A Review of Social-Aware Navigation Frameworks for Service Robot in Dynamic Human Environments

S. F. Chik[1], C. F. Yeong[2], E. L. M. Su[1] T. Y. Lim[3], Y. Subramaniam[4], P. J. H. Chin[4]

[1]*Faculty of Electrical Engineering Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia.*

[2]*Centre for Artificial Intelligence and Robotics, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia.*

[3]*Malaysia Japan Institute of Technolgy (MJIIT), Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia.*

[4]*DF Automation and Robotics Sdn. Bhd., Taman Impian Emas, 81310 Skudai, Johor, Malaysia.*

sfchik91@gmail.com

***Abstract*—The emergence of service robot into human daily life in the past years has opened up various challenges including human-robot interaction, joint-goal achievement and machine learning. Social-aware navigation also gains vast research attention in enhancing the social capabilities of service robots. Human motions are stochastic and social conventions are very complex. Sophisticated approaches are needed for a robot to abide to these social rules and perform obstacle avoidance. To maintain the level of social comfort and achieve a given task, the robot navigation is now no longer a search for a shortest collision-free path, but a multi-objective problem that requires a unified social-aware navigation framework. A careful selection of navigation components including global planner, local planner, the prediction model and a suitable robot platform is also required to offer an effective navigation amidst the dynamic human environment. Hence, this review paper aims to offer insights for service robot implementation by highlighting four varieties of navigation frameworks, various navigation components and different robot platforms.**

***Index Terms*—Navigation Framework; Path Planner; Review; Social-Aware.**

I. INTRODUCTION

Robots are not only operate in the industry world, but also venture into human daily lives, co-exist with people in restaurants, hotels, shopping malls, hospitals and healthcare centres [1-3]. Pepper [4], a human-like robot developed by Aldebaran is able to welcome customers in shops. REEM [5], a wheeled humanoid service robot is placed in shopping malls and exhibitions to give service and entertain people. Another mobile service robot, OSHbot [6], developed by Lowe's Innovation Labs and Fellow Robots can bring customer to the location of requested products in a hardware store. To navigate in dynamic human environment, robots have to handle the stochastic human motion and abide the social conventions to avoid human-robot conflicts. According to Hall [7], human proxemics can be categorized into intimate, personal, social and public, with different proxemics reserved for specific relationship. Kendon described [8] the formation of group conversation as F-formation, which consist of o-space, p-space and r-space. For a conversation group of two people, the formations can be further represented by N-shape, vis-a-vis, V-shape, L-shape, C-shape and side-by-side, showing that social rules are indeed complex. Social-aware navigation is vital for social acceptance in these scenarios where robots have to understand and respect human cues. Navigation in dynamic human environment is therefore no longer a problem to find a collision-free path, but is a task to achieve a joint goal with human-robot mutual understanding. Kruse et al. [9] did a review on human-aware navigation, highlighting various research areas including comfort, naturalness, sociability and also discussed about navigation components in social environments. Rios-Martinez [10] on the other hand surveyed on the importance of proxemics on social robot navigation. Various navigation methods have been proposed and hence navigation frameworks are important to unite methods from different research focuses for service robot implementation. Hence, this paper highlights four varieties of frameworks, various navigation components and robot platforms used in literatures related to social-aware navigation in dynamic human environment for the past three years. The literature search is conducted using online search engines and manual search of robotics conferences and journals, limited to English language literatures. The rest of the article is organized as follows: Section II introduces the navigation components including global planner, local planner and prediction model. Section III describes different kinds of navigation frameworks for social-aware navigation. Robot platforms used in researches of social-aware navigations are introduced in Section IV. A discussion on navigation frameworks, navigation components and robot platforms is presented in Section V. Section VI concludes with a summary of the selection of navigation components, frameworks and robot platforms for service robot implementation, and provides insight for future researches.

II. NAVIGATION COMPONENTS

A. Global Planner

Global planner provides a mobile robot an optimal and collision-free route from the current position towards the goal. A global planner requires a known or partially-known static map of an environment to process before proceed to navigation.

Rapidly-Exploring Random Trees (RRT) [11], a probabilistic global planner, is well-known for path planning. RRT offers a quick solution search across the problem domain, also identified as a metric space, X through random sampling. RRT treats the path planning problem as to find a path from an initial state, x_{init} to the goal state x_{goal}. For every iteration, a state transition to x_{new} is carried out and is bounded by the criteria that $x_{new} \in X_{free}$, where x_{free} is the free region within X. Complement of X_{free} is represented by X_{obs}, where X_{obs} can be the obstacle region or any configuration where a robot will end up with a collision. The state transition equation is as follows:

$$\dot{x} = f(x,u) \tag{1}$$

where $\dot{x}$ is the derivative of state with respect to time and u is a vector input from a set U required to transit from current state, x to x_{new}. Vertex generated by x_{new} and edge created between x_{new} and x are recorded for further expansions until a path is form between x_{init} and x_{goal}. The advantage of RRT is that f(x,u) is able to account for kinematic and dynamical constraints during state transition, which is suitable for many practical applications. Shrestha et al. used RRT [12] as global planner to plan path in an environment with human. Pérez-Higueras et al. [13] suggested RRT can also be used as local planner instead, due to its real-time capability which is very crucial to result in an effective human-avoidance. To further improve RRT path planner, researcher proposed different variants including RRT* [14] and dual-tree RRT (DT-RRT) [15].

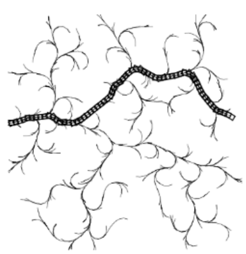

Figure 1: A path planned using RRT for kinodynamic car [11]

Another popular global planner is the A* search algorithm [16], which is a deterministic planner that utilizes the distance between the current processing node and the goal node on the solution space as heuristic [17] components to return a globally shortest path as shown in Figure 2. The cost function of A* algorithm is as follows:

$$f(n) = g(n) + h(n) \tag{2}$$

where *n* is the current node, g(n) is the path cost from the start node to node n and h(n) is the cost estimation of the cheapest path from node n to the goal. A* algorithm treats nodes with different states: open, closed and unvisited, then places them into respective lists. A* algorithm works by placing the current checking node n into the closed list while its surrounding nodes are put under the open list, then the cost function for each surrounding node is calculated. These surrounding nodes are the child nodes and are paired to the current checking node n, or simply the parent node. The next checking node is then selected from the open list with the smallest value of the cost function. Again, the cost function for each surrounding node is calculated and paired to the new node n. Special check is required for surrounding node that is already in the open list, whether the previous or the new path cost g(n) to that node is lower. If the current path cost is lower, then that node has to be paired to that new node n. This process is repeat until the goal node lies beside the current node n, and a shortest path between the start and the goal node can be form by tracing back the paired child and parent nodes.

