to infer the position of the people in the environment from laser range finder measurements. This method however, only provides the position of

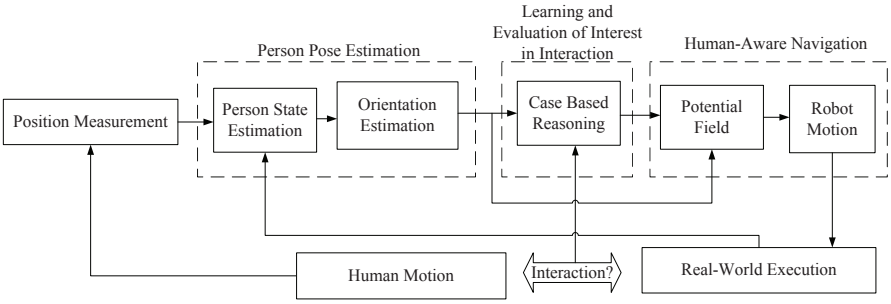


Fig. 1. An overview of how the different components are connected in the developed system. The three main parts encaged by the dashed squares.

a person. Thus, we develop a new Kalman filter based algorithm that takes into account the robot motion, to obtain velocity estimates for the people in the environment. This data is then post-filtered, using an autoregressive filter, which is able to obtain the orientation of the person as well. The method is briefly presented here, but is more thoroughly described in Svenstrup et al. (2009). The method is derived for only one person, but the same method can be utilised when several people are present.

We define the pose of a person as the position of the person given in the robot's coordinate system, and the angle towards the robot, as seen from the person (p_{pers} and θ in Fig. 2). The orientation θ is shown as approximately the angle between ϕ (the angle of the distance vector from the robot to the person) and $\boldsymbol{v}_{pers}$ (the angle of the person's velocity vector). The direction of $\boldsymbol{v}_{pers}$ is, however, only an approximation of θ, since the orientation of a person is not necessarily equal to the direction of the motion.

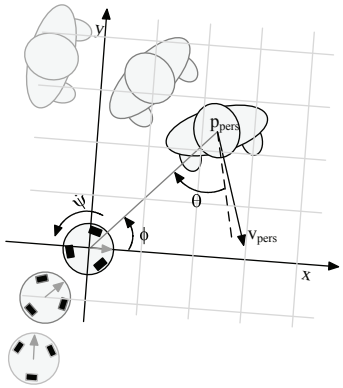


Fig. 2. The state variables $\boldsymbol{p}_{pers}$ and $\boldsymbol{v}_{pers}$ hold the position and velocity of the person in the robot's coordinate frame. θ is the orientation of the person and $\dot{\psi}$ is the rotational velocity of the robot.

The estimation of the person pose can be broken down into three steps:

1. Measure the position of the person using a laser range finder;
2. Use the continuous position measurements to find the state (position and velocity) of the person;

3. Use the velocity estimate to find the orientation (θ) of the person.

The position of the persons is estimated using a standard laser range finder based method, which is described in e.g. Feil-Seifer & Mataric (2005); Xavier et al. (2005). Our implementation is robust for tracking people at speeds of up to $2\frac{m}{s}$ in a real-world public space, which is described in Svenstrup et al. (2008).
The next step is to estimate velocity vector of the person in the robot's coordinate frame. This is done by fusing the person position measurements and the robot odometry in a Kalman filter, where a standard discrete state space model formulation for the system is used:

$$\boldsymbol{x}(k+1) = \Phi\boldsymbol{x}(k) + \Gamma\boldsymbol{u}(k) \tag{1}$$
$$\boldsymbol{y}(k) = H\boldsymbol{x}(k) \quad , \tag{2}$$

where $\boldsymbol{x}$ is the state, $\boldsymbol{y}$ is the measurements, $\boldsymbol{u}$ is the input and Φ, Γ, H are the system matrices, which are explained below.
The state is comprised of the person's position, person's velocity and the robot's velocity. The measurements are the estimate of the person's position and the robot odometry measurements.

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p}_{pers} \\ \boldsymbol{v}_{pers} \\ \boldsymbol{v}_{rob} \end{bmatrix} \quad , \quad \boldsymbol{y} = \begin{bmatrix} \hat{\boldsymbol{p}}_{pers} \\ \boldsymbol{v}_{odom,rob} \end{bmatrix} \quad . \tag{3}$$

$\boldsymbol{p}$ denotes a position vector and $\boldsymbol{v}$ denotes velocities, all given in the robot's coordinate frame. The $\hat{\boldsymbol{p}}$ is the estimate of the person's position from the person detection algorithm and $\boldsymbol{v}_{odom,rob}$ the robot's odometry measurement vector. The rotation of the robot causes the measurement equation to be non-linear. To maintain a linear Kalman filter, the rotational odometry from the robot, can be added to the input, such that:

$$\Gamma(k)\boldsymbol{u}(k) = \begin{bmatrix} p_{y,pers}(k) \\ -p_{x,pers}(k) \\ v_{y,pers}(k) \\ -v_{x,pers}(k) \\ 0 \\ 0 \end{bmatrix} T\hat{\dot{\psi}}(k) \quad , \tag{4}$$

where p_x and p_y are the x and y coordinates of the vector $\boldsymbol{p}$, T is the sampling time and $\hat{\dot{\psi}}$ is the measured rotational velocity of the robot. The derived system matrices are then inserted into a standard Kalman filter to obtain a state estimate of the person.
The direction of the velocity vector in the state estimate of the person is not necessarily equal to the orientation of the person. To obtain a correct orientation estimate the velocity estimate is filtered through a first order autoregressive filter, with adaptive coefficients relative to the velocity. When the person is moving quickly, the direction of the velocity vector has a large weight, but if the person is moving slowly, the old orientation estimate is given a larger weight. The autoregressive filter is implemented as:

$$\theta(k+1) = \beta\theta(k) + (1-\beta)\arctan\left(\frac{v_{y,pers}}{v_{x,pers}}\right) \quad , \tag{5}$$

where β has been chosen experimentally relative to the absolute velocity v as:

$$\beta = \begin{cases} 0.9 & \text{if } v < 0.1m/s \\ 1.04 - 1.4v & \text{if } 0.1m/s \leq v \leq 0.6m/s \\ 0.2 & \text{else} \end{cases} . \quad (6)$$

This equation means, that if the person is moving fast, the estimate relies mostly on the direction of the velocity, and if the person is moving slow, the estimate relies mostly on the previous estimate.

5. Estimate human intentions

To represent the person's interest in interaction, a continuous fuzzy variable, Person Indication (*PI*), is introduced, which serves as an indication of whether or not the person is interested in interaction. *PI* belongs to the interval $[0, 1]$, where $PI = 1$ represents the case where the robot believes that the person wishes to interact, and $PI = 0$ the case where the person is not interested in interaction.

Human intentions are difficult to decode, and thus difficult to hard code into an algorithm. So the robot needs to learn to find out if a person is interested in interaction. For this purpose, a Case-Based Reasoning (CBR) system, which enables the robot to learn to estimate the interest in interaction, has been developed in Hansen et al. (2009). The basic operation of the CBR system can be compared to how humans think. When we observe a situation, we use information from what we have previously experienced, to reason about the current situation. Using the CBR system, this is done in the following way. When a person is encountered, the specific motion patterns of the person is estimated as described above in Section 4. The estimates are compared to what is stored in a *Case Library*, which stores information about what happened in previous human encounters. The outcome, i.e. the associated *PI* values, of the previous similar encounters are used to update the *PI* in the current situation. When the interaction session with the person is over, the *Case Library* is revised with the new knowledge, which has just been obtained. This approach makes the robot able to learn from experience to become better at interpreting the person's interest in interaction.

5.1 Learning and evaluation of interest in interaction

The implementation of the CBR system is basically a database system which holds a number of cases describing each interaction session. There are two distinct stages of the CBR system operation. The first is the evaluation of the *PI*, where the robot estimates the *PI* during a session using the experience stored in the database. The second stage is the learning stage, where the information from a new experience is used to update the database. The two stages can be seen in Fig. 3, which shows a state diagram of the operation of the CBR system. Two different databases are used. The *Case Library* is the main database, which represents all the knowledge the robot has learned so far, and is used to evaluate *PI* during a session. All information obtained during a session is saved in the *Temporary Cases* database. After a session is over, the information from the Temporary Cases is used to update the knowledge in the Case Library.

