EU Horizon 2020 Research & Innovation Program
Advanced Robot Capabilities & Take-Up
ICT-25-2016-2017
Grant Agreement No. 779942

Safe Robot Navigation in Dense Crowds

http://www.crowdbot.org

Technical Report

D 1.1: Specification of Scenarios Requirements

Work Package 1 (WP 1)
Scenarios Co-Design & Evaluation

Task Lead: INRIA, France
WP Lead: University College London, UK

Table of Contents

Executive Summary

Scenarios are descriptors that portray use cases and operational procedures of mobile robots in human crowd environments such as hospitals, shopping malls, train stations and other public or private venues. Nowadays we are witnessing the presence of robots in both public and private places but their efficacy and technological features are rather modest due to their limited mobility and interaction with humans. The main focus of the Crowdbot Project is to demonstrate safe and efficient mobile robot navigation in a dense crowded human environment. This report details various navigation scenarios we plan to test and validate as part of the overall project goal of research, innovation, ethics and feasibility study for technology transfer in mobile robotics.

In **Sections 1** and **2** we provide a thorough coverage of our proposed definitions and relationships among the terms *general/social*, *operational* and *test* scenarios. Full description of various navigation test scenarios that the team plans to test and evaluate is provided in **Section 4**.

Closely related to scenarios are test events and system-level requirements to evaluate the outcomes (both success and failure) of such tests. Programmatic aspects of requirements development and test execution are covered in the latter part of **Section 2**.

In Crowdbot an internal Test & Evaluation (T&E) team is given the authority to prepare a System-Level Test Plan (STP) and its associated requirements list. An STP is prepared a month before the commencement of a test event. Two test events (the 1st in late 2019/early 2020 and the 2nd in early-to-mid 2021) are planned over the 42-month project span. A requirements list pertaining to all proposed Crowdbot navigation scenarios is presented in **Section 5**.

The T&E team is also responsible for evaluating test data and publication of a test report after the conclusion of each test event. The test report is shared with both internal teams in charge of Technology Development (TD), Robot System Integration (SI) and Design & Quality Control (QC) and external stakeholders. Based on test evaluation and recommendation from the T&E team and external stakeholders, the remaining internal teams (TD, SI and QC) devise technology enhancements to improve robotic navigation. Specific details are provided in **Sections 5** and **6**.

For validation and verification of test scenarios and robotic performance outcomes by external stakeholders, the team plans to solicit advice and feedback from technical experts as well as potential user communities via interview sessions and related engagements, coordinated meetings and joint publications. This topic is covered in **Section 6**.

A unique feature of the Crowdbot Project is its utilization and integration of navigational technologies onto three different types of robots. The team has already provided technical descriptions and specifications of system components and onboard sensor suite of our robots in already released reports *D2.1: Sensor Specifications* and *D5.1: System Architecture*. In **Section 3** we provide supplementary material of our robots with relevance to scenarios, requirements and test events: namely, details about each robot's physical frame structure, locomotion profile and human-machine interaction & communication options.

Finally, this report (D1.1) is the first in a series of two scenarios and requirements reports we plan to prepare and submit as official deliverables. The follow-on report D1.3 (Scenarios & Requirements Update) will be prepared and delivered a month after the release of 1st Test Report.

1. Introduction

The title of this report "Specification of Scenarios Requirements" is somewhat long-winded but in essence, this is a *requirements* document. Immediately, this raises two relevant questions:

Q1: What are scenarios?

Q2: Who is preparing scenario requirements and evaluating test outcomes?

The answer to **Q2** will be provided in Section 2 and further elaborated in Sections 5 and 6. First, we address **Q1**.

Since the main focus of the Crowdbot project is safe and intelligent navigation of a robot in a dense human crowd environment, the term *scenario* refers to a space-time event in which a mobile robot moves from an initial location (say, Point A) at stop watch time t_0 (time zero) to its final position (say, Point B) by time t_{fin} (finish time). To factor in the human crowd effect to a scenario, we must specify the physical environment (e.g. building structures, furniture, roads, trees, etc.) to bound the available space for movement and also characterize the behavior and motion profile of humans that the robot is likely to encounter during its traversal from Point A to Point B. A robotic navigation event can be summarized more precisely in terms of an *Operational Scenario*.

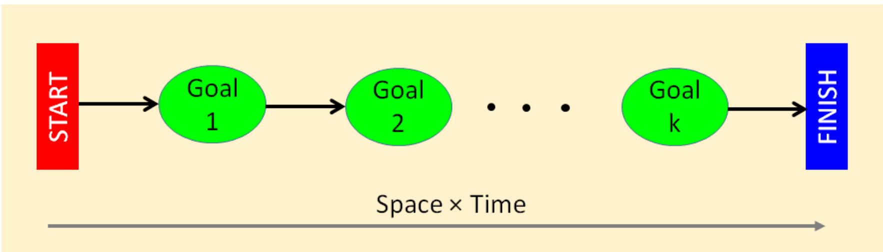


Figure 1.1: Description of a Navigation Scenario in terms of Space-Time Markers

As shown in Figure 1.1, the START marker is associated with start position (Point A) at time t_0 —denoted as (A, t_0) — and the FINISH marker as (B, t_{fin}). Note that several intermediate steps (k steps in Figure 1.1) are involved before reaching the finish line. In the robotics community each step is known as a goal and the position of the robot after the completion of a goal is called its goal location. In general, the overall START-to-FINISH operational scenario is broken into smaller goals because in each intermediate step, a unique task-specific operation must be carried out to reach a designated goal location. Mathematically, the goal location for Goal m can be denoted as L_m and its corresponding finish time as t_m. Hence, every START-FINISH operational scenario is a succession of goal-oriented, task-specific robotic operations, all executed in the correct order to reach the specified finish point.

START: (A, t_0) → G_1: (L_1, t_1) → G_2: (L_2, t_2) → • • • → G_m: (L_k, t_k) → FINISH: (B, t_{fin})

We now provide an example to illustrate and elaborate on a real-world scenario. Figure 1.2 shows a tele-operated robot by Starship Technologies [1] adapted for delivery of Domino's pizza. An employee loads the pizza into the robot's cargo bay and is later retrieved by a customer.

Figure 1.2: Domino's Pizza Delivery Robot by Starship Technologies [1]
Left: An employee of Domino's loading the pizza in the cargo bay; Source: [2]
Right: A woman opening the top hatch to access the cargo area; Source: [3]

It is important to note that this pizza delivery scenario falls under the category of a real-world *Social or General Scenario*. From an operational perspective, robotic pizza delivery is simply a navigational operation of a service robot carrying a package from Point A to Point B with possibly an upper bound on the total delivery time (to maintain crispiness of a freshly baked pizza). Strictly speaking, an operational scenario is task-specific and goal-oriented whereas a social or general scenario almost always has a story to tell (i.e. use case) in some social context. Other plausible examples of general/social scenarios for robotic navigation are:

- Hospital aid robot assisting patients, possibly guiding from one room/area to another
- Railway station guide providing directions to departure platforms and exits to city streets
- A service courier robot delivering mail or packages to offices and residential homes

Figure 1.3: Two very different Physical Environments for a Service Robot
Left: Patient waiting area and hallway of a typical hospital floor; Source: [4]
Right: Heavy foot traffic at Paris Gare du Nord train station; Source: [5]

It is obvious from above examples that general scenarios are not insightful in revealing underlying operational procedures and robotic technologies needed for a successful outcome. As the name suggests, they are "general" in describing a navigational event. In fact, from the above examples, we can see that the role of a robot guide in a hospital setting or at the train station appears similar even though the former takes place in an indoor cluttered office space (Figure 1.3, left) while the latter is a crowded, mostly open-space environment with disorderly human traffic patterns (Figure

1.3, right). Hence from an operational viewpoint, they are very different. On the other hand, the service courier and the pizza delivery robots are almost identical in their operational roles, the physical environment and human behavior profiles they each encounter. Nevertheless, general scenarios are still useful when communicating with the following target audiences:

- *General Public*: When a layperson hears that the Crowdbot team is developing a robot that moves and co-exists among humans, the immediate response is: "What's the purpose?" or "Why is it behaving this way?" When a robot's behavior is portrayed by telling a story in an acceptable social context via a general scenario, members of the general public are more receptive and accepting, and thus are more likely to tolerate or participate in actual tests.
- *Potential Customers*: How robots are used in a socially setting in the near and far future is anybody's guess. Promoting a robot for a specific task is also outside the scope of the Crowdbot project. However, the team cannot interact and explore potential use cases with a potential customer unless we have a plausible story to tell and the general scenario description is a meaningful and effective ice-breaker.
- *Stakeholders*: Stakeholders are individuals, groups and other entities that have a personal, professional or institutional interest in the Crowdbot project. A good example of stakeholders' concern is the safety and ethical implications of robots co-existing in human-only environments. In an indoor office setting, local, state and national level safety laws (c.f. OSHA Laws and Regulations [26]) prohibit the placement of bicycles nor the roaming of animals nor crying babies. Using general scenarios the Crowdbot team communicates with external stakeholders in resolving potential ethical issues and drafting new standards proposals for safe operation of robots in cluttered and crowded environments.