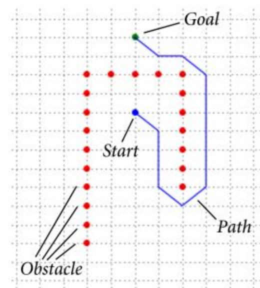

Figure 2: A* algorithm planned a shortest path on a grid environment

If the heuristic part, h(n) is omitted, the result is an algorithm namely the Dijkstra's algorithm [18]. Other variants of A* include: D* [19], Focussed D* [20], D* Lite [21] and LPA* [22]. A* algorithm and A* variants are able to return a shortest feasible path. Thus, many literatures [23-27] regarding navigation in dynamic human environments utilized A* algorithm for global planning. While some [13, 28, 29] use Dijkstra algorithm due to its simplicity when computational time is not crucial.

Table 1
A summary of Global Planners with Respective Literatures

Global Planners	Literatures	Advantages	Disadvantages
RRT	[12]	Accounts for robot kinematic and dynamical constraints	Does not result in shortest path
A* algorithm	[23-27]	Returns shortest path depending on defined grid size	Does not consider robot kinematic and dynamical constraints
Dijkstra's algorithm	[13, 28, 29]	Returns shortest path depending on defined grid size	High memory requirement and computational time for large environment
E*	[30]	Has dynamic replanning capability	Dynamic capability is redundant with the existence of local planner
Global artificial potential field	[31]	Real-time obstacle avoidance	Dynamic capability is redundant with the existence of local planner
Wavefront algorithm	[33]	Simple implementation	Greater memory requirement and computational time for large environment

Other than RRT and A* algorithm, Weinrich et al. [30] utilized E* algorithm as global planner that has dynamic replanning capability in the research of socially compliant robot navigation. In a study of indoor human monitoring, Lizuka et al. [31] used a Global Potential Field Approach that can overcome the local minimum issue in robot navigation. Wavefront algorithm or known as NF1 [32], is a simple global planner that expands the search to all adjacent nodes until the start node and goal node are covered, utilized by Oli et. al. [33] for path planning that incorporates human motion behavior. A summary of different global planners used in social-aware literatures is shown in Table 1.

B. Local Planner

Local planner focuses on collision avoidance for dynamic obstacles, where global planner could not handle efficiently. Fox et al. [34] proposed the Dynamic Window Approach (DWA), a local planner, which take account of robot kinematic and dynamic constraints. DWA algorithm plans collision-free trajectory in two steps. First, DWA reduces the search space by pruning those non-achievable velocities. This step takes account of three sets of velocities: circular trajectories, admissible velocities and dynamic window. Circular trajectories, V_s consists of velocities for the next time interval that does not intersect with an obstacle. Admissible velocities, V_a represents a set of velocities that a robot is able to stop before it reaches an obstacle. While dynamic window, V_d consists only velocities that can be reached within the next time interval. The search space V_r is then restricted by intersecting V_s, V_a and V_d. The second step of DWA is to maximize an objective function by choosing the possible velocities in V_r from step one. The objective function is as follows:

$$G(v,\omega) = \sigma \cdot (\alpha \cdot head(v,\omega) + \beta \cdot dist(v,\omega) + \gamma \cdot vel(v,\omega)) \quad (3)$$

where head(v,ω) measure the heading of the robot with the goal position, dist(v,ω) defines the distance closest obstacle detected and vel(v,ω) represents the speed of the trajectory, σ is used to normalized the weightages α, β and γ to [0,1].

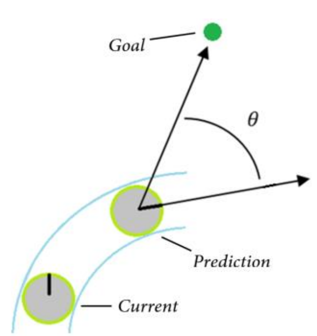

Figure 3: DWA aligns the robot current heading to goal using the angle θ from a velocity predicted position [34]

Figure 3 shows DWA is able to align the heading of the robot to the goal point using velocity predicted position. The nature of DWA that derive path from motion dynamics enables several literatures [24, 30, 33, 35, 36] successfully implemented DWA for human collision avoidance.

DWA however does not consider the velocity of the obstacle as Velocity Obstacle (VO) [37] does. VO is a planner, which generates avoidance manoeuvres by selecting the robot velocities outside the collision cone, where collision cone consists of velocities that would result in collision with obstacles moving at given velocities, at some time in future. Figure 4 shows that a size-reduced robot A and obstacle B are moving at velocities Â with magnitude V_A and $\hat{B}$ with magnitude V_B respectively. The first step to compute VO is to reduce the size of A to a point as shown in Figure 4 and enlarge obstacle B by the radius of A. Then, define the collision cone, $CC_{A,B}$ as follows:

$$CC_{A,B} = \{\vec{V}_{A,B} \mid \lambda_{A,B} \cap \hat{B} \neq \phi\} \quad (4)$$

where $\vec{V}_{A,B}$ is relative velocity of Â with respect to $\hat{B}$, calculated using $\vec{V}_{A,B} = \vec{V}_A - \vec{V}_B$ and $\lambda_{A,B}$ is the trajectory of $\vec{V}_{A,B}$.