Specifying a case is a question of determining a distinct and representative set of features connected to the event of a human-robot interaction session. The set of features could be anything identifying the specific situation, such as, the person's velocity and direction, position, relative position and velocity to other people, gestures, time of day, day of the week, location, height of the person, colour of their clothes, their facial expression, their apparent age

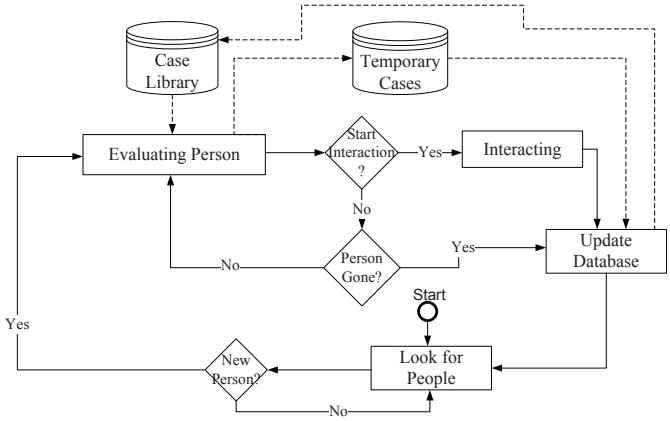


Fig. 3. Operation of the CBR system. First the robot starts to look for people. When a person is found, it evaluates whether the person wants to interact. After a potential interaction, the robot updates the knowledge library with information about the interaction session, and starts to look for other people.

and gender etc. The selected features depend on available sensors. For the system considered in this chapter, we consider only the human motion.
Initially, when the robot locates a new person in the area, nothing is known about this person, so PI is assigned to a default value $PI = 0.5$. After this, the PI of a person is continuously evaluated using the Case Library. First a new case is generated by collecting the relevant set of features above. The case is then compared to existing cases in the Case Library to find matching cases. If a match is found in the Case Library, the PI is updated towards the value found in the library according to the formula

$$PI_{new} = 0.2PI_{library} + 0.8PI_{old} \quad , \tag{7}$$

where PI_{new} is the new updated value of PI. This update is done to continuously adapt PI according to the observations, but still not trusting one single observation completely. If the robot continuously observes similar $PI_{library}$ values, the belief of the value of PI, will converge towards that value, e.g., if the robot is experiencing a behaviour, which earlier has resulted in interaction, lookups in the Case Library will result in $PI_{library}$ values close to 1. This means that the current PI will quickly approach 1 as well. After this, the case is copied to Temporary Cases, which holds information about the current session. If no match is found in the Case Library, the PI value is updated with $PI_{library} = 0.5$, which indicates that the robot is unsure about the intentions of the person. The case is still inserted into the Temporary Cases. When the session is over, the robot will know if it resulted in close interaction with the person, and thus if the person was interested in interaction. All the information in the Temporary Cases can thus be used to update the knowledge of the world, which is stored in the Case Library.

6. Adaptive human-aware navigation

The robot has two different tasks related to moving around in the environment, which it can perform. Either it can move around relative to one specific person that might want to interact, as described in the previous section. Otherwise it can have the objective to move from

one point to another in the environment. This section describes the adaptive human-aware navigation around a specific person. The robot must move safe and naturally around the person, who might be interested in interaction. Furthermore, the motion of the robot must adaptively be adjusted according to the estimated *PI* value. One way to view this problem is to see humans as dynamic obstacles, that have social zones, which must be respected. The zones used here are developed by the anthropologist Hall (1963). Hall divides the area around a person into four zones according to the distance to the person:

- the public zone, where $d > 3.6m$
- the social zone, where $1.2m < d \leq 3.6m$
- the personal zone, where $0.45m < d \leq 1.2m$
- the intimate zone, where $d \leq 0.45m$

If it is most likely that the person does not wish to interact (i.e. $PI \approx 0$), the robot should not violate the person's personal space, but move towards the social or public zone. On the other hand, if it is most likely that the person is willing to interact or is interested in close interaction with the robot (i.e. $PI \approx 1$), the robot should try to enter the personal zone in front of the person. Another navigation issue is that the robot should be visible to the person, since it is uncomfortable for a person if the robot moves around, where it cannot be seen. Such social zones together with the desired interaction and visibility motion, can be represented by potential fields that govern the motion of the robot, see e.g. Sisbot et al. (2006); Svenstrup et al. (2009).

All navigation is done relative to the person, and hence no global positioning is needed in the proposed model. The potential field landscape is derived as a sum of several different potential functions with different purpose. The different functions are:

- **Attractor**. This is a negative potential used to attract the robot towards the person;
- **Rear**. This function ensures that the robot does not approach a person from behind;
- **Parallel**. This is an adaptive function, which is used to control the direction to the person and to keep a certain distance;
- **Perpendicular**. This function is also adaptive and works in cooperation with the parallel potential function.

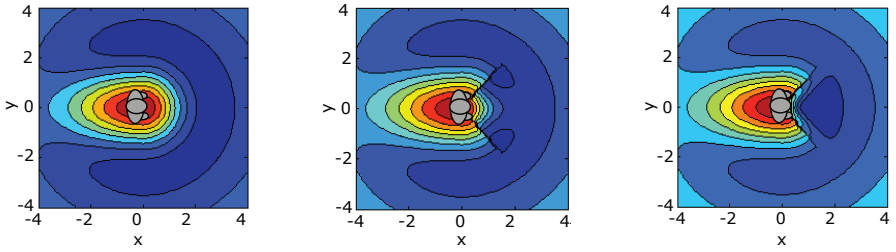
All four potential functions are implemented as normalised bi-variate Gaussian distributions. A Gaussian distribution is chosen for several purposes. It is smooth and easy to differentiate to find the gradient, which becomes smaller and smaller (i.e. has less effect) further away from the person. Furthermore, it does not grow towards ∞ around 0 as, for example, a hyperbola (e.g. $\frac{1}{\boldsymbol{x}}$), which makes both computational feasible and intuitively perceivable. The combined potential field around a person can be calculated as:

$$f(\boldsymbol{x}) = \sum_{k=1}^{4} c_k \exp\left(-\frac{1}{2}[\boldsymbol{x} - \mathbf{0}]^T \Sigma_k^{-1} [\boldsymbol{x} - \mathbf{0}]\right) \quad , \tag{8}$$

where c_k are a normalizing constants, $\boldsymbol{x}$ is the position, $\mathbf{0}$ is the position of the person, in this case the origin, and Σ_k are the covariances of each of the Gaussian distributions.

The attractor and rear distribution are both kept constant for all instances of *PI*. The parallel and perpendicular distributions are continuously adapted, by adjusting Σ_k according to the *PI* value and Hall's proximity distances during an interaction session. Furthermore, the

preferred robot-to-person encounter direction, reported in Dautenhahn et al. (2006); Woods et al. (2006), is taken into account by changing the width and the rotation of the distributions. The adaptation of the potential field distributions enables the robot to continuously adapt its behaviour to the current interest in interaction of the person in question.



(a) *PI*=0 (b) *PI*=0.5 (c) *PI*=1

Fig. 4. Shape of the potential field around a person facing right for (a), a person not interested in interaction, (b) a person maybe interested in interaction, and (c) a person interested in interaction. The robot should try to get towards the lower points, i.e. the dark blue areas.