Further details of general scenarios and their ethical implications are detailed in our companion report, *D6.1: Overview of Risks when Using Robots in Crowds* [6].

Now that we have provided clear examples delineating the differences between an operational from a general/social scenario, we return to the original pizza delivery scenario to elaborate further on the required operational procedures. One possible navigation scenario of the Domino's robot is as follows: A pizza is delivered from a local Domino's bakery store (START marker) to a customer staying at a nearby hotel (FINISH marker). Other intermediate steps are:

1. *Transitional* (stop-go-merge) Operation: The robot starts from the store front of a local Domini's pizza store. It leaves the store premise and merges onto the sidewalk with human pedestrian flow and heads in the direction of the customer (see Figure 1.4, left).
2. *Flow with Traffic* Operation: The robot traverses across city streets and pedestrian sidewalks. In some cases, it follows the human crowd. In another case, it negotiates with the cross traffic (i.e. vehicles) when it has to cross a street (see Figure 1.4, middle).
3. *Mapping, Localization & Path Planning* Operation: The robot has knowledge of a path it has to take to arrive at the hotel entrance where the customer is staying. The finish position can be the hotel room of the customer (at a higher floor level) or merely the reception desk (ground level) or the front entrance area of the hotel.
4. *Safety & Robustness* Operation: During the entire journey the robot must be mindful of potential hazards: collision with pedestrians; collision with man-made and natural objects;

getting stuck in a pothole or train track (see Figure 1.4, right); falling down via the bank of a sidewalk; getting lost and unable to arrive at its next goal location, etc.

Figure 1.4: Task-specific Operations of a Pizza Delivery Service Robot
Left: Robot traversing along with pedestrians on a paved walking path; Source: [28]
Middle: Robot attempting to cross along the zebra against vehicular cross traffic; Source: [29]
Right: Robot maneuvering over city tram rail tracks and vehicular traffic; Source: [30]

The relationship between a general scenario and its corresponding task-specific robotic operations are shown in the top half of Figure 1.5. That is, every general scenario can be described in terms of an equivalent operational scenario which is a succession of task-specific and goal-oriented operations arranged in a specific order:

General Scenario → Operational Scenario = (List of Operations + Execution Order)

At each intermediate step, a navigation goal can be defined in terms of its task-specific operation. For example, for the Domino's robot, its initial stop-go-merge operation is equivalent to reaching Goal 1 from its START position.

In general the order of execution (of operations) is important. For example, the operation of a robot exiting via a door (say, from a furnished and cluttered office room) and walking along a corridor requires a different set of technologies from the case where a robot walks along the corridor first and then finds an exit via a door to an office room.

As shown in the bottom half of Figure 1.5, each task-specific operation is associated with one or more technology profiles. A *Technology Profile* is a unique operational procedure that lays out all the required technological components as well as their integration for a particular robot such that the desired goal is met.

Task-Specific Operation → Collection of Technology Profiles (robot specific)

In essence, a technology profile is a recipe description for cooking an operational dish. Since the ingredients in the recipe are derived from the robot itself (its sensors, its processors, its software modules, its body frame and other hardware components), a technology profile is relevant and meaningful only when paired with a specific robot and its technical specifications. Therefore, the same operation for robot navigation (say, entering/exiting a door) may require a different technology profile for a humanoid robot from that of a smart wheelchair.

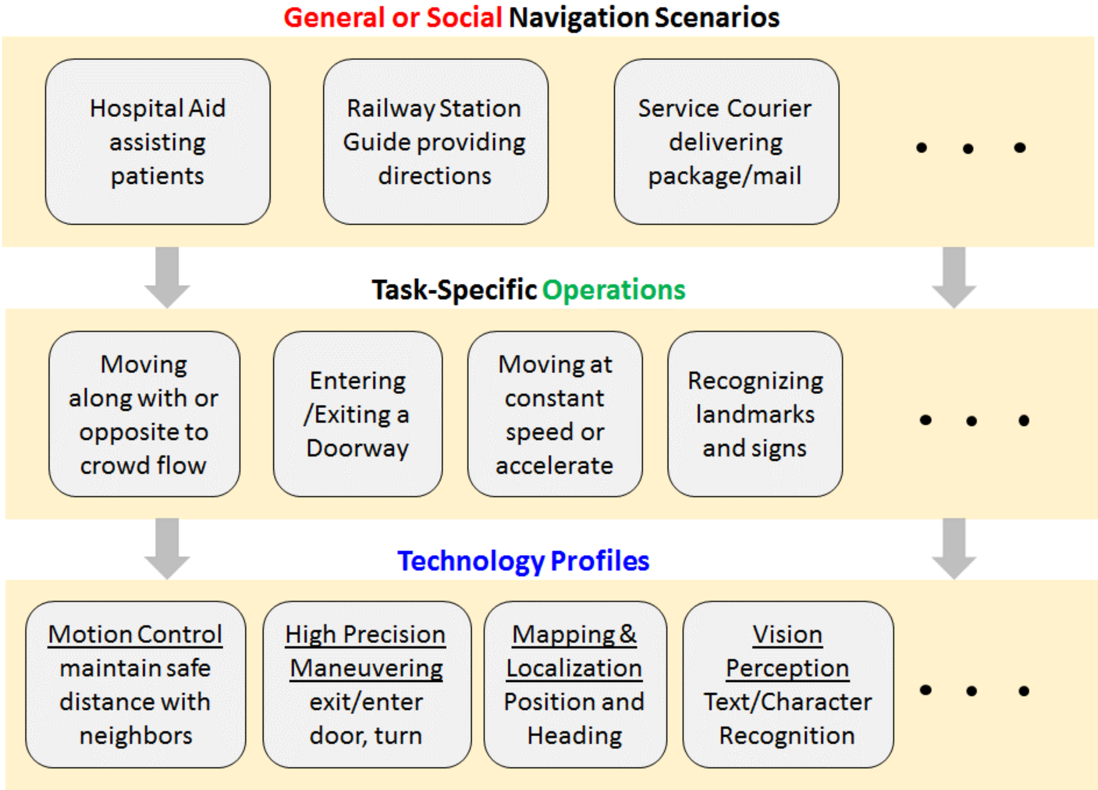


Figure 1.5: Categorization of a Different Navigation Scenarios

Note that this report is a requirements document and thus we do not stipulate or promote a certain type of sensing or processing technology, or robotic hardware design to fulfill a particular operational objective. We, however, do take into account specific hardware limitations that exist in some types of robotic platforms and are mindful that certain operational tasks cannot be handled by such robots. In this case, this operational task is excluded from its originating operational and general/social scenarios. We again use the pizza delivery robot to give a final example.

In the Domino's pizza delivery scenario, we stated earlier that the finish location can be

1. The customer's hotel room (at a higher floor level)
2. The reception desk at ground level
3. The hotel entrance area (possibly outside the main entrance door)

It is apparent that the operational complexity for each option is different from the other two. Equivalently, different technology profiles are needed to fulfill the goal in each option. Option 1 is the most difficult since it requires the robot to enter through the main door, operate an elevator to reach the desired floor level and navigate in the corridor and accurately locate the final position —the customer's hotel room. In contrast, option 3 is the simplest. The customer or a hotel staff must retrieve the pizza from the robot just outside the main entrance. Figure 1.6 shows three different types of robots (ETH Zurich's ANYmal [7], Savioke room service robot [8] and a TUG robot [10] for luggage delivery), all capable of using an elevator. The ANYmal robot is equipped with a limb to press the elevator's buttons while the other two are embedded with radio devices to wirelessly communicate with the elevator operator to request elevator use privileges.