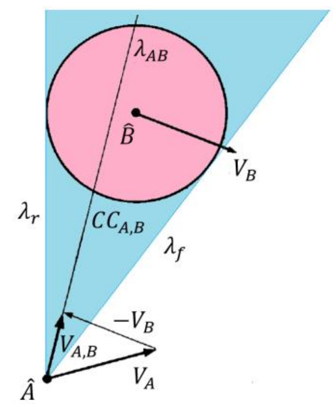

Figure 4: Collision cone $CC_{A,B}$ defined by VO for moving robot A and moving obstacle B [37]

Any relative velocity outside of the collision code $CC_{A,B}$ bounded by λ_r and λ_f is guarantee to be collision-free, provided that the obstacle $\hat{B}$ maintains its current shape and speed. VO is able to simplify a complex dynamic situation using velocity space. To obtain the absolute velocity of A, just simply add the velocity of B, $\vec{V}_B$ to each of the velocities in $CC_{A,B}$ as shown in (5) as follows:

$$VO = CC_{A,B} \oplus \vec{V}_B \quad (5)$$

where ⊕ is the Minkowski vector sum operator. Then for the case of multiple obstacles, the resulting VO is the union of the individual velocity obstacles given by (6).

$$VO = \bigcup_{i-1}^{m} VO_{B_i} \quad (6)$$

where *m* is the total number of obstacles and VO_{B_i} is the velocity obstacle for *i*th obstacles. Hence, by selecting the velocity outside of VO can ensure a collision-free path. However, VO treats the collision-avoidance as a task to be done by the robot alone, which does not imply the case where human will take reciprocal action to avoid the robot. Berg et al. proposed Reciprocal Velocity Obstacle (RVO) [38], which is an extension of VO that in a multi-agent environment, each agent moves by considering the behavior of other agents to achieve a mutual collision avoidance. DongXiang et al. [39] used a variant of RVO, the Optimal Reciprocal Collision Avoidance (ORCA) [40] to pro-actively avoid pedestrians. There are researches [28, 41] used Dijkstra's Algorithm as local planner by computing fractions of shortest path to achieve a temporal goal. X-Y-T Space [42] and time-dependent A* [26] local planners used the same concept by generating sub-goals in order to avoid human and non-human obstacles. There are some other local planners that directly incorporate the information from human prediction model (Section II C) to replicate human in collision avoidance. A summary of local planners used in social related robot navigation is presented in Table 2.

Table 2
Different Local Planners Used in Social-Aware Literatures

Local Planners	Literatures	Advantages	Disadvantages
DWA	[24, 30, 33, 35, 36]	Accounts for robot kinematic and dynamic constraints	Does not predict obstacle motion
ORCA (VO Variant)	[39]	Considers and predicts obstacle motion	Does not take account of kinematic and dynamic constraints
Dijkstra's algorithm	[28,41]	Simple implementation	Does not take account of robot kinematic and dynamic constraints and obstacle motion
X-Y-T Space	[42]	Accounts for robot kinematic and dynamic constraints and predicts obstacle motion	Performance and computational effort depends of defined grid size
Time-dependent A*	[26]	Simple implementation	Does not take account of robot kinematic and dynamic constraints and obstacle motion
With human prediction model	Table 3	Predicts human dynamic motion and accounts for social conventions	Sophisticated implementation

C. Prediction Model

Prediction of human motion further improves the effectiveness of the navigation in a populated dynamic environment. A simple way to predict human motion is by using the linear model where human trajectories are formed by mostly straight lines, as used in VO local planner.

To better represent the stochastic human behaviour, Helbing and Molnar [43] proposed the Social Force Model (SFM). SFM defines that pedestrian motions are motivated by the environment where social forces are coming from the destination, surrounding objects and other pedestrians. The general model for a pedestrian *i* is given by the social force term:

$$\frac{dv_i(t)}{dt} = \vec{F}_i(t) + \vec{\xi}(t) \tag{7}$$

where $\vec{\xi}$ is a fluctuation term to represent random variations of behaviour. The term $\vec{F}_i$ is the summation of the pedestrian's desired force towards a goal $\vec{f}_i^{goal}$ and other interacting forces $\vec{F}_i^{int}$.

$$\vec{F}_i = \vec{f}_i^{\,goal} + \vec{F}_i^{\,\text{int}} \tag{8}$$

The term $\vec{f}_i^{goal}$ defines that each pedestrian i moves in a desired velocity $\vec{v}_i^{\,0}$ and is subjected to necessary deviation of $\vec{v}_i$ and hence can be represented as follows:

$$\vec{f}_i^{\,goal} = \frac{1}{\tau_i}(\vec{v}_i^{\,0} - \vec{v}_i) \tag{9}$$

where τ_i is the relaxation time. $\vec{F}_i^{int}$ is the interacting forces applied on pedestrian i, from other pedestrians and obstacles. In the application of robot navigation, $\vec{F}_i^{int}$ includes the interaction with robot. Helbing and Molnar [43] did a software simulation on SFM with a walkway (Figure 5) of 10 meter wide and 50 meter long and a uniform pedestrian motion is observed, reflecting the real world scenario.

SFM is implemented in several literatures [23, 35, 44-47] for human-aware navigations, proving the effectiveness and practicality of the prediction model.

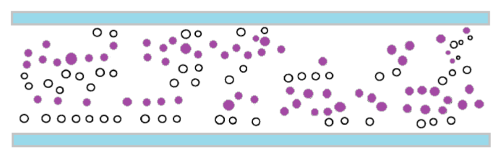

Figure 5: Pedestrian motion based on SFM where the filled and unfilled circles indicating pedestrians walking in opposite directions [43]

Another modelling method is through observing the human motion and apply feature extraction and learning using Inverse Reinforcement Learning (IRL) [48]. A brief introduction of Markov Decision Process (MDP) [49] is needed to describe IRL. MDP is an environment representation using a tuple (S,A,{P_{sa}},γ,R), where S is a finite set of N states s, A is a set that consist of k number of actions a, P_{sa}(s') is the probability of state transition from s to s' by taking action a, γ∈[0,1] is the discount factor and R is the reward function at a given state s. From this state space representation, MDP offers an optimal policy π* for state transitions based on state utility U to obtain the best reward. The optimal policy is computed as follows:

$$\pi^*(s) = \arg\max_{a \in A(s)} \sum_{s'} P_{sa}(s')U(s') \tag{10}$$

where U, the utility of state s, is computed using the following Bellman equation:

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P_{sa}(s')U(s') \tag{11}$$

MDP is a probabilistic planning technique to obtain an optimal policy of a situation with given finite states and actions by maximizing reward. IRL however deals with the inverse problem where a policy and elements in the MDP tuple (S,A,{P_{sa}},γ,R) are given except the reward function R. The task of IRL is to identify the reward function R such that π* is an optimal policy in the MDP. In the context of social-aware navigation, the finding of reward function R is to describe an observed behaviour such as a crowd motion and a demonstration from an expert. Vasquez et al. [28] did a comparative study between the Max-margin IRL [50] and Maximum Entropy IRL [51] for crowd navigation and both learning-method offering similar performance, which is better than manual weight tuning. Kim and Pineau [41] proposed a socially adaptive path planning using IRL to define the cost function for the planner, offering comfortable and safe trajectory. Kuderer et al. [25] proposed IRL for navigation in human environments, showing better results than other dynamic global planner methods. Pérez-Higueras et al. [13] also used IRL and is able to transfer human motion behaviour to a mobile robot.