The resulting potential field contour around a person facing right can be seen in Fig. 4 for three specific values of *PI*. With $PI = 0$ the potential field will look like Fig. 4(a) where the robot will move to the dark blue area, i.e. the lowest potential approximately $3.6m$ in front of the person. The other end of the scale for $PI = 1$ is illustrated in Fig. 4(c), where the person is interested in interaction and, as a result, the potential function is adapted so the robot is allowed to enter the space right in front of the person. In between, Fig. 4(b), is the default configuration of $PI = 0.5$, in which the robot is forced to approach the person at approximately 45°, while keeping just outside the personal zone.
Instead of just moving towards the lowest point at a fixed speed, the gradient of the potential field $\nabla f(\boldsymbol{x})$, is used to set a reference velocity for the robot. This way of controlling the robot's velocity allows the robot to move quickly when the potential field is steep. On the other hand, the robot has slow comfortable movements when it is close to where it is supposed to be, i.e. near a minimum of the field.

7. Trajectory planning problem for crowded environments

The above described potential field governs the motion around one person. The navigation problem for moving through an environment with many people, can then be formulated by summing potential fields for all the people in the environment, plus a potential for the desired robot motion in environment. An example of this potential field landscape, together with three possible trajectories for a robot, is shown in Fig. 5. Notice, that even though a trajectory seems to pass through a person, it might not be a bad trajectory, since the person may have moved when the robot reaches the point of apparent collision. Conversely the robot may also run into a person, who was not originally on the path. Therefore it is important to take into account the dynamics of the obstacles (i.e. the humans), when planning trajectories. So the objective is to minimise the cost of traversing the potential field landscape that changes over time as the people move around.

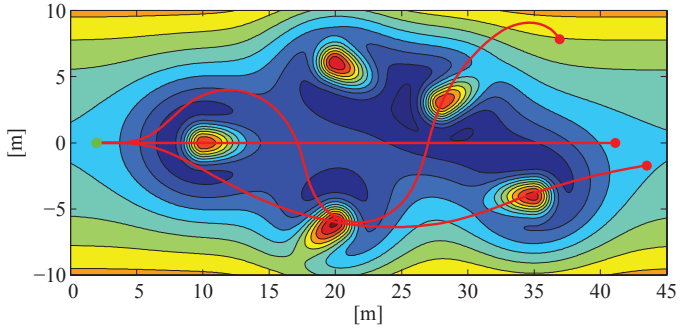


Fig. 5. Person landscape, which the robot has to move through. The robot starting point is the green dot at the point $(2, 0)$. Three examples of potential robot trajectories are shown. Even though it looks like the trajectories goes through the persons, this is not necessarily the case, since the persons might have moved, when the robot comes to the point.

Given the dynamic nature of the problem, robotic kinodynamic and nonholonomic constraints must also be considered. Also the person motion, i.e. the change of the potential field over time must be taken into account.

7.1 Robot dynamics

The robot is modelled as a unicycle type robot, i.e. like a Pioneer, an iRobot Create or a Segway. A good motion model for the robot is necessary because it operates in dynamic environments, where even small deviations from the expected trajectory may result in collisions. So instead of using a purely kinematic robot model of the robot, it is modelled as a dynamical system, with accelerations as input. This describes the physics better, since acceleration and applied force are proportional. This dynamical model can be described by the five states:

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \\ v(t) \\ \theta(t) \\ \dot{\theta}(t) \end{bmatrix} \begin{matrix} \rightarrow \text{x position} \\ \rightarrow \text{y position} \\ \rightarrow \text{linear velocity} \\ \rightarrow \text{rotation angle} \\ \rightarrow \text{rotational velocity} \end{matrix} \tag{9}$$

The differential equation governing the robot behaviour is:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{v}(t) \\ \dot{\theta}(t) \\ \ddot{\theta}(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos(\theta(t)) \\ v(t)\sin(\theta(t)) \\ u_v(t) \\ \dot{\theta}(t) \\ u_\theta(t) \end{bmatrix} = \begin{bmatrix} x_3(t)\cos(x_4(t)) \\ x_3(t)\sin(x_4(t)) \\ u_1(t) \\ x_5(t) \\ u_2(t) \end{bmatrix}, \tag{10}$$

where $u_1 = u_v$ is the linear acceleration input and $u_2 = u_\theta$ is the rotational acceleration input.

7.2 Dynamic potential field

The value of the potential field, denoted G, at a point in the environment is calculated as a sum of the cost associated with three different aspects:

1. A cost related to the robot's desired position in the environment without obstacles. For example high costs might be assigned close to the edges of the environment.

2. A cost associated with the robot position relative to humans in the area.
3. A cost rewarding moving towards a goal.

The combined cost can be written as:

$$G(t) = g_1(\boldsymbol{x}(t))) + g_2(\boldsymbol{x}(t), \mathcal{P}(t)) + g_3(\boldsymbol{x}(t)) \quad . \tag{11}$$

$\mathcal{P}(t)$ is a matrix containing the position and orientation of persons in the environment at the given time. $g_1(\boldsymbol{x})$, $g_2(\boldsymbol{x})$ and $g_3(\boldsymbol{x})$ are the three cost functions. They are further described below.

7.2.1 Cost related to environment

This cost function is currently designed for motion in open spaces, such as a pedestrian street. It has the shape of a valley, such that it is more expensive to go towards the sides, but cheap to stay in the centre of the environment:

$$g_1(\boldsymbol{x}(t)) = c_y y^2(t) \tag{12}$$

where c_y is a constant determining how much the robot is drawn towards the centre, and y is the distance to the center line of the street.

7.2.2 Cost of proximity to humans

The cost of being near to humans is calculated by summing the potential fields for each person as described in Section 6. As described, the potential field for each person is a summation of four normalized bi-variate Gaussian distributions, which means that $g_2(\boldsymbol{x})$ for each individual person can be written as:

$$g_{2,person}(\boldsymbol{x}_{1:2}, \boldsymbol{\mu}) = \sum_{k=1}^{4} c_k \exp(-\frac{1}{2}[\boldsymbol{x}_{1:2} - \boldsymbol{\mu}]^T \Sigma_k^{-1} [\boldsymbol{x}_{1:2} - \boldsymbol{\mu}]) \quad , \tag{13}$$

where c_k are normalizing constants, $\boldsymbol{x}_{1:2}$ are the first two states of the robot state, i.e. the position where the cost function is evaluated. $\boldsymbol{\mu}$ is the position of the person and Σ_k are the covariances of each of the Gaussian distributions, which are related to the orientation and interest state of the person, as described in Section 6.

7.2.3 Cost of end point in trajectory

The cost at the end point penalizes if the robot does not move forwards, and if the robot orientation is not in a forward direction. An exponential function is used to penalize the position. It is set up, such that short distances are penalized much, while it is close to the same value for larger distances, i.e. it does not change much if the robot goes 19 or 20 meters from its starting position.

$$g_3(\boldsymbol{x}(t)) = c_{e1} \exp(c_{e2}(x(t) - \tilde{x}(0))) + c_\theta \theta^4(t) \quad , \tag{14}$$

where $c_{(\cdot)}$ are scaling constants and $\tilde{x}(0)$ is the desired initial position. Note that c_{e2} must be negative to make the potential decrease as the robot gets further. The reason that θ is raised to the fourth, is to keep the term closer to zero in a larger neighbourhood of the origin. This means that the robot will almost not be penalized for small turns. On the other hand larger turns, like going the wrong way, will be penalized more.