Figure 1.6: Three different types of robots with technology profile to use an elevator

Left: ANYmal robot with dexterity and precision to press the elevator button; Source: [7]
Middle: Wally room-service Robot using the elevator via its radio communication module; Source: [8, 9]
Right: TUG luggage delivery robot maneuvering its way to the elevator; Source: [10, 11]

For a robot without either of these technology profiles, this type of operation is excluded from the available list. In Section 2.3 we outline the operational scope and technical capabilities of our robots. In short, none of the Crowdbot robots support technology profiles that require high-precision movement of body parts. Interaction with humans via head and joint movement, vision sensors, audio and visual cues and communication devices applies to the humanoid Pepper only.

2. Scenario Requirements Overview

In the previous section we provided ample detail and relevant examples to clarify the terms "operational" and "general/social" scenarios. Both terms are useful and serve different purposes in conveying the main objectives of the Crowdbot project. In this section we introduce another related term called the "test" scenario which is used to derive *requirements*. We will use mathematical notations in order to provide concise definitions of general, operational and test scenarios.

If we denote the list of all conceivable general/social scenarios as **W**:

$$\mathbf{W} = (W_1, W_2, W_3, \ldots)$$

and the equivalent operational scenarios as **V**:

$$\mathbf{V} = (V_1, V_2, V_3, \ldots)$$

Each operational scenario V can be described in terms of several individual goals bounded by START and FINISH markers. This is denoted as (without indexing)

$$V = [\text{START} \rightarrow G_1 \rightarrow G_2 \rightarrow \ldots G_k \rightarrow \text{FINISH}]$$

START, FINISH and Goals are space-time milestone posts. To transition from one post to its successor, a robot must complete a task-specific operation. To reach Goal *m*, it carries out Operation m (O_m). Of course, O_1 is the first operation initiated from the START marker. Likewise, the $(k+1)^{st}$ operation (O_{k+1}) is executed to reach the FINISH marker:

O_1: START $\rightarrow G_1$, O_m: $G_{m-1} \rightarrow$ G_*m*, O_{k+1}: $G_k \rightarrow$ FINISH

An equivalent representation of an operational scenario in terms of task-specific operations is:

$$V = [\, O_1 \rightarrow O_2 \rightarrow O_3 \rightarrow \ldots O_k \rightarrow O_{k+1} \,]$$

A key takeaway from above description is that the sets **V** and **W** for operational and general scenarios are very large, and possibly infinite in size whereas the set of robot operations $\{O_m\}$ is finite. Let's denote this set of operations as **D** (i.e., D_m for "do this operation *m*"):

$$\mathbf{D} = (D_1, D_2, \ldots, D_p)$$

Every operation O_m belongs to the set D of finite entries up to **p**. Furthermore, if we describe all possible technology profiles required by a particular robot to successfully complete a goal as

$$\mathbf{T} = (T_1, T_2, \ldots, T_n)$$

where T_m is a unique technology profile of type *m*, then **T** is also finite.

Referring to the elevator example in Section 1, we deduce that floor-to-floor navigation is an operational task (say, D_m) and its associated technology profiles are radio communication (say, T_j) or mechanical button press (say, T_k). Of course, each D_m is usually associated with more than one technology profile. In our example, for a robot to press an elevator button, other technology profiles such as high-precision vision to locate the button and high-precision motor control to move the limb/finger to the right position are required. The implication of these realizations to Crowdbot robots is two-fold:

1. If a certain robot is lacking a specific technology profile, then the operation that requires its execution will not be successful. Therefore, such operation must be excluded when testing and evaluating an operational scenario. Even better is to exclude operational scenarios that involve such an operation due to the fact that operational scenarios are operation-order dependent (see Section 1 for additional details).
2. Since each operation relies on the execution of one or more technology profiles in unison, test results suffer from compound effect; that is, when there is a test failure, it is difficult to determine the root cause or culprit that is responsible. For this reason, tests should be structured in a hierarchical fashion where each base test is used to evaluate and validate one or two technology profiles only.

We are now ready to provide a concise definition of *Test Scenario*. As shown in Figure 2.1, there is an equivalence mapping from a general to an operational scenario. However, an operational scenario test cannot be executed successfully using a particular robot due to its lack of required technology profiles. Furthermore, even if a robot supports such technology profiles, a test should be excluded if the cause of failure cannot be analyzed without ambiguity. Therefore, test scenarios are a subset of operational scenarios where we take into account

- Available technology profiles of the robot under test
- Other test constraints (physical environment, human participants, etc.)
- Clear and concise test objectives that lead to further enhancements to the robot

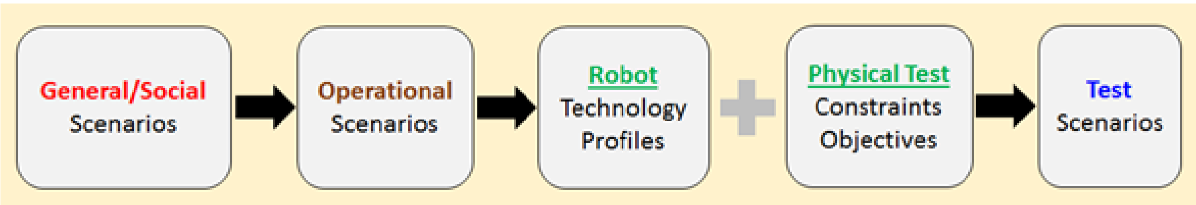


Figure 2.1: Relation between General and Test Scenarios

From this point forward, we will use the term "scenario" to refer to a test scenario. If there is potential for confusion, we will explicitly state whether a scenario is general, operational or test. Section 4 covers all aspects of (test) scenarios relevant to the Crowdbot project. In fact, there are only seven main scenarios, denoted as:

$$\mathbf{S} = (S_1, S_2, \ . \ . \ . \ , S_7)$$

Each main scenario can have a number of derivative or sub-scenarios. For test and evaluation purposes, there are requirements associated with each scenario. They are all listed and elaborated in section 5. The notational convention is as follows. All requirements associated with Scenario *m* are denoted as $\mathbf{R}_m$ and the complete set of all test requirements is **R**:

$$\mathbf{R}_m = (R\ m.1,\ R\ m.2,\ R\ m.3, \ldots), \qquad \mathbf{R} = (\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \ldots)$$

Before leaving this section, we have one more important topic to discuss. Thus far we have provided reasoning and justification (mostly due to technology constraints) for conducting well-controlled test scenarios instead of the more realistic operational or general scenarios in real-world environments such as a hospital floor or a railway station platform. There are also obvious logistical, financial and legal reasons that prohibit the team from conducting such real-world tests. Nevertheless, our tests and gained insights are meaningful and of value to stakeholders only if we can infer or deduce from test results their applicability to operational or general scenarios. In the commercial industry, this step is known as *requirement trace*. We noted above that a set of requirements $\mathbf{R}_m$ is associated with each test scenario S_m. In reverse, we can specify an operational or a general scenario in terms of its requirements from a user or customer point-of-view (as opposed to goals and operations for the technology developers). Using mathematical notation,

Requirements for V or W → R *i.j*, R *m.n*, R *p.q*, . . . (**Test Scenario Requirements**)

That is, we can trace all requirements from test scenarios that can be used to define an operational or a general scenario. Note, however, some requirements for V or W may not exist in the set **R**. In this case, they are known as *deficiencies* to be met by further technology enhancements by the Crowdbot technology team or carried over to future research teams.