There are also other modelling methods such as multi-goal Interacting Gaussian Processes (mgIGP) [52, 53] that is able to reason multiple goals of human motions. Hamiltonian Markov Chain Monte Carlo sampling (HMCMC) [54], another method that can learn the stochasticity of the observed human trajectories. Human Motion Behaviour Model (HMBM) [33], is a model that enables robot to perform human-like decision in various commonly human encountered scenario. For simplicity, several researches [26, 39, 42, 55, 56] used linear model that just consider human moves linearly. Table 3 summarized the prediction models proposed by selected literatures.

Table 3
Literatures Using Different Prediction Models to Social-Aware Navigation

Prediction Models	Literatures	Advantages	Disadvantages
Linear	[26, 39, 42, 55, 56]	Easy implementation	Unable to fully represent a human motion
SFM	[23,35,44-47]	The relationship between pedestrians, robot and static obstacles are presented in discrete components and can be adjusted separately	The environment has to be clearly known beforehand in order to predict the possible goals of the pedestrians
Learning approach (IRL, HMCMC)	[13,25,28, 41,54]	Able to model the human motion and easily adapt to different environment	Performance depends on proper selections of observed features and local planner
mgIGP	[52,53]	Able to react base on predicted human motion goal	Requires external sensor setups
HMBM	[33]	Able to perform human-like decision in various commonly human encountered scenario	Have to define every possible scenario

III. NAVIGATION FRAMEWORKS

Global planner, local planner and prediction model, each navigation component plays distinct roles in a social-aware navigation. Hence, a proper navigation framework is needed to unify these components. Different navigation frameworks are formed under the combination of different components, where this paper highlights four types of frameworks. The arrows in Figures 6 until 9 represent the data flow between components.

A. Framework 1: Sole Global Planner

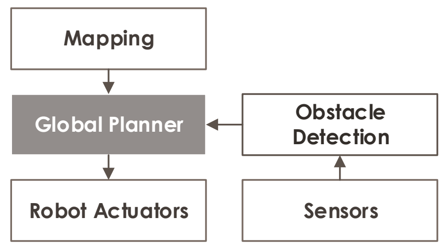

Figure 6: A Sole Global Planner navigation framework

A navigation framework with only global planner requires least implementation effort. Once the map information is obtained, the global planner computes a shortest or any feasible path from current location to the goal and gives command to the robot actuators. When an obstacle is detected, the global planners will have to repeat the path planning process, this results in move-stop-move scenario that impact the path smoothness. Although some improvements are done on global planners to account for dynamic obstacles, this framework is still insufficient to produce socially acceptable trajectories. According to Kollmitz et al. [26], paths produced by global planner often require large robot motion divergent which reduces the level of human comfort. This framework is used by Shrestha et al. [12], where replanning is required whenever a person is blocking the original path.

B. Framework 2: Global Planner with Local Planner

In order to improve collision-avoidance in path planning, reactive local planner has to be introduced into the navigation framework. The problem with local planners such as VO and DWA is that they are not suitable for stand-alone applications due to insufficient future planning. However, these local planners have collision-avoidance capability that outperform global planner with poor dynamic obstacle handling but has better look-ahead. Hence, the combination of global planner and local planner as shown in Figure 7 is able to minimize drawbacks of each method. This framework is utilized in the research work of Xia et al. [36] for navigation and exploration in human environment. However, this framework is still insufficient to deal with highly dense human environment with complex human motion.

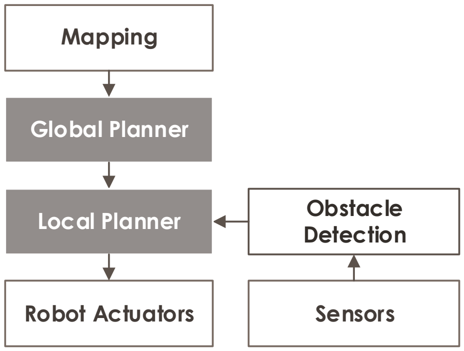

Figure 7: A Global Planner with Local Planner navigation framework

Local planner can be improved to be more pro-active in obstacle avoidance. As mentioned by DongXiang et al. [39], VO and DWA are classical local planners that avoid obstacles reactively and can be enhanced using reciprocal and pro-active collision avoidance methods such as ORCA and machine learning. Pro-active methods are able to predict the stochastic motion of human by using human behavioral models to achieve a more effective human-robot collision-avoidance. To predict human motion, several components have to be added, which will be presented in the next navigation framework.