7.3 Minimisation problem

Given the above cost functions the potential landscape may be formed as previously illustrated in Fig. 5 for a pedestrian street landscape with five persons. If the current time is $t = 0$, the planning problem can be posed as follows. Given an initial robot state $\boldsymbol{x}_0$, and trajectory information for all persons until the given time $\tilde{\mathcal{P}}_{start:0}$. Determine the control input $\tilde{\boldsymbol{u}}_{0:T}$, which minimises the cost of traversing the potential field, subject to the dynamical robot model constraints and predicted human motions:

$$
\begin{aligned}
\textit{minimise}\quad & I(\tilde{\boldsymbol{u}}_{0:T}) = \\
& \int_0^T \left[g_1(\boldsymbol{x}(t)) + g_2(\boldsymbol{x}(t), \mathcal{P}(t)) \right] dt + g_3(\boldsymbol{x}(T)) \\
\textit{s.t.}\quad & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \tilde{\boldsymbol{u}}_t\right) \\
\textit{where}\quad & g_1(\boldsymbol{x}(t)) = c_y x_2(t)^2 \\
& g_2(\boldsymbol{x}(t), \mathcal{P}(t)) = \\
& \sum_{j=1}^{p} \sum_{k=1}^{4} c_k \exp(-\frac{1}{2} [\boldsymbol{x_{1:2}} - \boldsymbol{\mu}_j]^T \Sigma_{j,k}^{-1} [\boldsymbol{x_{1:2}} - \boldsymbol{\mu}_j]) \\
& g_3(\boldsymbol{x}(T)) = c_{e1} \exp(c_{e2}(x_1(T) - x_1(0))) + c_\theta x_4^4(T) \quad ,
\end{aligned}
\tag{15}
$$

were $\boldsymbol{x_{1:2}} = [x_1(t), x_2(t)]$ is the position of the robot at time t, $\tilde{\boldsymbol{u}}_{0:T}$ is the discrete input sequence to the robot. T is the ending time horizon of the trajectory, $g_x(\cdot)$ are cost functions and p is the number of persons in the area. The position and orientation of all persons at time t is predicted from the knowledge of the trajectories until the current moment ($\tilde{\mathcal{P}}_{start:0}$), and given by $\mathcal{P}(t)$. $\boldsymbol{\mu}_j$ is the center of the j-th person at a given time.

To be able to calculate the cost of a trajectory according to Eq. (15), only the prediction of the person trajectories, $\mathcal{P}(t)$, remains to be defined. A simple model is that the person will continue with the same speed and direction (Bruce & Gordon, 2004). More advanced human motion models could be used without changing the planning algorithm, but it is outside the scope of this chapter to derive a complex human motion model.

8. RRT Based trajectory planning

It has been chosen to solve the above described minimisation problem by using a Rapidly-exploring Random Tree algorithm (LaValle, 2006; LaValle & Kuffner Jr, 2001). The RRT algorithm is in Svenstrup et al. (2010) modified in several ways to obtain an algorithm that works for planning a trajectory for the given problem. The overall mission of the robot is, to move forward through the environment with a desired average speed and direction, which can be set by a top level planner. This chapter contributes by enhancing the basic RRT planning algorithm to accommodate for navigation in a potential field and take into account the kinodynamic constraints of the robot. The RRT is expanded using a control strategy, which ensures feasibility of the trajectory and a better coverage of the configuration space. The planning is done in $\mathcal{C} - \mathcal{T}$ space using an Model Predictive Control (MPC) scheme to incorporate the dynamics of the environment.

The structure of the proposed planning algorithm for planning a trajectory through a dynamic environment, as shown in Fig. 5, can be seen in Fig. 6. Comparing to Fig. 1, this figure corresponds to the *robot motion* block. The idea is that while a part of a calculated trajectory is executed, a new is calculated on-line. Input to the trajectory planner is the previous best trajectory, the person trajectory estimates, and the dynamic model of the robot.

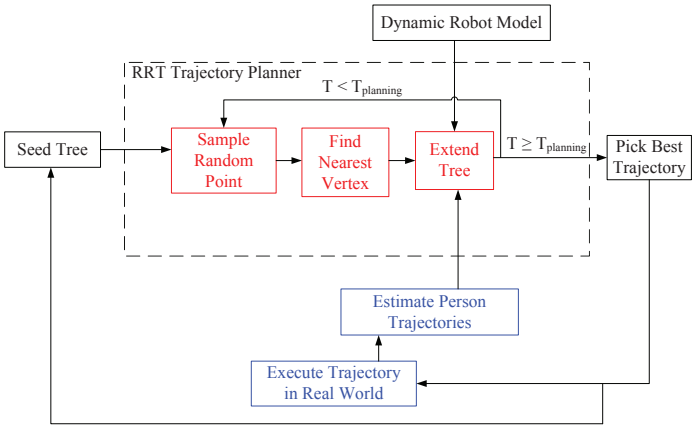


Fig. 6. The overall structure of the trajectory generator. The blue real world part and the red trajectory planning part are executed simultaneously.

The minimisation problem stated in Eq. (15) is addressed by a Rapidly-exploring Random Tree (RRT) algorithm. A standard RRT algorithm is shown in Algorithm 1, where the lines 4, 5, 6 correspond to the three blocks in the larger *RRT Trajectory Planner* box in Fig. 6.

Algorithm 1 Standard RRT (see Ferguson & Stentz (2006))

RRTmain()

```
1: Tree = q.start
2: q.new = q.start
3: while Dist(q.new , q.goal) < ErrTolerance do
4:     q.target = SampleTarget()
5:     q.nearest = NearestVertex(Tree , q.target)
6:     q.new = ExtendTowards(q.nearest,q.target)
7:     Tree.add(q.new)
8: end while
9: return Trajectory(Tree,q.new)
```

SampleTarget()

```
1: if Rand() < GoalSamplingProb then
2:     return q.goal
3: else
4:     return RandomConfiguration()
5: end if
```

The method presented here differs from the standard RRT in lines 1, 3, 6, 9, which are marked red. Furthermore, between line 6 and 7, node pruning is introduced. Since an MPC scheme is used, only a small portion of the planned trajectory is executed, while the planner is restarted to plan a new trajectory on-line. When the small portion has been executed, the planner has an updated trajectory ready. To facilitate this, the stopping condition in line 3 is changed. When a the robot needs a new trajectory, or when certain maximum number of vertices have been extended, the RRT is stopped. Even though the robot only executes a small part of the

trajectory, the rest of the trajectory should still be valid. Therefore, in line 1, the tree is seeded with the remaining trajectory.
In line 9 the trajectory with the least cost according to Eq. (15) is returned, instead of returning the trajectory to the newest vertex. The tree extension function and the pruning method are described below.

8.1 RRT Control input sampling

When working with nonholonomic kinodynamic constrained systems, it is not straightforward to expand the tree towards a newly sampled point in the configuration space (line 6 in Algorithm 1). It is a whole motion planning problem in itself to find inputs, that drive the robot towards a given point Siciliano & Khatib (2008). The algorithm proposed here uses a controller to turn the robot towards the sampled point and to keep a desired speed. A velocity controller is set to control the speed towards an average speed around a reference velocity. The probabilistic completeness of RRT's in general, is ensured by the randomness of the input. So to maintain this randomness in the input signals, a random value sampled from a Gaussian distribution, is added to the controller input. The velocity controller is implemented as a standard proportional controller. The rotation angle is a second order system with θ and $\dot{\theta}$ as states, and therefore a state space controller is used for control of the orientation. The control input can be written as:

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} k_v(\mu_v - v(t)) \\ k_{\theta 1}(\phi_{point} - \theta(t)) - k_{\theta 2}\dot{\theta} \end{bmatrix} + \begin{bmatrix} \mathcal{N}(0, \sigma_v) \\ \mathcal{N}(0, \sigma_\theta) \end{bmatrix}. \tag{16}$$

μ_v is the desired average speed of the robot and ϕ_{point} is the angle towards the sampled point. $k_{(\cdot)}$ are controller constants and σ_v, σ_θ are the standard deviations of the added Gaussian distributed input.
The new vertex to be added to the tree is now found by using the dynamic robot motion model and the controller to simulate the trajectory one time step. This ensures that the added vertex will be reachable.

8.2 Tree Pruning and trajectory selection

A simple pruning scheme, based on several different properties of a node, is used. If the vertex corresponding to the node ends up in a place where the potential field has a value above a specific threshold, then the node is not added to the tree. Furthermore a node is pruned if $|\theta(t)| > \frac{\pi}{2}$, which means that the robot is on the way back again, or if the simulated trajectory goes out of bounds of the environment. It is not desirable to let the tree grow too far in time, since the processing power is much better spend on the near future, because of the uncertainty of person positions further into the future. Therefore the node is also pruned if the time taken to reach the node is above a given threshold. Finally, instead of returning the trajectory to the vertex of the last added node, the trajectory with the lowest cost (calculated from Eq. (15)), is returned. But to avoid the risk of selecting a node, which is not very far in time, all nodes with a small time are thrown away before selecting the best node.
The final algorithm is shown in Algorithm 2.