2.1 **Requirements Authority**

In the beginning of Section 1 we raise the question **Q2**: *Who is preparing scenario requirements and evaluating test outcomes*? This section provides the answer. Unlike commercial product development contracts, the Crowdbot project is focused on research and innovation. We do not have an external customer that prepares a requirements document as part of contractual obligations; however, we are guided and overseen by our funding authority, the European Commission and are chartered to work closely with external stakeholders. Details of our interaction with stakeholders are discussed further in Section 6.3. Here we provide a brief summary.

For the Crowdbot project, the internal Test & Evaluation (T&E) team is in charge of requirements specification (including this report), interaction with relevant stakeholders, preparation of potential use cases, and foremost, evaluation of test data and delivering recommendations for further technology enhancements. For the purpose of this discussion, the Crowdbot team as a whole can be grouped into four sub-teams (see left side of Figure 2.2):

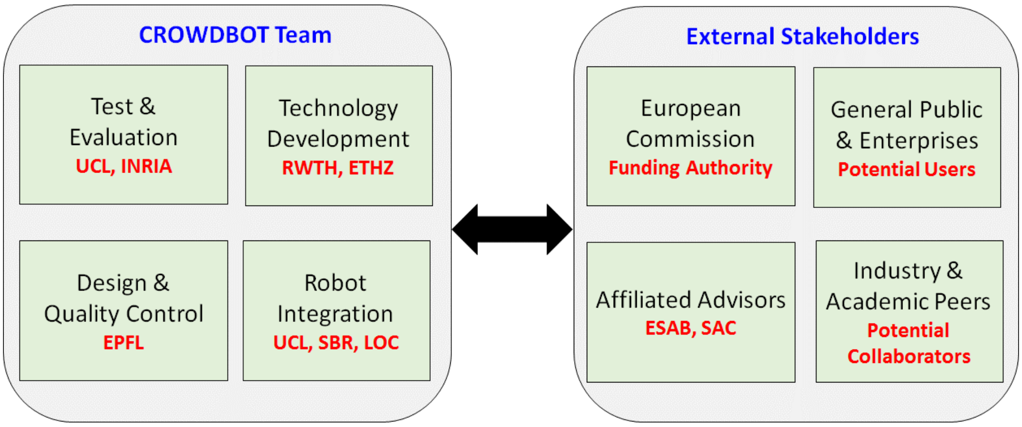


Figure 2.2: Crowdbot team composition and interaction with external stakeholders

ESAB: Ethics & Safety Advisory Board, **SAC**: Scientific Advisory Committee

The lead partners in each sub-team are highlighted in red. The Test & Evaluation team, which is responsible for this report as well as evaluation of test results, is led by UCL and INRIA. Two other partners RWTH and ETHZ are the main developers and enablers of key technologies in sensing, processing and navigation. They lead the Technology Development (TD) team. Integration of such technologies into specific robots are assigned to the owners of such robots: UCL (Smart Wheelchair), SBR (Humanoid Robot) and Locomotec (CuyBot), collectively known as System Integration (SI). The T&E team is concerned with the maturity and adaptability of new robotic technologies developed by RWTH and ETHZ for specific navigation tasks whereas the Design & Quality Control (QC) team led by EPFL is focused on the overall robotic product. Specifically, it is in charge of safety and design enhancements, interaction with stakeholders concerning standards and use cases plus potential ethical issues that may arise when deploying particular types of robots in different social environments. Testing of scenarios (venue planning, logistics and execution) is carried out by the entire team. As can be seen from the sub-team composition in Figure 2.2, different partners are in charge of each sub-team. This assures integrity, fairness and neutrality in assessing and judging work products of each sub-team by remaining partners. Furthermore, the team also schedules periodic review sessions with EC and solicits feedback from external stakeholders as an additional layer of checks and balances.

2.2 **What's New**

Mobile robotic research has been ongoing for several decades now. Navigational tests using electric wheelchairs started from early 1990's when proximity sensors (e.g. sonar) and portable computers were within the budget range of academic researchers. Nowadays, with the ubiquity of vision sensor modules, fast microprocessors and cheap memory resources, the focus has shifted from hardware-constrained robot prototype tests to exploitation of advanced artificial intelligence algorithms and real-time fusion and interpretation of multi-sensor data streams. The emphasis now is less on hardware design and more on software technologies. Any suitable robotic platform —commercial model or an academic prototype— is adequate as long as it supports a fairly open software architecture and additional hardware can be augmented with little difficulty. In fact, this same approach is taken by the Crowdbot team. Several members of the team have also gained knowledge and experience by participating in prior and ongoing European/EU national research projects such as EUROPA, EUROPA2, SPENCER, CV-SUPER, ADAPT, Mummer, STRANDS, iCub and so on. Nevertheless, we highlight several unique features and goals of Crowdbot that stand out when compared to other mobile robotic projects:

- *Use of Multiple Robots*: Crowdbot will test its navigation system and associated technologies on three different robotic platforms. The emphasis is obvious: the team's goal is to develop robotic software that is portable or adaptable across various kinds of autonomous mobile machines. Integration of software onto an existing robotic hardware is always a challenge. The team will pursue such integration using three very different robotic platforms: a closed, proprietary commercial humanoid (Pepper by SoftBank Robotics), a commercial electric wheelchair (Quickie mid-drive model) with extensive room for modification and a new service robot (CuyBot by Locomotec) in prototype/test phase.
- *Multi-Disciplinary Effort*: Robotics is already a multi-disciplinary field. Here, we are referring to cross-pollination of robotics and human crowd analysis. The study of human crowd behavior and motion profiles is the convergence of several disciplines: psychology, thermal dynamics, traffic engineering, mathematics, and computer science. In the past humans are treated merely as dynamic objects in a robotic test. Likewise, crowd analysis is restricted to human-to-human interaction only. The Crowdbot team has expertise in both robots and crowds; this synergy will be used to tackle an emerging problem in mobile robotics: social navigation of robots among dense human crowds.
- *Technology Transfer*: There are three main emphases here: 1) Academic to Commercial: Since two of the seven partners are non-academic (commercial) members, the team aims to demonstrate methodical transitioning of stand-alone intelligent software (developed by academic partners) into an embedded module of a commercial-grade robotic system. 2) Platform Portability: The same intelligent software as well as sensing and computing hardware will be adapted from one robotic platform to another. In general, this is not a simple task since each robot has its own design constraints (dimension, shape, software architecture and embedded hardware). 3) Current generation to the Next: Two rounds of navigation tests are planned and it is expected that 2nd round tests will undergo further enhancements based on lessons learned from round 1. Since sensor and computing technologies are advancing at a rapid rate, we also anticipate upgrading both sensing and computing technologies in 2nd round tests. The team will then gain insight into technology migration potential using the same robot.

In short, all three goals of Crowdbot aim to maintain its relevance after the completion of the project. They facilitate subsequent follow-on work by Crowdbot researchers and their peers.

2.3 **What's Outside of Scope**

The following system features are neither available nor activated among Crowdbot robots for navigational operations. Exceptions are noted whenever applicable.

- *Body Part Movement*: All Crowdbot robots move on wheels. Crowdbot sensing and navigation technologies may not apply to bi-pedaled (two-legged) or multi-legged robots with lateral and vertical motion profiles while moving forward. Humanoid robots tend to support pitch, raw and roll movements of certain body parts. These features are generally not exploited with the exception of head, limb and torso movement of Pepper in social navigation scenarios (see Section 4.7).
- *Human-Machine Interaction (HMI)*: Crowdbot robots use Human-Machine Interfaces (HMI) for in-seat (e.g. wheelchair) or remote control for testing, monitoring and troubleshooting by a human operator. However, once a test event starts, HMI for navigation tasks by a human is permitted for the wheelchair only. (See below for the restriction against tele-operation.)
- *Radio Communication*: It is anticipated that two-way wireless communication equipment will be used during various test phases to monitor system parameters as well as activate the kill switch when there is a safety concern. However, Crowdbot robots are designed to be semi- or fully-autonomous, self-sufficient and self-powered. Therefore, they cannot use off-site processors (e.g. cloud computing), nor real-time information exchange (e.g. map updates, navigation and timing data). They are not equipped with electronics for indoor or outdoor positioning or timing services (e.g. Global Positioning System, ultra-wideband localization, cellular networking timing). Here the reason for exclusion of such system features is not due to technology limitation. We do not allow circumvention of a technical problem by utilizing an easier work-around or by substitution of another technology.
- *Other Modes of Communication*: In socially interactive scenarios for Pepper and the wheelchair user, verbal and non-verbal communication with by-standing or moving humans using audio-visual cues (via its speakers and light/video displays) or use of a voice synthesizer and microphone to talk and listen to sound and speech is permitted. Of course, the wheelchair operator is allowed to communicate with other humans.
- *First-Person View & Teleoperation*: Since Crowdbot robots are likely to be equipped with vision cameras and two-way communication devices, they naturally lead to the possibility of First-Person View or tele-operation. This option is not permitted for navigation tasks. The robot must navigate using its own programmed or learned intelligence or be under continuous control of the human operator in the case of the smart wheelchair.