C. Global Planner with Predictive Local Planner

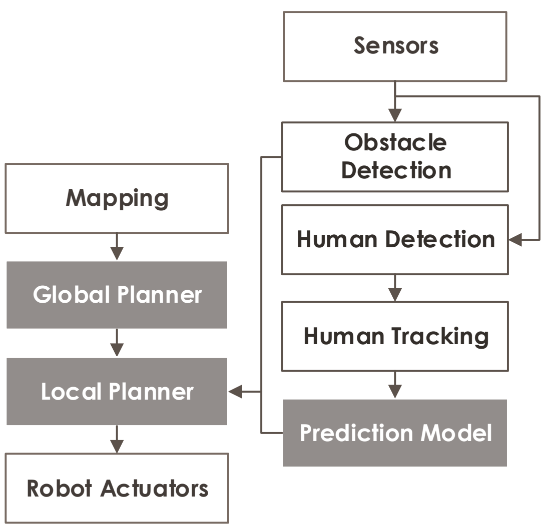

Figure 8: A Global Planner with Predictive Local Planner navigation framework

In the prior mentioned navigation frameworks, obstacle detection is for collision-avoidance between robot and both non-human and human subjects. In this framework, human detection is separated from obstacle detection and this enables the robot to behave differently when encountered a human or a non-human obstacle. Some literatures [24, 26] proposed to assign a Gaussian cost function to human to avoid personal space intrusion. Hence, the robot trajectory often keeps a further distance between human than that of other obstacles. This method of keeping distance often uses a linear human prediction model and thus it is the easiest way to provide acceptable level of comfort to pedestrians. However, using a linear model is susceptible to the “freezing robot problem” (FRP) [57], where the robot decides that stop moving is the safest choice instead of planning another path. Hence, instead of a linear model, a Social Force Model (SFM) [43] is used to improve the navigation framework. A stream of publications [23, 35, 44-47] used SFM prediction model to produce an effective navigation amidst pedestrians. Some researchers [13, 25, 28, 41, 54] take effort to model the crowd motion and expert demonstrations using machine learning and transfer the human behavior model to the local planner. With the predictive model, local planner is able to plan a better collision-free path based on the trajectories of human that are likely to follow in future instead of reactively avoiding them. Pérez-Higueras et al. [13] utilized this navigation framework to enable their service robot to move through populated public premises. Talebpour et al. [24] designed a domestic robot using this navigation framework to produce socially acceptable robot movement. Other literatures [33, 41, 46, 47] implemented this framework to predict and reason human motion for better obstacle avoidance and to display social-aware behavior.

D. Social-aware Global Planner with Predictive Local Planner

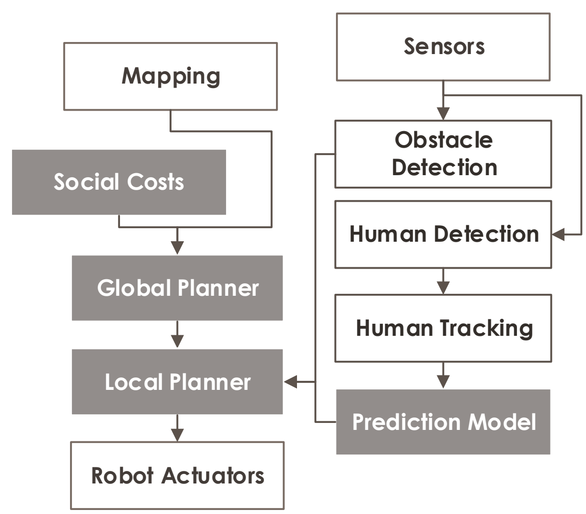

Figure 9: A Social-aware Global Planner with Local Planner navigation framework

To further improve the social-aware navigation framework, the global planner can be fed with social costs reviewed by Kruse et al [9], such as object padding, object occlusion and hidden zones. Social costs include object padding, object occlusion, hidden zones and many more. This results in a higher overall social awareness for the robot trajectory. With added social costs, a robot is able to plan path to avoid expected crowded areas, select a favorable human approaching direction and pick a right side on a pedestrian street even before starting navigation. This navigation framework reduces the burden of local planner to perform a social-aware motion, and thus resulting a more human-like robot behavior.

IV. ROBOT PLATFORM

Different types of robot platform can yield different performance in social-aware navigation due to its capability in tracking the planned path and performing human-robot collision avoidance.

A. Differential Drive

Differential drive platform enables a robot to change its orientation θ by varying the speed of the wheels at both sides (Figure 10), but limited to 2 degree of freedom (DOF). In an ideal environment, when both wheels are rotating at the same speed and same direction, the robot can either move forward or backward in a straight line. The robot can make a turn by commanding one of the wheel to move at different speed. When both wheels turn at the same speed but different directions, the robot will rotate in place. The instantaneous robot orientation θ(t) at t time is given by Equation (12):

$$\theta(t) = (V_R - V_L)t / b + \theta_o \tag{12}$$

where V_R and V_L are the velocities of right wheel and left wheel respectively, b is the distance between 2 wheels and θ_o is the initial robot orientation. Due to the simple kinematics of this platform, various robot prototypes [23-26, 35, 36, 39, 41, 44, 55] in social-aware navigation researches utilize differential steering platform to for path planning in human environment. However, path planner has to incorporate the complex non-holonomic constraints [58] of the differential drive platform.

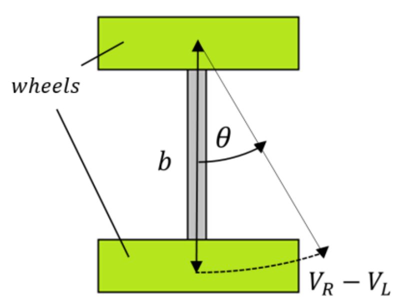

Figure 10: A differential drive platform can change its orientation θ by varying the speed of the wheels (V_R and V_L). [59]

B. Omnidirectional Drive

Omnidirectional drive is able to overcome the non-holonomic constraints of differential drive by employing special kind of wheel such as the Omni wheel or Mecanum wheel [60]. The free rotating rollers around the periphery of the wheel (Figure 11) enables omnidirectional drive platform to move laterally and rotates while moving in any direction in a planar space. The kinematics equation of the 3-wheeled omnidirectional drive platform with δ=60° can be written in the following matrix form:

$$\begin{bmatrix} V_x \\ V_y \\ \omega_p \end{bmatrix} = \begin{bmatrix} -1/2 & -1/2 & 1 \\ \sqrt{3}/2 & -\sqrt{3}/2 & 0 \\ 1/R & 1/R & 1/R \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \tag{13}$$

where V_x and V_y are the velocity components of robot platform in the reference frame X and Y, ω_p is the rate of rotation about pivot axis, R is the distance from the pivot axis to the wheel and V_1, V_2 and V_3 are the tangential velocities the wheels. By varying the velocity of the wheels, this robot platform is able to change its rotational and linear velocities at the same time.