9. Experiments and simulations

The pose and intention estimation algorithm together with the adaptive human-aware navigation is demonstrated to work in experiments with random test persons. Additionally

Algorithm 2 Modified RRT for human environments

RRTmain()

```
Tree = q.oldBestTrajectory
while (Nnodes < maxNodes) and (t < tMax) do
    q.target = SampleTarget()
    q.nearest = NearestVertex(Tree , q.target)
    q.new = CalculateControlInput(q.nearest,q.target)
    if PruneNode(q.new) == false then
        Tree.add(q.new)
    end if
end while
return BestTrajectory(Tree)
```

SampleTarget()

```
if Rand < GoalSamplingProb then
    return q.goal
else
    return RandomConfiguration()
end if
```

the trajectory planning through densely populated environments is demonstrated in a simulated pedestrian street environment.

9.1 Experimental set-up

The person state estimation and human-aware navigation subsystems are tested decoupled from the rest of the system in a laboratory setting in Svenstrup et al. (2009). Therefore, the experiments here focus on testing the function of the combined system including the intention estimation. The experiments are designed to illustrate the operation and the proof of concept of the combined methods. It took place in an open hall, with only one person at a time in the shared environment. This allowed for easily repeated tests with no interference from other objects than the test persons. The test persons were selected randomly from students on campus. None had prior knowledge about the implementation of the system.

9.1.1 Test equipment and implementation

The robot used during the experiments was a FESTO Robotino platform, which provides omnidirectional motion. A head, which is capable of showing simple facial expressions, is mounted on the robot (see Fig. 7). The robot is equipped with an URG-04LX line scan laser range finder for detection and tracking of people, and a button to press for emulating the transition to close interaction (see Fig. 7). If the test persons passed an object to the robot, they would activate the button, which was perceived as an interest in close interaction. If the test person did not pass an object (i.e. the button was not activated) within 15 seconds or disappeared from the robot's field of view, this was recognised as if no close interaction had occurred, and thus that the person was not interested in interaction.

9.2 Test specification

For evaluation of the proposed methods, two experiments were performed. During both experiments the full system, as seen in Fig 1, is used, i.e. the pose estimation and the human-aware navigation are running, as well as the interest learning and evaluation. All

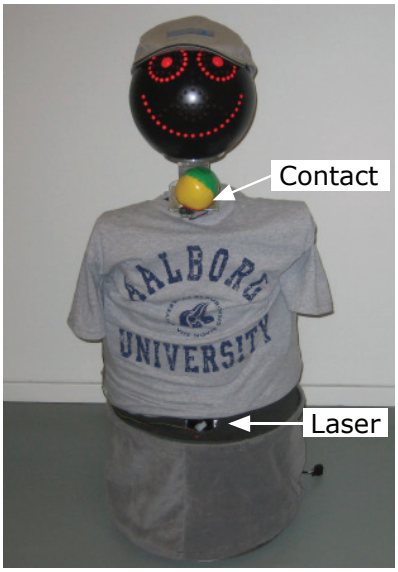


Fig. 7. The FESTO Robotino robot used for the experiments.

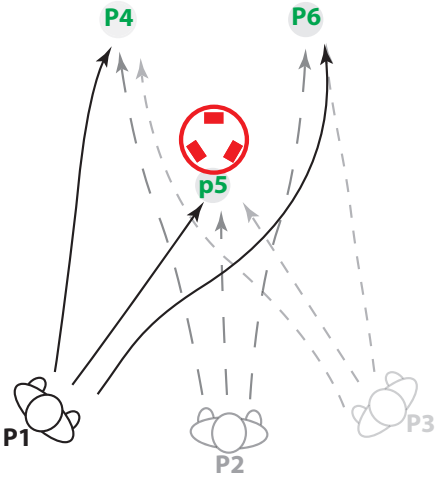


Fig. 8. Illustration of possible pathways around the robot during the experiment. A test person starts from points P1, P2 or P3 and passes through either P4, P5 or P6. If the trajectory goes through P5, a close interaction occurs by handing an object to the robot.

output values (*PI*), and input values (pose and velocity) were logged for later analysis during both experiments.

In Experiment 1, the objective was to see if the system was capable of learning to estimate *PI* based on interaction experience from several different people. As the number of cases increase, the system should be better able to estimate *PI* of a person and do it more quickly. Furthermore, the information in the CBR database should be generic, such that information obtained with some people can be used when other people occur. Starting from a CBR system with no knowledge, i.e. an empty database, a total of five test persons were asked to approach or pass the robot 12 times using different motion paths, which are illustrated in Fig. 8. The starting and end points of each session were selected randomly, while the specific route was chosen by the test person. The random selection was designed so the test persons would end up with close interaction in 50% of the sessions. In the other sessions, the test persons would pass the robot either to the left of the right without interacting.

In Experiment 2, the objective was to test the adaptiveness of the CBR system. The system should be able to change its estimation of *PI* over time if behaviour patterns change. A total of 36 test approaches were performed with one test person. The test person would start randomly at P1, P2 or P3 (see Fig. 8) and end the trajectory at P4, P5 or P6. In the first 18 sessions the test person would indicate interest in close interaction by handing an object to the robot from P5, while in the last 18 sessions the person did not indicate interest and the trajectory ended at P4 or P6.

9.3 Simulations in crowded environments

The algorithm for trajectory planning in crowded environments, described in Section 7, is implemented and demonstrated to work on a simulated pedestrian street, as shown in Fig. 5. The experiments consist of two parts. First, the algorithm is applied on the environment shown in Fig. 5. This will demonstrate that the algorithm is capable of planning a trajectory, which does not collide with any persons. It will also demonstrate how the tree expands. Next, a simulated navigation through several randomly generated worlds is performed. This will demonstrate the robustness of the algorithm over time. Specific parameters for the algorithm can be seen in Svenstrup et al. (2010).

9.3.1 Robustness test

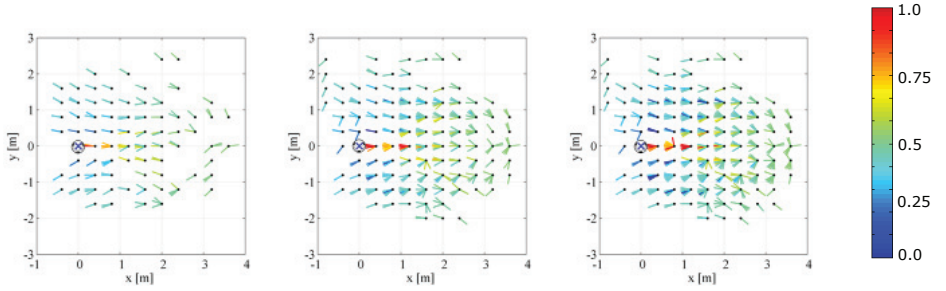
The robustness test is performed in 50 different randomly generated environments, where the robot has to navigate forwards in during a one minute period. With an average speed of $1.5\frac{m}{s}$, this corresponds to the robot moving approximately $90m$ ahead along the street. In each simulation the robot's initial state is at position $(0, 0)$ and with zero velocity.

Initially a random number of persons (between 10 and 20) are placed randomly in the world. Their velocity is sampled randomly from a Gaussian distribution. The motion of each person is simulated as moving towards a goal $10m$ ahead of them. The goal position of the $y - axis$ of the street is sampled randomly, and will also change every few seconds. Additional Brownian motion is added to each person to include randomness of the motion. Over time new persons will enter at the end of the street according to a Poisson process. This means that at any given time, persons will appear and disappear at the ends of the street. Because of this randomness, the number of persons can differ from the initial number of persons, and ranges from 10 to around 40, which is different for each simulation.

At each time instant, the robot will only know the current position and velocity of each of the persons within a range of 45m in front of the robot, and it has no knowledge about where the persons will go in the future.
As it is a simulation, there are no real time performance issues, and nothing has been done to optimise the code for faster performance. So the tree is set to grow a fixed number of 2000 vertices at each iteration. The planning horizon is set to 20 seconds and at each iteration the robot executes 2 seconds of the trajectory, while a new trajectory is planned.