3. Review of Our Robots

The main message to be conveyed in this section is that a single set of requirements cannot be prepared for all three Crowdbot robots. The reader is referred to our companion reports *D2.1: Sensor Specifications* [15] and *D5.1: System Architecture* [16] for more in-depth technical description of our robots. Here the focus is to highlight features that are different and unique in each of our robots (from an operational perspective) and how they may result in specification of separate requirements for each robot. Key features relevant to requirements specifications are:

- *Frame & Body Composition*: A robot's physical dimension is critical for obstacle avoidance and safety measures when navigating through clutter and human crowd. The Pepper robot has the smallest footprint (size and weight) of a child while the CuyBot is similar to that of an adult. The wheelchair is the largest and heaviest robot with a rectangular, rigid frame. The composition of body parts limits the type and number of sensing and computing modules that can be attached or augmented to a particular robot. This constraint is applicable to both mechanical and electrical enhancements that may be necessary for successful navigational tasks.

- *Motion Profile*: Some of our robots are holonomic while some are non-holonomic. Even though all robots can move forward and backward, they may require a 180-degree turn to reverse its heading since vision sensors do not have a 360-degree field of view. In such a scenario, a non-holonomic robot requires additional clearance space to complete a turn.
- *Technology Integration*: This issue here is forward/backward compatibility and openness of the onboard digital system to technology enhancements. If the system is proprietary, new sensing and computing technologies cannot be integrated. The backward/forward compatibility issue is related to the integration of Crowdbot hardware and software onto the robot's pre-existing hardware, software and interconnect (data bus) architectures.

3.1 Humanoid Robot (HR)

The Pepper robot (left, Figure 3.1) is the model chosen for Crowdbot navigation tests. Other humanoids are also shown in Figure 3.1 to convey that humanoids come in different shapes, sizes and locomotion profiles. The Romeo (middle, Figure 3.1) has legs (i.e. bi-pedaled or two legs) whereas the remaining two are wheel-driven. As noted in Section 2.3, Crowdbot sensing and navigation technologies are limited to robots on wheels. As shown in Figure 3.2, a humanoid can be decomposed into several body parts: base, body, limbs and top or head. The PR robot (Figure, 3.1, right) is not very human-like but it has limbs (arms and claws) and a top where most of its vision sensing modules reside. As noted already, Crowdbot navigation tasks do not exploit dexterity or limb movements of a humanoid. The body parts most relevant to Crowdbot are base, body and top. The joints/limbs are not exploited (see Figure 3.2, left). Sensors exist as embedded (already integrated inside the robot), attached (placed or mounted externally to a body part) or augmented (attached or substituted in place of the previous embedded sensor).

When humanoids are compared to service robots (Section 3.3), they are both similar in many attributes except that the latter lacks joints/limbs and may or may not have a prominent top/head.

For the sake of technology integration, the Pepper humanoid is best viewed as a commercially available, closed-form marketplace robot that will undergo technology enhancements for navigation in a crowd. It base (wheels, motors and power electronics circuitry) cannot be modified and is used as-is. The digital electronics boards in the main body and head are also tightly integrated with embedded sensors and thus cannot be replaced or augmented. The availability of computing resources for running additional tasks and interconnection with external devices is highly dependent on a particular model and the age of its circuitry. On the other hand, a service robot or a smart wheelchair is a prototype that is already modular for augmentation or can be modified from its base design to support crowd navigation.

Figure 3.1: Different types of Humanoid Robots

Left: The three-wheeled Pepper robot by SoftBank Robotics [13] is used in Crowdbot
Middle: The bi-pedaled Romeo robot by SoftBank Robotics [13]
Right: The eight-wheeled PR2 robot by Willow Garage [14]

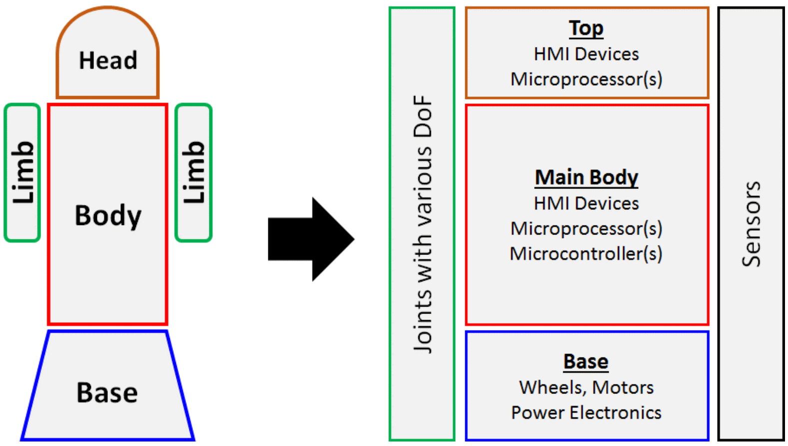


Figure 3.2: Framework & Composition of a generic Humanoid Robot

3.2 Smart Wheelchair (WC)

A smart or an intelligent wheelchair is a commercially available electric wheelchair that has been modified to support semi-autonomy features for navigation. In the past, researchers have custom built electric wheelchairs from the ground up when developing navigation technologies and testing of autonomy features. However, we can safely assume that current and future smart wheelchairs will be augmented or modified versions of commercially available electric wheelchairs. Three different models of an electric wheelchair by Quickie Wheelchairs [12] are shown in Figure 3.3. Such wheelchairs are also known as *motorized* or *power/powered* since they all run on battery power and electric motors for mobility. The electric wheelchair chosen for Crowdbot is similar to the mid-drive model (left side, Figure 3.3) where the motorized wheels are placed in the mid-section of the wheelchair. It has casters as front guiding wheels and casters in the rear for anti-tilt. We also show other possible drive mechanisms: rear-wheel and front-wheel drives (middle and right respectively in Figure 3.3). Note that each electric model has a structure that is slightly different from the other two and this has performance implications when augmenting sensing and computing technologies for intelligent and safe navigation purposes. We will revisit this topic shortly; we now cover the meaning of the term *autonomy* applied to the electric wheelchair.

Figure 3.3: Three different models of a power wheelchair from the same manufacturer
Left: Push handles, exposed side panels, caster wheels to the side of footrest
Middle: Adjustable armrest, folding headrest, folding footrest in front of caster wheels
Right: Front-wheel drive, anti-tilt wheels in front, reduced space under seat

A smart or intelligent wheelchair differs from a conventional powered wheelchair because it is able to perceive the environment (to some extent) and is equipped with some degree of autonomy. Just like in the automotive industry and as first described by Sheridan and Verplank (1978) [27] several different levels of autonomy are possible. Smart wheelchair autonomy can be broadly categorized into three different types:

1) *Full Autonomy*: In this option the human (in chair, remote location or aid) has no control over the wheelchair's motion behavior as its traverses from START to FINISH markers. Of course, the initial setup and programming of an operational scenario is done by a human. In practice, most end users prefer to maintain authority and develop skills where possible, so this type of control is not generally desired.
2) *Override Autonomy*: In this option the human operator relies on the machine to achieve all goals in the navigation profile, but maintains the capability to halt, modify or abandon any ongoing or future operations. This is the kill switch option and the human operator can restart or resume an operation as appropriate. This option is also known as *reactive navigation*.
3) *Assisted Autonomy (Shared Control)*: In this option both the human and the machine work in tandem by sharing the navigation task. When viewed in terms of space-time markers and goals, goals are achieved by combining the human input with the machine input to realize a final goal location output. If the human stops providing a continuous input, the wheelchair stops moving. In the literature this option is known more widely as *shared-control navigation* and will be the focus of the wheelchair control paradigms used in the Crowdbot project.