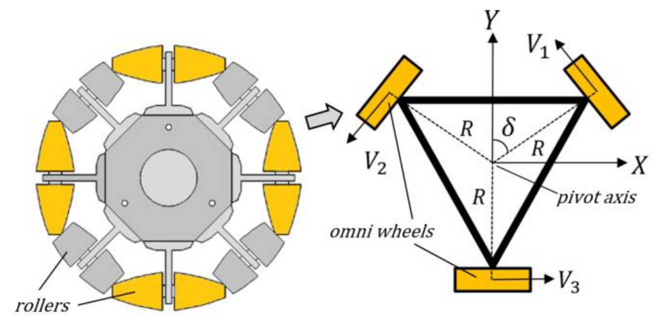

Figure 11: A 3-wheeled omnidirectional drive platform [61]

Omnidirectional drive platform can better perform obstacle avoidance due to its holonomic characteristic. Hence, several social-aware navigation studies used this platform for their experimental robots [4, 12, 29, 42]. Table 4 summarized the robot platforms used by selected literatures.

Table 4
Robot Platforms Used by Selected Literatures

Robot Platform	Literatures	Advantages	Disadvantages
Differential drive	[13, 23-26, 35, 36, 39, 41, 44, 55]	Simple platform construction	Suffers from non-holonomic constraints, poor obstacle avoidance capability
Omni-directional drive	[4, 12, 29, 42]	Flexible movement, good obstacle avoidance capability	Complex platform construction

V. DISCUSSION

Various navigation components and frameworks have been introduced in the previous sections and selecting the right ones can improve the overall performance of the robot social-aware navigation. Hence, this section aims to provide a comparative discussion on each methods, in the context of service robot. For global planner, A* algorithm is the most suggested planner to be used due to its easy implementation and is able to yield a shortest path. Algorithms for dynamic environments such as D*, LPA* and E* are complex to be implemented and furthermore, dynamic factors can be easily encountered using local planners. RRT that uses a random sampling technique to explore the search space might find a solution faster than A*, however it does not result in a shortest path. Solution path provided by RRT is also poor in smoothness. However, when the robot is required to navigate in a very confined space and to reach its goal in a specific orientation, A* algorithm can fail to plan a feasible path for robot with differential drive platform since it does not consider the kinematics constraints of the platform. Solutions to this can be either using an omnidirectional drive platform or use RRT that is able to incorporate the constraints into its planning. Both A* and RRT requires greater computational time as the problem domain scales up, but this problem can be solved using temporal planning methods reviewed by Kruse et al. [9].

For local planner, DWA that accounts for robot kinematics and dynamical constraints is suggested to be used to ensure the robot plans achievable paths. Local planner alone is insufficient for human-robot collision avoidance, where a prediction model is needed. For simplicity, local planner ORCA with linear prediction model can be considered because it can be used without the usage of sophisticated crowd motion model. The hardware setups for ORCA requires only minimal on-board sensors mounted the robot. Thus implementing ORCA is relatively simple as compared to other methods that require several over-head sensors to monitor the pedestrian movements. ORCA is suitable for service robot navigation in large environments such as shopping malls and event halls where external sensors are difficult to be installed. For smaller indoor environments such as cafeterias, restaurants and health care centers, external sensors such as depth camera and Laser Range Finder (LRF) can be installed to model and predict the crowd motion. Some literatures [53, 55] utilized this kind of hardware setting to replicate human motion and produce effective social-aware navigation. However, it is important to realize that the implementation is very complex.

SFM and modelling based on machine learning techniques can outperform a linear model, putting aside the complexity. SFM has the advantage of presenting the relationship between pedestrians, robot and other obstacles in discrete components, which can be adjusted separately. However, a full detail of an environment must be known beforehand in order for SFM perform optimally. Hence, in an unknown or partially known environment, one can consider using the machine learning methods which can adapt to different environments over time, but require more sensors to extract crowd features as compare to other methods.

Deciding which navigation framework to be used is always based on the components (global planner, local planner and prediction model) available. The most complete social-aware navigation framework is Framework 4 mentioned in the previous section. Framework 3 and framework 4 are difference by an added social cost component for global planner. The latter has advantage in certain specific occasion such as approaching human, else, both frameworks are able to provide a highly social acceptable navigation in dynamic human environments. However, both frameworks are almost impossible to be implemented without a complex prediction model and external sensor setups. To achieve the minimum requirement of a decent social-aware navigation, Framework 2 can be considered where DWA as the local planner, which take account of the robot dynamic constraints. Framework 2 can also be easily modified with a social-cost added global planner to enhance the social capability of the service robot. Classical Framework 1 are not encouraged to be considered in service robot implementation since it lacks of a local planner for collision avoidance which is not suitable to handle the stochastic human behavior. Table 5 shows a summary of literatures that utilized different frameworks for social-aware navigation.

Selection of robot platform can also impact the effectiveness of the selected navigation framework. Omnidirectional drive platform is preferable to differential drive due to its holonomic characteristic that can perform better human-robot collision avoidance.

Table 5
Different Social-Aware Navigation Frameworks Used in Literatures

Frameworks	Literatures	Advantages	Disadvantages
Sole Global Planner	[12]	Simple implementation for overall-path-focused applications	Poor obstacle avoidance
Global Planner with Local Planner	[36]	Accounts for performance of overall-path and obstacle avoidance	Poor obstacle motion prediction
Global Planner with Predictive Local Planner	[13, 24, 33, 41, 46, 47]	Good obstacle motion prediction using defined or learnt model	Social-aware aspects depends solely on local planner
Human-aware Global Planner with Predictive Local Planner	-	Able to reduce the burden of local planner in social-aware motion using global planner	Sophisticated implementation

VI. CONCLUSION

Navigation in dynamic human environment is a task more than just seeking for the shortest collision-free path between two locations. It is a task to maintain the human comfort level, perform human-like motion, respect social conventions and at the same time to accomplish a given goal. Existing literatures provide various methods to cater for different social scenarios, with each method having its merits and caveats. Hence, it often leads to confusion in selecting methods for implementation. This review thereby highlights four navigation frameworks including: sole global planner (Framework 1), global planner with local planner (Framework 2), global planner with predictive local planner (Framework 3), and social-aware global planner with predictive local planner (Framework 4), to provide an insight for researchers on which navigation frameworks and components to be used for service robot implementation. It is suggested that A* algorithm to be used as global planner. Local planner however has to be selected based on the framework used. Hence, if Framework 2 is chosen, DWA is advisable to be the local planner. For Framework 3 or Framework 4, ORCA local can be selected for easier hardware implementation that uses a linear prediction model. To obtain the best performance for navigation in dynamic human environment, SFM or IRL as the prediction model can be considered. For robot platform, omnidirectional drive is preferred for a better human-robot collision avoidance. A unified navigation framework such as Framework 4 can be further studied and implemented in order to better realize the pros and cons. Future researches might also consider comparing and evaluating the performance of different human motion modelling techniques to reduce ambiguity during implementation of social-aware navigation.