10. Results

10.1 Experiment 1



(a) After 1st test person (b) After 3rd test person (c) After 5th test person (d) PI

Fig. 9. The figures show the values stored in the CBR system after completion of the 1st, 3rd and 5th test person. Note that the robot is located at the origin $(0, 0)$, since the measurements are in the robot's coordinate frame, which follows the robot's motion. Each dot represents a position of the test person in the robot's coordinate frame. The direction of the movement of the test person is represented by a vector, while the level (PI) is indicated by the colour range.

As interaction sessions are registered by the robot, the database is gradually filled with cases. All entries in the database after different stages of training are illustrated by four-dimensional plots in Fig. 9. The (x, y) coordinates are the position coordinates (p_x, p_y) of the person in the robot's coordinate frame, as estimated by the pose estimation algorithm. The coordinates are the dots in the 40 x 40cm grid. At each position, the orientation of the person (θ) is illustrated by a vector. The colour of the vector denotes the value of PI. Blue indicates that the person does not wish close interaction, while red indicates that the person wishes to engage in close interaction, i.e. $PI = 0$ and $PI = 1$ respectively. A green vector indicates $PI = 0.5$.
Figs. 9((a)-(c)) show how information in the database evolves during the experiment. Fig. 9(a) shows the state of the database after the first person has completed the 12 sessions. Here, the database is seen to be rather sparsely populated with PI values mostly around 0.5, which means that the CBR system is not well trained yet. Fig. 9(b) shows the state after 3 test persons have completed the 12 sessions and, finally, Fig. 9(c) shows all cases (around 500) after all 5 test persons have completed the sessions. Here the database is more densely populated, and PI values are in the whole range from 0 to 1. As can be seen in Figs. 9((a)-(c)), the number of plotted database entries increases as more interaction sessions occur. This shows the development of the CBR system, and clearly illustrates how the CBR system gradually learns from each person interaction session.

In all three figures (Figs. 9((a)-(c))), the vectors in the red colour range (high *PI*) are dominant when the orientation of the person is towards the robot, while there is an excess of vectors not pointing directly towards the robot in the blue colour range (low *PI*). This complies with intuition and reflects that a person, who wishes to interact, has a trajectory moving towards the robot, which is as expected for a normal human-human interaction process.
In Fig. 10, the *PI* development for six individual sessions, are plotted as a function of time. This is done for test persons 1, 3 and 5. *PI* is plotted twice for each test person: once for a randomly selected session where the test person wishes interaction with the robot; and once for a randomly selected session where the test person passes the robot with no interest in interaction. For the first test person, *PI* increases to a maximum around 0.65 for a session ending with a close interaction. For the same test person, *PI* drops to a minimum of 0.48 for a session where no close interaction occurs. For the 3rd test person, *PI* ends with a value around 0.9 for a session where close interaction has occurred, while $PI = 0.35$ for a session where no close interaction has occurred. For the last test person, *PI* rapidly increases to a value around 1 for a session where close interaction occurs, and has *PI* around 0.18 when the person is not interested in interaction. Generally this illustrates that *PI* is estimated more quickly and with more certainty the more the system is trained. It also can be seen how maximum and

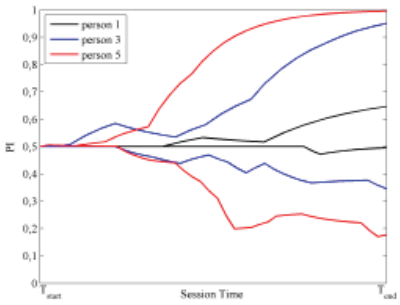


Fig. 10. *PI* as a function of the session time for three different test persons. For each test person, *PI* is plotted for a session where close interaction occurs and for a session where no close interaction occurs. The *x*-axis shows the session time. The axis is scaled such that the plots have equal length.

minimum values for *PI* increase as more test persons have been evaluated. After evaluating one test person, the robot has gathered very little interaction experience, and has difficulties in determining the correspondence between motion pattern and end result - hence *PI* stays close to 0.5. After the third test person, the robot has gathered more cases and, therefore, improves in estimating the outcome of the behaviour. For the last test person, the robot is clearly capable of estimating the intention state of the person, and thereby able to predict what will be the outcome of the interaction session.

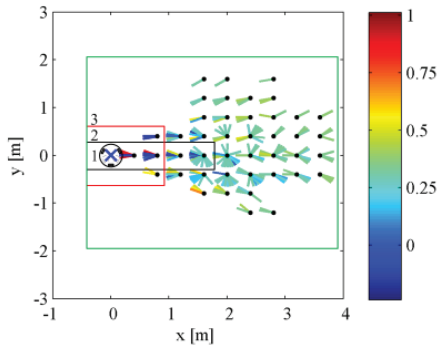
10.2 Experiment 2

Experiment 2 tests the adaptiveness of the CBR system. In order to see the change in the system over time, the average *PI* value in the database after each session is calculated. The averages have been calculated as an average for three different areas (see Fig. 11(a)), to be able to compare how the database changes in different areas relative to the robot. The areas are:

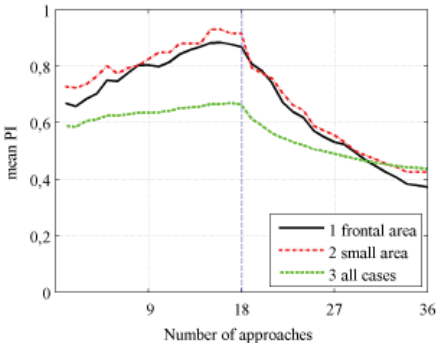
- **Area 1:** The frontal area just in front of the the robot.

- **Area 2:** A small area around the robot, which includes some of the frontal area, and some to the sides as well.
- **Area 3:** All cases stored in the database.

Fig. 11(b) shows the development of the average values of *PI* for the 36 interaction sessions for one person.



(a) A snapshot of the database after the second experiment was done. It shows how the mean value for *PI* is calculated for three areas: 1) the frontal area; 2) the small area and; 3) for all cases. The development of the mean values over time for all three areas are illustrated in Fig. 11(b)



(b) Graph of how the average of *PI* evolves for the three areas indicated in Fig. 11(a). 36 person interaction sessions for one test person is performed. The red vertical line illustrates where the behaviour of the person changes from seeking interaction to not seeking interaction.

Fig. 11. Plots of the database and how *PI* evolves from experiment 2

As can be seen from Fig. 11(b), the average value of *PI* increases for the first 18 sessions, where the person is interested in close interaction. This is especially the case for areas 1 and 2, which have a maximum value at 0.9 and 0.85 respectively, but less for area 3 (around 0.65). After 18 sessions, where the behaviour is changed, *PI* starts to drop for all areas. Most notably, area 1 drops to a minimum of 0.39 after 36 sessions. This is because in most sessions the person's trajectory goes through the frontal area, thereby having the highest number of updates of *PI*. To sum up, these experiments show that:

- Determination of *PI* improves as the number of CBR case entries increases, which means that the system is able to learn from experience.
- The CBR system is independent of the specific person, such that experience based on motion patterns of some people, can be used to determine the *PI* of other people.
- The algorithm is adaptive, when the general behaviour of the people changes.

Generally, the conducted experiments show that CBR can be applied advantageously to a robot, which needs to evaluate the behaviour of a person. The method for assessment of the person's interest in interaction with the robot is based on very limited sensor input. This is encouraging as the method may easily be extended with support from other sensors, such as computer vision, acoustics etc.

10.3 Trajectory generation results

An example of a grown RRT, with 2000 vertices, from the initial state can be seen in Fig. 12. The simulated trajectories of the robot are the red lines, and the red dots are vertices of the tree. It is seen how the RRT is spread out to explore the configuration space, although only every 10th vertex is plotted to avoid clutter on the graph. Note that the persons are static at their initial position on the figure, and some trajectories seem to pass through persons. But in reality, the persons have moved when the robot passes the point of the trajectory. The best of the all trajectories, which is calculated using Eq. (15), is the green trajectory.