Options 2 and 3 are collectively known in the robotics community as semi-autonomous navigation. Option 1 can be viewed as machine-based navigation without a human-triggered kill switch. Note that different levels of autonomy exist because each is more suited to a particular type of wheelchair user. It is important to note that although wheelchairs are used by people with severe mobility impairments, many have compound and complex needs. For example, aside from the motor impairment, some users also have cognitive and/or sensory impairments. This topic is outside of the scope of this report and the reader is referred to these excellent sources [31, 32] for additional information regarding medical aspects of assisted wheelchair use.

From an operational perspective, these three options require different technologies and operational profiles for reaching a goal. Option 1 (full autonomy) is the most complex in design, use of sensors and computing resources since it assumes no human involvement. Option 2, reactive navigation, could be viewed as option 1 with a kill switch. Since the human operator has the final override authority, he/she must have the mental capacity and sufficiently fast reflex to make a quick judgment, but may lack mechanical means or long-term concentration to provide continuous control input to the system. Thus, option 2 is less complex than option 1 since it can rely on the human operator for safety concerns and against navigational hazards. Option 3 (shared control) assumes regular continuous human-machine interaction and the human operator plays a more significant role in the navigation task. Although one might imagine that having a human in the loop would lessen the design complexity and workload for the machine, it is actually still quite complex, especially given that the user input is likely to be fairly noisy (if the user input were perfect, there would be no need for shared control.) A typical example of assistance would be proactive obstacle avoidance e.g. helping to steer around a table. However in some cases, the user may actually wish to dock to the table. This type of ambiguity makes the system design rather challenging and is only further complicated by the type of dynamic "obstacles" (people) that we will be dealing with in Crowdbot.

Referring to the three models (from the same manufacturer) shown in Figure 3.3, we observe slight differences in the overall body structure of one model from the other two. This variation is to be expected since a different model is tailored for a certain type of user. This, however, complicates our task of requirements specification. As noted in Sections 2 and 3, certain operations cannot be performed by a robot due to limitation in its technology profiles.

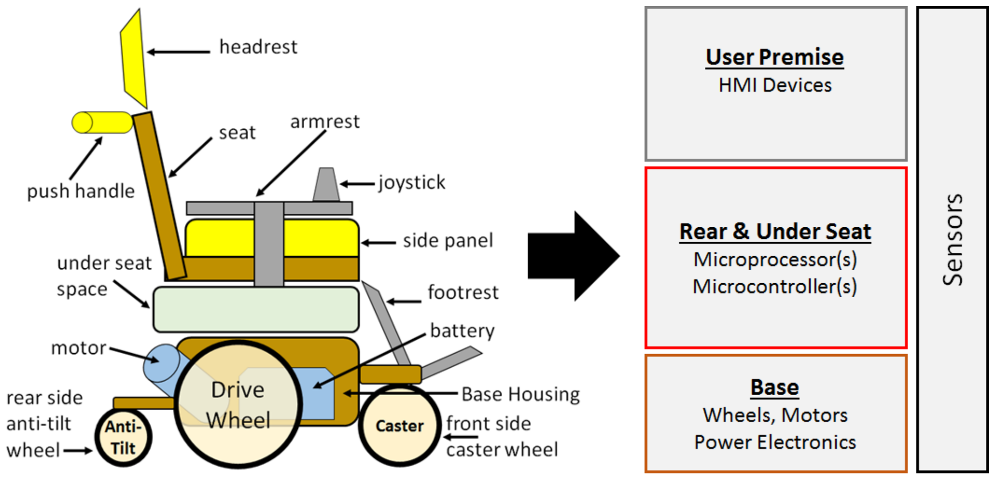


Figure 3.4: Framework & Composition of a generic Smart Wheelchair

Yellow-colored items are optional in certain models; Gray-colored items are non-rigid (i.e. movable)

For the electric wheelchair, we highlight in Figure 3.4 structural components that are rigid (consistent across all models) and those that may be optional (i.e. non-existent) in certain types.

We also show structures that are non-rigid (movable, foldable, extendable, etc.) such that any sensor attached to them may change position relative to the fixed base. Unless noted in our requirements specifications, we assume the smart wheelchair is composed of fixed base structures (excluding yellow-color parts) as shown in Figure 3.4. The available space for storage of a computer and other electronics is also highly dependent on the model; the mid-drive model has the largest rear space due to its large anti-tilt casters in the rear. The rear space behind the back support can also be used to house augmented computing equipment but this area is generally reserved for storage of handbag or backpack of the operator.

Compared to a humanoid or service robot, the smart wheelchair is the most flexible in terms of hardware attachment and technology integration since none of these features exist in the baseline electric model. However, unlike its counterparts, the wheelchair exists in many different physical models (even from the same manufacturer), each with a unique frame and body structure, such that care must be taken in both design and integration of Crowdbot technologies such that intelligent modules tested in one model could eventually be transferred to or adapted to a different model. In Crowdbot, we will focus on developing and evaluating the proof of concept using the popular Quickie Salsa M^2 mid-wheel drive. However, an optional requirement item is included in the list (see Table 5.1) that addresses this specific topic: Applicability and adaptability of a technology profile from one model to another.

3.3 CuyBot (CB)

The CuyBot robot by Locomotec is the third entry to our line of robots for the Crowdbot project. It falls under the category of a non-humanoid service robot. It is part of an ongoing development of a robot product line of which the first prototype is presented here [35]. During the course of the Crowdbot project more significant updates of the robot platform will take place. Also the name CuyBot is still a provisional name and might change in the future.

The prototype presented here has been built in the context of the R&D project NaRKo funded by the German government. The goal was to build a compliant robot that can work in a crowded hospital environment in order to execute service and logistics tasks. A picture of the fully-assembled robot is shown in Figure 3.5 (left) and its base in Figure 3.5 (right).

Figure 3.5: Picture of Locomotec's CuyBot (left)
At the bottom are three SmartWheels connected to a plate (right)

The robot's main features and structural details are as follows:

- *Base*: A new kind of kinematic units called "Smart Wheels". These units consist each of two wheels with independently controlled direct-drive motors which enable a quasi-holonomic motion of the robot. They come as compact units integrated with controller electronics and offer an excellent weight to payload ratio. The base unit itself is planned for commercialization as stand-alone components that can be integrated into robot bodies or other developers and robot manufacturers.
- *Body*: The robot main body that includes computing unit, sensors, battery, etc. An outer hull that is mounted compliantly on the base robot platform.
- *Top*: A rotating head with touch display and audio elements.
- *Sensors*: Several sensors, in particular a 2D laser range finder (Sick LMS 111) for navigation and a 3D camera (Intel RealSense D435) for people tracking.

A next iteration will exploit the platform concept in order to add transport capabilities with some cargo space (cf. Figure 3.7 left) to the robot. The current setup of the robot is explicitly targeted at the safe motion within crowds. Due to the concept of the robot it is expected to be a very suitable test platform in Crowdbot where sensors and components required for the implementation of project developments can be very flexibly integrated.

Figure 3.6: Different types of wheeled Service Robots

Sharship's pizza delivery robot (left); Savioke's room-service robot (middle); Aethon's luggage carrier (right)

Other service robots similar to the CuyBot are shown in Figure 3.6. Unlike a humanoid, they come in many shapes, sizes and weight classes. We can, however, define all wheeled service robots into the physical blocks shown in Figure 3.7 (left): base, body, top and cargo space. In some designs there is no separate structure for the top module but is blended with the main body. The main electrical and electronic components are grouped into its physical structures (Figure 3.7, right). Compared to a humanoid, a service robot is likely to be equipped with a cargo space to carry objects; it lacks joints or movement of any body parts. Similar to all other robot types, sensors are placed as needed in all structural parts of a robot.