ACKNOWLEDGMENT

The authors are grateful to Universiti Teknologi Malaysia for providing facilities to conduct this survey and this work is supported by the Universiti Teknologi Malaysia Research Grant (00M81). Besides, the authors also thank to DF Automation & Robotics Sdn Bhd for providing useful robotic facilities, information and insights.

REFERENCES

[1] E. Zalama, J. G. García-Bermejo, S. Marcos, S. Domínguez, R. Feliz, R. Pinillos, et al., "Sacarino, a Service Robot in a Hotel Environment," in *Proc. ROBOT2013: First Iberian Robotics Conference: Advances in Robotics*, Madrid, Spain, 2013, pp. 3-14.

[2] E. Broadbent, R. Stafford, and B. MacDonald, "Acceptance of Healthcare Robots for the Older Population: Review and Future Directions," *International Journal of Social Robotics*, vol. 1, pp. 319-330, 2009.

[3] L. Marchionni, J. Pages, J. Adell, J. R. Capriles, and H. Tomé, "REEM Service Robot: How May I Help You?," in *Proc. Natural and Artificial Models in Computation and Biology*, Mallorca, Spain, 2013, pp. 121-130.

[4] J. Lafaye, D. Gouaillier, and P. B. Wieber, "Linear model predictive control of the locomotion of Pepper, a humanoid robot with omnidirectional wheels," in *Proc. IEEE-RAS International Conference on Humanoid Robots*, Madrid, Spain, 2014, pp. 336-341.

[5] Ferro, Francesco, and Luca Marchionni. "REEM: A Humanoid Service Robot," in *Proc. ROBOT2013: First Iberian Robotics Conference*, Madrid, Spain, 2013, pp. 521-525.

[6] R. Bogue, "Sensors for robotic perception. Part one: human interaction and intentions," Industrial Robot: An International Journal, vol. 42, pp. 386-391, 2015.

[7] Edward, Hall. *The hidden dimension*. Anchor Books, 1990.

[8] A. Kendon, "Spacing and orientation in co-present interaction," *Development of Multimodal Interfaces: Active Listening and Synchrony*, pp. 1-15, 2010.

[9] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726-1743, 2013.

[10] Rios-Martinez, A. Spalanzani, and C. Laugier, "From Proxemics Theory to Socially-Aware Navigation: A Survey," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 137-153, 2014.

[11] S. M. LaValle and James J. Kuffner. "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378-400, 2001.

[12] M. C. Shrestha, Y. Nohisa, A. Schmitz, S. Hayakawa, E. Uno, Y. Yokoyama, et al., "Using contact-based inducement for efficient navigation in a congested environment," in *Proc. 24th IEEE International Symposium on Robot and Human Interactive Communication*, Kobe, Japan, 2015, pp. 456-461.

[13] N. Pérez-Higueras, R. Ramón-Vigo, F. Caballero, and L. Merino, "Robot local navigation with learned social cost functions," in *Proc. 11th International Conference on Informatics in Control, Automation and Robotics*, Vienna, Austria, 2014, pp. 618-625.

[14] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, *vol.* 30, no. 7, pp. 846-894, 2011.

[15] C. b. Moon and W. Chung, "Kinodynamic planner dual-tree RRT (DT-RRT) for two-wheeled mobile robots using the rapidly exploring random tree," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 2, pp. 1080-1090, 2015.

[16] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, 1968.

[17] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," *Natural Computing: an international journal*, vol. 8, no. 2, pp. 239-287, 2009.

[18] T. J. Misa and P. L. Frana, "An interview with edsger w. dijkstra," *Communications of the ACM*, vol. 53, no. 8, pp. 41-47, 2010.

[19] A. Stentz, *Optimal and efficient path planning for unknown and dynamic environments*, Carnegie Mellon Robotics Institute Technical Report CMU-RI-TR-93-20. 1993.

[20] A. Stentz, "The Focussed D* Algorithm for Real-Time Replanning," in *International Joint Conference on Artificial Intelligence*, Montreal, Quebec, 1995, pp. 1652-1659.

[21] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 354-363, 2005.

[22] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning A∗," *Artificial Intelligence*, vol. 155, no. 1, pp. 93-146, 2004.

[23] G. Ferrer, A. Garrell, and A. Sanfeliu, "Social-aware robot navigation in urban environments," in *Proc. European Conference on Mobile Robots*, Barcelona, Catalonia, Spain, 2013, pp. 331-336.

[24] Z. Talebpour, I. Navarro, and A. Martinoli, "On-Board Human-Aware Navigation for Indoor Resource-Constrained Robots: A Case-Study with the Ranger," in *Proc. IEEE/SICE International Symposium on System Integration*, Meijo University, Japan, 2015, pp. 63-68.

[25] M. Kuderer, H. Kretzschmar, and W. Burgard, "Teaching mobile robots to cooperatively navigate in populated environments," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo Big Sight, Japan, 2013, pp. 3138-3143.

[26] M. Kollmitz, K. Hsiao, J. Gaa, and W. Burgard, "Time dependent planning on a layered social cost map for human-aware robot navigation," in *Proc. European Conference on Mobile Robots*, Lincoln, UK, 2015, pp. 1-6.

[27] H. Li, O. A. I. Ramirez, and M. Chetouani, "Potential human reaction aware mobile robot motion planner: Potential cost minimization framework," in *Proc. RO-MAN: The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Edinburgh, Scotland, UK, 2014, pp. 441-448.

[28] D. Vasquez, B. Okal, and K. O. Arras, "Inverse reinforcement learning algorithms and features for robot navigation in crowds: an experimental comparison," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, Illinois, USA, 2014, pp. 1341-1346.

[29] J. Biswas and M. M. Veloso, "Localization and navigation of the cobots over long-term deployments," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1679-1694, 2013.