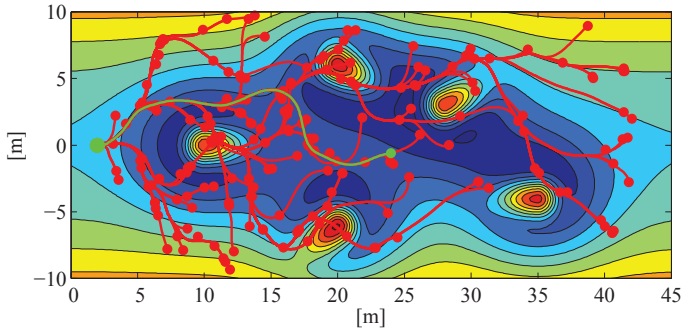


Fig. 12. An RRT for a robot starting at $(2, 0)$ and the task of moving forward through the human populated environment. Only every 10th vertex is shown to avoid clutter of the graph. The vertices are the red dots, and the lines are the simulated trajectories. The green trajectory is the least cost trajectory.

In Fig. 13 a typical scene from one of the 50 simulations can be seen. The blue dots are persons, and the arrows are velocity vectors, with a length proportional to the speed. The black star with the red arrow, is the robot position and velocity.

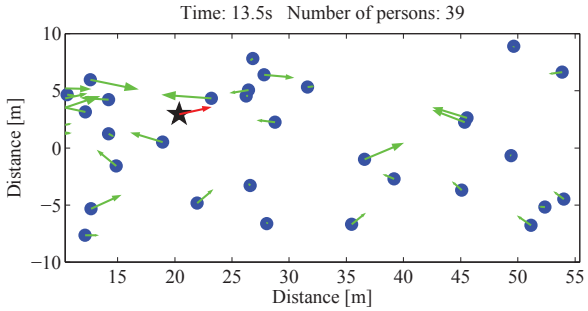


Fig. 13. A scene from one of the 50 simulations. The blue dots are persons, with their corresponding current velocity vectors. The black star is the robot.

The distance to the closest person over time, for a randomly selected simulation is shown on Fig. 14. Though it is only one sample, it is representative for all the simulations. It can be seen that very rarely the robot is closer to any person than $1.2m$, which is the distance of the personal zone of the humans.

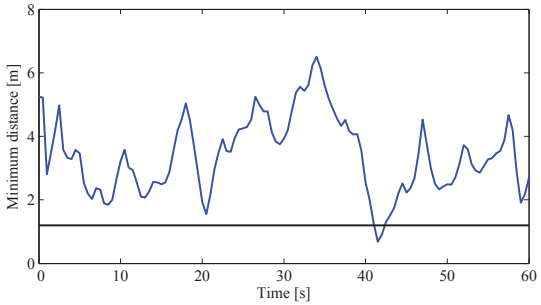


Fig. 14. The figure shows the closest distance of any person for one randomly chosen simulation. The horizontal black line, is at 1.2m distance, which corresponds to the transition between the social zone and the personal zone, which the robot should try to stay out of.

In none of the 50 one minute period simulations the robot ran into a person. This demonstrates that the algorithm is robust enough plan safe trajectories through an environment, where human motion is governed by changing goals and additional random motion, even though using a simple human motion model for prediction, when planning. A few close passes in the combined 50 minute run, down the pedestrian street, are seen. The time in each zone, in which the robot is located, for the person that is closest to the robot at any given time, is shown in Fig. 15. It is seen that the robot stays out of the personal zone of any of the persons in more than 97.5% of the time, and out of the intimate zone (a distance closer than 0.45m) 99.7% of the time. This only occurs on 9 separate instances of the 50 minutes of driving.

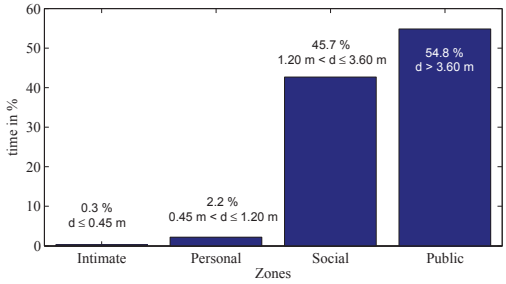


Fig. 15. The bar plot shows how large a part of the time the closest person to the robot has been in each zone. The robot should try to stay out of the personal and intimate zones. The distance interval for each zone, can also be seen in the plot.

Except in very densely populated environments, the model runs approximately one third of real time on a 2.0 GHz CPU running MATLAB. This is considered to be reasonable, since no optimisation for speed has been done. In very dense environments, the planning takes longer, since many new added vertices, are pruned again, and hence more points has to be sampled before 2000 vertices are expanded. Additionally the more persons in the area, the longer it takes to evaluate the cost of traversing the potential field. Generally, it is demonstrated, that the algorithm is able to plan and execute a trajectory through a human environment without colliding with the people walking in the environment.

11. Conclusions

In this work, we have described an adaptive system for safe and natural motion interaction between mobile robots and humans. The system forms a basis for human-aware navigation respecting a person's social spaces. The system consists of four independent components:

- A method for pose estimation of a human using laser rangefinder measurements.
- Learning to interpret human behaviour using motion patterns and Case-Based Reasoning (CBR).
- An adaptive human-aware navigation algorithm based on a potential field.
- A trajectory planning algorithm for planning a safe trajectory through a crowded human environment.

A Kalman filter based algorithm is used to derive pose estimates for the people in the environment. The pose estimates are used in a CBR system to estimate the person's interest in interaction. The spatial behaviour strategies of the robot are adapted according to the estimated interest by using adaptive potential field functions to govern the motion.
Furthermore a new algorithm for trajectory planning for a kinodynamic constrained robot in a human environment, has been described. The robot is navigating in a highly dynamic environment, which in this case is populated with humans. The algorithm, which is based on RRT's, enables the costs of traversing a potential field to be minimised, thereby supporting planning of comfortable and natural trajectories. A new control input sampling strategy, that together an MPC scheme is used to enable the planner to continuously plan a reachable trajectory on an on-line robotic system, has been presented.
The evaluation of the system has been conducted through two experiments in an open environment and simulation of robot navigation in a crowded environment. The first of the two experiments of the combined system shows that the CBR system is able to gradually learn from interaction experience. The experiment also shows how motion patterns from different people can be stored and generalised in order to predict the outcome of an interaction session with a new person. The second experiment shows how the estimated interest in interaction adapts to changes in behaviour of a test person. The trajectory generation algorithm proved in simulation that it is able to avoid collision with people in the environment. The algorithm is challenged, although still avoiding collisions, when the environments become very densely populated, but so are humans. Humans react by mutual adaptation and allowing violation of the social zones. This is not done here, where the robot takes on all the responsibility for finding a collision free trajectory.
The presented system is a step forward in creating socially intelligent robots, capable of navigating in everyday environments and interacting with human beings by understanding their interest and intention. In the long-term perspective, the results could be applied to service or assistive robot technology.

12. References

Althaus, P., Ishiguro, H., Kanda, T., Miyashita, T. & Christensen, H. (2004). Navigation for human-robot interaction tasks, *Proceedings of IEEE International Conference on Robotics and Automation, 2004. ICRA '04.*, Vol. 2, pp. 1894–1900.

Brooks, A., Kaupp, T. & Makarenko, A. (2009). Randomised mpc-based motion-planning for mobile robot obstacle avoidance, *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 3962–3967.

Bruce, A. & Gordon, G. (2004). Better motion prediction for people-tracking, *Robotics and Automation, 2004. ICRA '04. IEEE International Conference on*.

Bruce, A., Nourbakhsh, I. & Simmons, R. (2001). The role of expressiveness and attention in human-robot interaction, *In Proceedings, AAAI Fall Symposium*.
URL: *citeseer.ist.psu.edu/article/bruce02role.html*

Christensen, H. I. & Pacchierotti, E. (2005). Embodied social interaction for robots, *in* K. Dautenhahn (ed.), *AISB-05*, Hertfordshire, pp. 40–45.