As evident from the three service models shown in Figure 3.6, not much can be said about the motion profile of this class since they vary from one model to another. Compared to a humanoid, a service robot to likely to have a sturdier and more stable base construction with sufficient battery power and motor torque to achieve higher acceleration and greater maneuverability. Some are targeted for indoor smooth surfaces only but others can navigate through rough terrain.

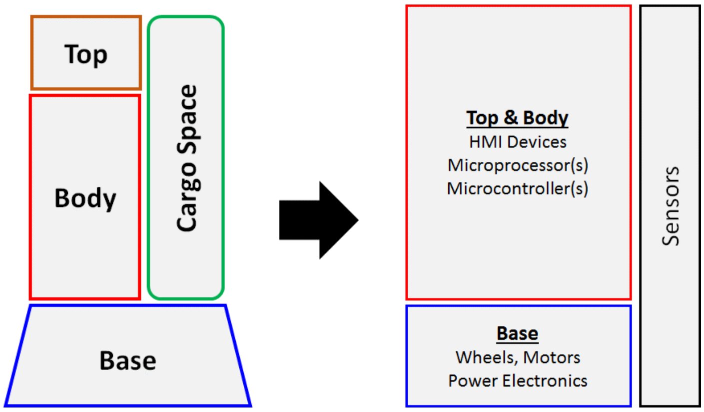


Figure 3.7: Framework & Composition of a generic Service Robot

3.4 Motion Profiles

All Crowdbot robots run on wheels. The non-holonomic wheelchair uses two drive wheels for both rotation and translation of its frame. It therefore cannot move sideways (translation without rotation) nor execute an omni-directional turn (rotation without translation). In contrast, both motion profiles are available in a holonomic robot such as Pepper. Both holonomic and non-holonomic motion profiles are illustrated in Figure 3.8. Certain wheelchair models have a drive train that allows wheels to rotate simultaneously in opposite direction and thus achieving an omni-directional (rotation only) turn but still requires the same maneuvers as a standard wheelchair for a sideway displacement.

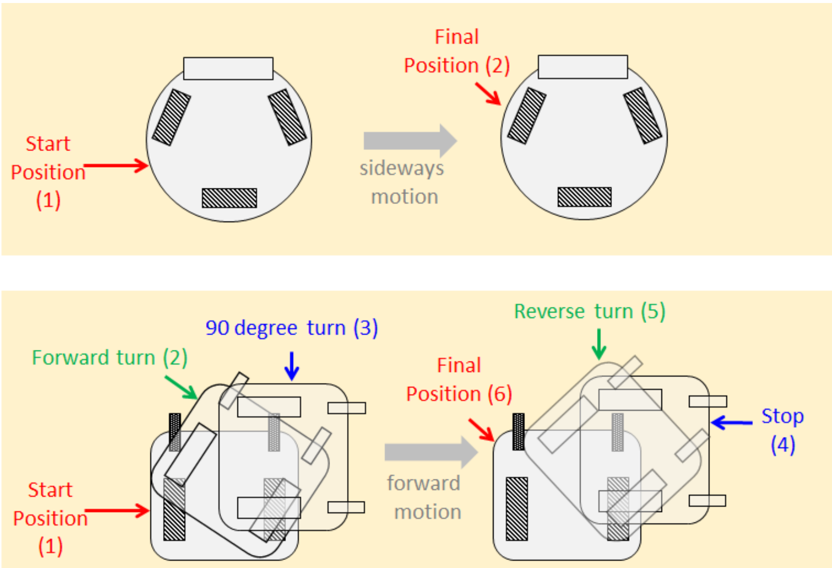


Figure 3.8: Motion Profiles for sideway movement of a Holonomic (top) vs. Non-Holonomic (bottom) Robot

3.5 Human-Machine Interface (HMI) Options

Crowdbot robots are designed to run in either semi- or full-autonomy mode of navigation. Even in full autonomous mode with no human intervention during actual navigation runs, it is expected that human-machine communication is still required for 1) pre-run setup and troubleshooting, 2) remote control and monitoring during run by a human operator, and 3) post-run data dump. Of course, for the smart wheelchair, continuous in-seat HMI device control is assumed in actual navigation runs since its default mode is semi-autonomous, shared control operation.

Commonly used human-robot communication methods are Ethernet, WiFi, Bluetooth or a computer communication bus (e.g. USB, serial port) or a microcontroller bus (e.g. CAN, I2C). As a safety measure, each robot must be equipped with a kill switch that can be activated remotely. This is listed as a requirement item in Table 5.1. A physical "red button" kill switch on the robot may not suffice since in an emergency it may be difficult to approach and make contact with a machine that is misbehaving. Other options are remote kill via wireless communication or blockage of future motion using a physical obstacle.

For in-seat control of wheelchair maneuvering, many technology options have been developed over the years to accommodate a user/operator with a specific type of disability or impairment. Details of these HMI options are outside the scope of this report. A summary can be found in [24, 25]. In Figure 3.9 we provide a pictorial summary of such HMI options. In-seat maneuvering involves changes to both direction and speed. If an HMI device is sensitive to speed adjustments, then it is called proportional. Non-proportional HMI switches are limited to simple tasks: stop-and-go, forward-and-backward and so on. In all Crowdbot tests the most common HMI option ─the proportional joystick─ will be used. No further HMI device requirements beyond the generic joystick are specified in the requirements list of Table 5.1.

Figure 3.9: HMI Options for the Smart Wheelchair Operator

A subject closely related to HMI options is the process flow of human-to-machine communication. A high-level description is portrayed in Figure 3.10. The left-side block "Human Operator" can be viewed as a remote controller, an in-seat human operator or a computer program that executes a programmed navigation task. Its output is typically (translational) speed (m/s or km/hr) and direction. All objects in the figure (human operator, HMI device, translator and motor control) are integrated hardware-software modules of a robot. The thick black arrows denote communication interfaces from one object to the other. In the case of a physical joystick, the communication interface is the human hand that jolts the stick to a certain direction. In the case of remote or computer control, contact with the joystick is non-physical and its output is simulated. Typical HMI outputs are forward, backward, left, right, neutral (mid-point in a joystick) and a proportionality value that corresponds to translational speed. The translator then converts speed and direction to rotational speeds (rpm's) of all motors with the aid of readings from other sensors for motor/wheel position, rotational speed and inertia measurement units.

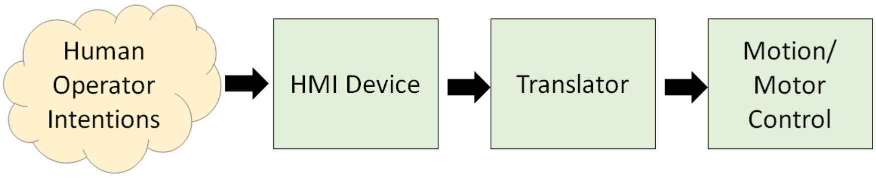


Figure 3.10: Human-Machine Interface Process Flow

In fully functional commercial machines (marketplace robots and electric wheelchairs), the motor control object is an integral component in the base of a robot with little or no room for design change or modification. In an electric wheelchair the HMI device is a physical component mounted on the body (e.g. the joystick sits on the top of an armrest). If the HMI device is physical, then the translator object already exists as an internal module, developed by the robot/wheelchair manufacturer. In this case the Crowdbot team can maneuver the robot by sending signals that imitate HMI device outputs using the same HMI-translator communication interface. On the other hand, if the robot is not equipped with a physical HMI device, the team must find an option to communicate with the translator object —the communication protocol and accepted input data formats of the translator must be known. In terms of requirements, no specific option is specified regarding the HMI process flow. The Technology Development (TD) and System Integration (SI) team are allowed to use any available means for robot maneuvering. The chosen methods are reported as requirements items.