[30] C. Weinrich, M. Volkhardt, E. Einhorn, and H. M. Gross, "Prediction of human collision avoidance behavior by lifelong learning for socially compliant robot navigation," in *Proc. IEEE International Conference on Robotics and Automation*, Karisruhe, Germany, 2013, pp. 376-381.

[31] S. Iizuka, T. Nakamura, and S. Suzuki, "Robot Navigation in dynamic environment for an indoor human monitoring," in *Proc. RO-MAN: The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Edinburgh, Scotland, UK, 2014, pp. 698-703.

[32] J.-C. Latombe. *Robot motion planning*. Springer Science & Business Media, 2012.

[33] S. Oli, B. L. Esperance, and K. Gupta, "Human Motion Behaviour Aware Planner (HMBAP) for path planning in dynamic human environments," in *Proc. 16th International Conference on Advanced Robotics*, Universidad de la República, Montevideo, Uruguay, 2013, pp. 1-7.

[34] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23-33, 1997.

[35] M. Shiomi, F. Zanlungo, K. Hayashi, and T. Kanda, "Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model," *International Journal of Social Robotics*, vol. 6, no. 3, pp. 443-455, 2014.

[36] F. Xia, L. Tyoan, Z. Yang, I. Uzoije, G. Zhang, and P. A. Vela, "Human-aware mobile robot exploration and motion planner," In *Proc. SoutheastCon*, Fort Lauderdale, Florida, USA, 2015, pp. 1-4.

[37] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760-772, 1998.

[38] J. v. d. Berg, L. Ming, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE International Conference on Robotics and Automation*, Pasadena, California, USA, 2008, pp. 1928-1935.

[39] Z. Dongxiang, X. Zongjun, L. Pengfei, Y. Jiahui, and C. Xiaoping, "Real-time navigation in dynamic human environments using optimal reciprocal collision avoidance," in *Proc. IEEE International Conference on Mechatronics and Automation*, Beijing, China, 2015, pp. 2232-2237.
[40] J. Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Proc. The 14th International Symposium of Robotics Research*, Astoria Hotel, Lucerne, Switzerland, 2011, pp. 3-19.
[41] B. Kim and J. Pineau, "Socially adaptive path planning in human environments using inverse reinforcement learning," *International Journal of Social Robotics*, vol. 8, no. 1, 51-66, 2016.
[42] I. Nishitani, T. Matsumura, M. Ozawa, A. Yorozu, and M. Takahashi, "Human-centered X–Y–T space path planning for mobile robot in dynamic environments," *Robotics and Autonomous Systems*, vol. 66 no. C, pp. 18-26, 2015.
[43] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, pp. 4282, 1995.
[44] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo Big Sight, Tokyo, Japan, 2013, pp. 1688-1694.
[45] P. Ratsamee, Y. Mae, K. Ohara, T. Takubo, and T. Arai, "Human–robot collision avoidance using a modified social force model with body pose and face orientation, " *International Journal of Humanoid Robotics*, vol. 10, no. 1, pp. 1350008, 2013.
[46] A. Colombo, D. Fontanelli, A. Legay, L. Palopoli, and S. Sedwards, "Motion planning in crowds using statistical model checking to enhance the social force model," in *Proc. IEEE 52nd Annual Conference on Decision and Control*, Palazzo dei Congressi, Florence, Italy, 2013, pp. 3602-3608.
[47] A. Colombo, D. Fontanelli, D. Gandhi, A. D. Angeli, L. Palopoli, S. Sedwards, et al., "Behavioural templates improve robot motion planning with social force model in human environments," in *Proc. IEEE 18th Conference on Emerging Technologies & Factory Automation*, Cagliari, Italy, 2013, pp. 1-6.
[48] S. Russell, "Learning agents for uncertain environments," in *Proc. 11th annual conference on Computational Learning Theory*, University of Wisconsin, Madison, Wisconsin, USA, 1998, pp. 101-103.
[49] R. Bellman, "A Markovian Decision Process," *Indiana Univ. Math. J.*, vol. 6, no. 4, pp. 679-684, 1957.
[50] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proc. 21st International Conference on Machine learning*, Banff, Alberta, Canada, 2004, pp. 1.
[51] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum Entropy Inverse Reinforcement Learning," in *Proc. 23rd AAAI Conference on Artificial Intelligence*, Chicago, Illinois, USA, 2008, pp. 1433-1438.
[52] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: the case for cooperation," in *Proc. IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013, pp. 2153-2160.
[53] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335-356, 2015.
[54] H. Kretzschmar, M. Kuderer, and W. Burgard, "Learning to predict trajectories of cooperatively navigating agents," in *Proc. IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014, pp. 4015-4020.
[55] H. Hisahara, Y. Ishii, M. Ota, T. Ogitsu, H. Takemura, and H. Mizoguchi, "Human Avoidance Function for Robotic Vacuum Cleaner through Use of Environmental Sensors: Roomba® Making Way for Humans," in *Proc. 5th International Conference on Intelligent Systems, Modelling and Simulation*, Langkawi, Malaysia, 2014, pp. 64-67.
[56] J. Guzzi, A. Giusti, L. M. Gambardella, G. Theraulaz, and G. A. D. Caro, "Human-friendly robot navigation in dynamic environments," in *Proc. IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013, pp. 423-430.
[57] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 797-803.
[58] J. Barraquand and J.-C. Latombe, "Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles," *Algorithmica*, vol. 10, no. 2, pp. 121-155, 1993.
[59] G. Lucas, "A tutorial and elementary trajectory model for the differential steering system of robot wheel actuators," *The Rossum Project*, 2001.
[60] O. Diegel, A. Badve, G. Bright, J. Potgieter, and S. Tlale, "Improved mecanum wheel design for omni-directional robots," in *Proc. Australasian Conference on Robotics and Automation*, Auckland, 2002, pp. 117-121.
[61] A. S. Al-Ammri and I. Ahmed, "Control of Omni-Directional Mobile Robot Motion," *Al-Khwarizmi Engineering Journal*, vol. 6, no. 4, pp. 1-9, 2010.
[62] W. La, T. Koogle, D. Jaffe, and L. Leifer, "Microcomputer-controlled Omnidirectional Mechanism for Wheelchairs," *IEEE Transactions On Biomedical Engineering*, vol. 28, no. 8, pp. 594-594, 1981.