Cielniak, G., Treptow, A. & Duckett, T. (2005). Quantitative performance evaluation of a people tracking system on a mobile robot, *Proc. 2nd European Conference on Mobile Robots*.

Dautenhahn, K., Walters, M., Woods, S., Koay, K. L., Sisbot, E. A., Alami, R. & Simï£¡on, T. (2006). How may i serve you? a robot companion approaching a seated person in a helping context, *HRI Human Robot Interaction '06 (HRI06)*, Salt Lake City, Utah, USA.

Dornaika, F. & Raducanu, B. (2008). Detecting and tracking of 3d face pose for human-robot interaction, *Proceedings IEEE International Conference on Robotics and Automation*, pp. 1716–1721.

Feil-Seifer, D. & Mataric, M. (2005). A multi-modal approach to selective interaction in assistive domains, *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pp. 416–421.

Ferguson, D., Kalra, N. & Stentz, A. (2006). Replanning with rrts, *Proc. IEEE International Conference on Robotics and Automation ICRA 2006*, pp. 1243–1248.

Ferguson, D. & Stentz, A. (2006). Anytime rrts, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5369–5375.

Fod, A., Howard, A. & Mataric, M. J. (2002). Laser-based people tracking, *In Proc. of the IEEE International Conference on Robotics & Automation (ICRA*, pp. 3024–3029.

Govea, D. A. V. (2007). *Incremental Learning for Motion Prediction of Pedestrians and Vehicles*, PhD thesis, Institut National Polytechnique de Grenoble.

Hall, E. (1966). *The Hidden Dimension*, Doubleday.

Hall, E. T. (1963). A system for the notation of proxemic behavior, *American anthropologist* 65(5): 1003–1026.

Hanajima, N., Ohta, Y., Hikita, H. & Yamashita, M. (2005). Investigation of impressions for approach motion of a mobile robot based on psychophysiological analysis, *IEEE International Workshop on Robots and Human Interactive Communication ROMAN 2005*, pp. 79–84.

Hansen, S. T., Svenstrup, M., Andersen, H. J. & Bak, T. (2009). Adaptive human aware navigation based on motion pattern analysis, *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, Toyama, Japan, pp. –.

Jenkins, O. C., Serrano, G. G. & Loper, M. M. (2007). *Recognizing Human Pose and Actions for Interactive Robots*, I-Tech Education and Publishing, chapter 6, pp. 119–138.

Jurisica, I. & Glasgow, J. (1995). Applying case-based reasoning to control in robotics, *3rd Robotics and Knowledge-Based Systems Workshop, St. Hubert Quebec*.

Kelley, R., Tavakkoli, A., King, C., Nicolescu, M., Nicolescu, M. & Bebis, G. (2008). Understanding human intentions via hidden markov models in autonomous mobile robots, *HRI '08: Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, ACM, New York, NY, USA, pp. 367–374.

Kirby, R., Forlizzi, J. & Simmons, R. (2007). Natural person-following behavior for social robots, *Proceedings of Human-Robot Interaction*, pp. 17–24.

Kirby, R., Simmons, R. & Forlizzi, J. (2009). Companion: A constraint-optimizing method for person-acceptable navigation, *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pp. 607 –612.

Kleinehagenbrock, M., Lang, S., Fritsch, J., Lomker, F., Fink, G. & Sagerer, G. (2002). Person tracking with a mobile robot based on multi-modal anchoring, *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication (ROMAN), 2002.* 1: 423–429.

LaValle, S. (2006). *Planning algorithms*, Cambridge Univ Pr.

LaValle, S. & Kuffner Jr, J. (2001). Randomized kinodynamic planning, *The International Journal of Robotics Research* 20(5): 378.

Likhachev, M. & Arkin, R. (2001). Spatio-temporal case-based reasoning for behavioral selection, *Proc. IEEE International Conference on Robotics and Automation, ICRA*, Vol. 2, pp. 1627–1634.

Michalowski, M., Sabanovic, S. & Simmons, R. (2006). A spatial model of engagement for a social robot, *The 9th International Workshop on Advanced Motion Control, AMC06*, Istanbul.

Munoz-Salinas, R., Aguirre, E., Garcia-Silvente, M. & Gonzalez, A. (2005). People detection and tracking through stereo vision for human-robot interaction, *MICAI 2005: Advances in Artificial Intelligence* 3789/2005: 337–346.

Ram, A., Arkin, R. C., Moorman, K. & Clark, R. J. (1997). Case-based reactive navigation: A case-based method for on-line selection and adaptation of reactive control parameters in autonomous robotic systems, *IEEE Transactions on Systems, Man, and Cybernetics* 27: 376–394.

Rodgers, J., Anguelov, D., Pang, H.-C. & Koller, D. (2006). Object pose detection in range scan data, *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* 2: 2445–2452.

Siciliano, B. & Khatib, O. (2008). *Handbook of Robotics*, Springer-Verlag, Heidelberg.

Sisbot, E. A., Alami, R., Simï£¡on, T., Dautenhahn, K., Walters, M., Woods, S., Koay, K. L. & Nehaniv, C. (2005). Navigation in the presence of humans, *IEEE-RAS International Conference on Humanoid Robots (Humanoids 05)*, Tsukuba, Japan.

Sisbot, E. A., Clodic, A., Urias, L., Fontmarty, M., Brï£¡thes, L. & Alami, R. (2006). Implementing a human-aware robot system, *IEEE International Symposium on Robot and Human Interactive Communication 2006 (RO-MAN 06)*, Hatfield, United Kingdom, pp. 727–732.

Svenstrup, M., Bak, T. & Andersen, H. J. (2010). Trajectory planning for robots in dynamic human environments, *IROS 2010: The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan.

Svenstrup, M., Bak, T., Maler, O., Andersen, H. J. & Jensen, O. B. (2008). Pilot study of person robot interaction in a public transit space, *Proceedings of the International Conference on Research and Education in Robotics - EUROBOT 2008*, Springer-Verlag GmbH, Heidelberg, Germany, pp. 120–131.

Svenstrup, M., Hansen, S. T., Andersen, H. J. & Bak, T. (2009). Pose estimation and adaptive robot behaviour for human-robot interaction, *Proceedings of the 2009 IEEE International Conference on Robotics and Automation. ICRA 2009.*, Kobe, Japan, pp. 3571–3576.

Takayama, L. & Pantofaru, C. (2009). Influences on proxemic behaviors in human-robot interaction, *IROS'09: Proceedings of the 2009 IEEE/RSJ international conference on intelligent robots and systems*, IEEE Press, Piscataway, NJ, USA, pp. 5495–5502.

van den Berg, J. (2007). *Path planning in dynamic environments*, PhD thesis, Ph. D. dissertation, Universiteit Utrecht.

Walters, M. L., Dautenhahn, K., Koay, K. L., Kaouri, C., te Boekhorst, R., Nehaniv, C., Werry, I. & Lee, D. (2005). Close encounters: Spatial distances between people and a robot of mechanistic appearance, *Proceedings of 2005 5th IEEE-RAS International Conference on Humanoid Robots*, Tsukuba, Japan, pp. 450–455.

Walters, M. L., Dautenhahn, K., te Boekhorst, R., Koay, K. L., Kaouri, C., Woods, S., Nehaniv, C., Lee, D. & Werry, I. (2005). The influence of subjectsï£¡ personality traits on personal spatial zones in a human-robot interaction experiment, *Proc. IEEE Ro-man*, Hashville, pp. 347–352.

Woods, S., Walters, M. L., Koay, K. & Dautenhahn, K. (2006). Methodological issues in hri: A comparison of live and video-basedmethods in robot to human approach direction trials, *The 15th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN06)*, Hatfield, UK, pp. 51–58.

Xavier, J., Pacheco, M., Castro, D., Ruano, A. & Nunes, U. (2005). Fast line, arc/circle and leg detection from laser scan data in a player driver, *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 3930–3935.

Zucker, M., Kuffner, J. & Branicky, M. (2007). Multipartite rrts for rapid replanning in dynamic environments, *Proc. IEEE International Conference on Robotics and Automation*, pp. 1603–1609.