3.6 Structural, Electrical & Electronics Augmentation

It is anticipated that Crowdbot robots will be augmented with additional sensors, computer processors and related accessories to meet robot safety, design and navigation goals laid out in this report. Since electrical and electronic components and modules must be embedded or attached to the baseline robot platform, structural modifications such as gluing, fastening and harnessing are also expected. Furthermore, the robot must be mobile and tetherless without any dangling cables or power cords. Thus, electrical DC or AC power must be supplied to augmented electrical and electronic devices via an onboard primary energy source such as a battery or a source such as a DC power supply module. For the sake of replication, reproducibility and technology migration purposes, the team will report on the following modifications to the baseline robot:

- Electrical and electronic device augmentation
- Additional computing resources used
- Structural modifications to accommodate augmentation
- Changes to dimension, weight and locomotion profile
- Removal or disabling of any embedded sensors and electronic devices
- Power supply or energy source modifications to accommodate augmentation

All modifications to the base structure and hardware components are annotated using robot design figures similar to those shown in Figures 3.11 and 3.12. For example, when a new sensor is augmented, its placement location as well as additional structural support to mount this sensor is shown on the base structure of Figure 3.11. Electrical power and data bus connections for this new sensor are annotated using a functional block drawing similar to the format shown in Figure 3.12.

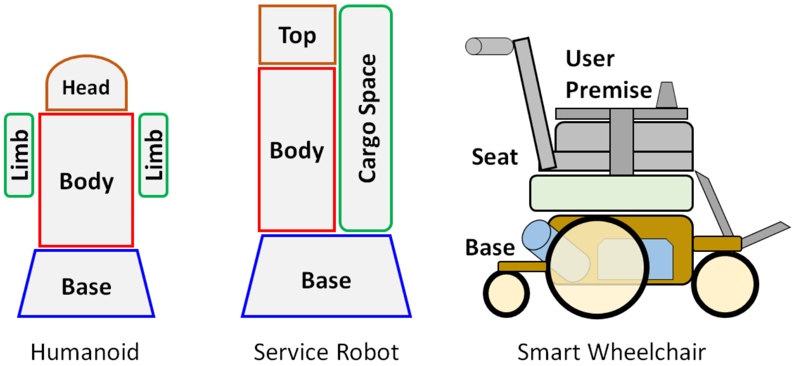


Figure 3.11: High-Level Structural Description of Crowdbot Robots

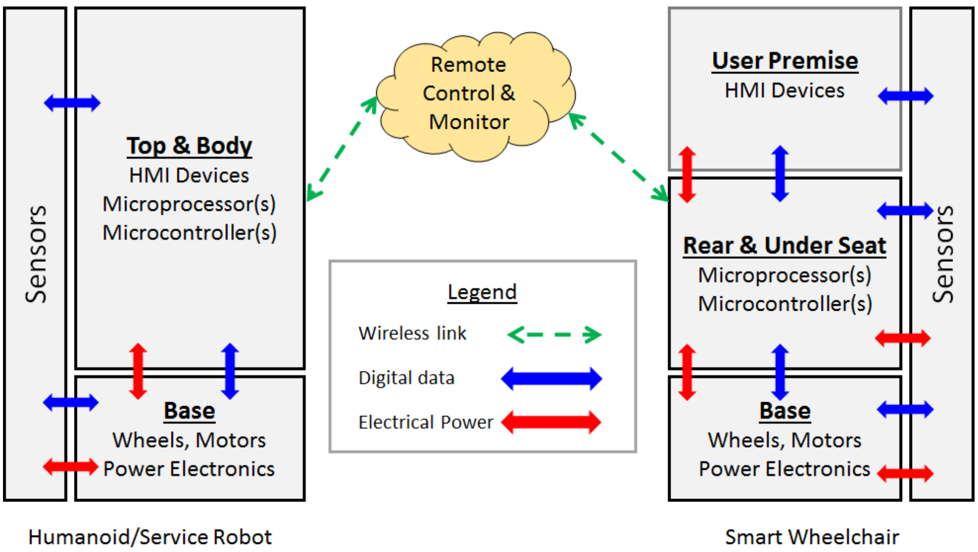


Figure 3.12: Hardware Composition and Interconnects of a Robot System

4. Test Scenarios

Seven main test scenarios are envisaged for assessment of Crowdbot navigation technologies. They are not mutually exclusive in testing various robotic features but each one targets a critical navigation operation. From a main scenario a number of derived sub-scenarios exist. Each is a baseline or a variation or more advanced version of another baseline scenario.

Before we dive into specific details of test scenarios, we first provide a high-level overview of problems that we plan to solve and insights we wish to gain from these tests. Figure 4.1 provides a breakdown of the three components (robot, humans and the environment) involved in a robotic navigation system test. A robot is a complex electro-mechanical machine with many built-in intelligent features but for each Crowdbot robot under test, we are particularly interested in its system-level features pertaining to safe and intelligent navigation. Let's clarify a bit further using one of the system features of a robot (Figure 4.1) as an example: A robot is equipped with a number and sensors and processors. Some are for internal system use (e.g. a voltage sensor to monitor the battery capacity) or a vision camera or proximity sensor to perceive the external world around it. Our test scenarios are not designed to assess the performance of a robot at the component/sub-system level. Hence we do not conduct specific tests to measure the quality of images captured by a color camera nor the upper limits of frame processing rate of an onboard computer. Instead we are interested in a robot's ability to navigate around or over objects that are scattered along its path. In fact, this example is defined as test scenario *S5.2: Hazards*.

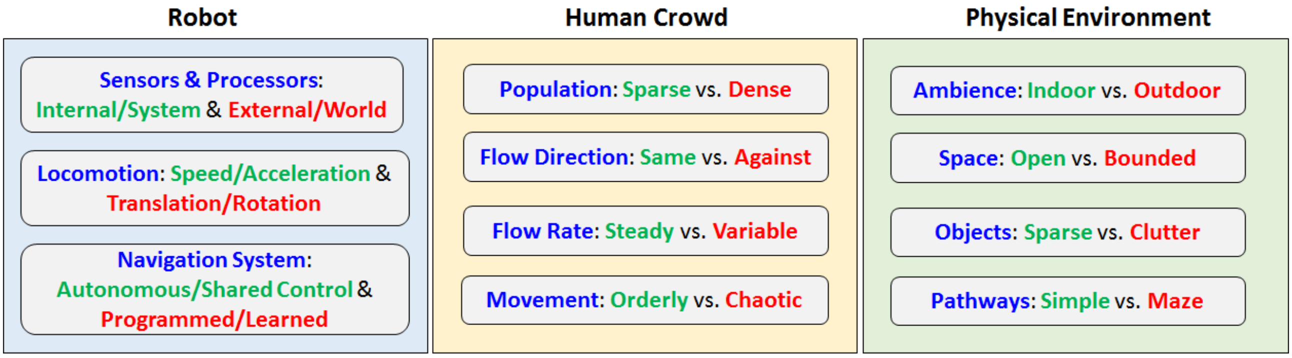


Figure 4.1: System level description of the components used in a robotic navigation system test

Besides a robot's system-level features, two other components —humans and the environment— are critical in bounding the objectives (i.e. requirements) and outcomes (i.e. test data) of a navigation test. From Figure 4.1 it can be seen that there are four main (navigation pertinent) features listed for both the human crowd and the physical environment. Other features such as the age or height distribution of the human crowd are not directly related or less critical to a navigation test. For each feature block in Figure 4.1, we define the two extremes (e.g. sparse vs. dense for human population) but any intermediate state/value between these extremes can be selected as a constraint in a test scenario. For example, the human population count in a test event (in a well-defined and bounded physical space) can be zero (none), ~3 (sparse), ~10 (moderate) or ~30 (dense). Please refer to Section 4.8 for concise definitions of human crowd density.

In summary, every test scenario is designed to assess a robot's system features in completing a specific navigation task under constraints defined for the human crowd and the physical environment. Four main feature options are defined for both the human crowd and the physical environment. Note that the execution of navigation tests with all possible combinations of human-environment constraints is not possible due to limited time, human resource and budget. Thus, in the section we list a selected set of scenarios that are most insightful (technical) and also meet project objectives (contractual) stated in the original proposal [23].

As shown in Figure 4.2 we will be using such icons to represent both a robot (as a tetrahedron) and humans (as rectangular cuboids) in *Scenario Maps* that portray stationary and dynamic behaviors of agents. In most cases the maps are in top-view (the ground plane is viewed from the top as in aerial view) or ground-view (the view is from the ground with the viewer sitting or standing on the ground) mode. When viewed from the top (see Figure 4.2b), the robot appears as an isosceles triangle with the third (non-equal) side serving as the rear part.

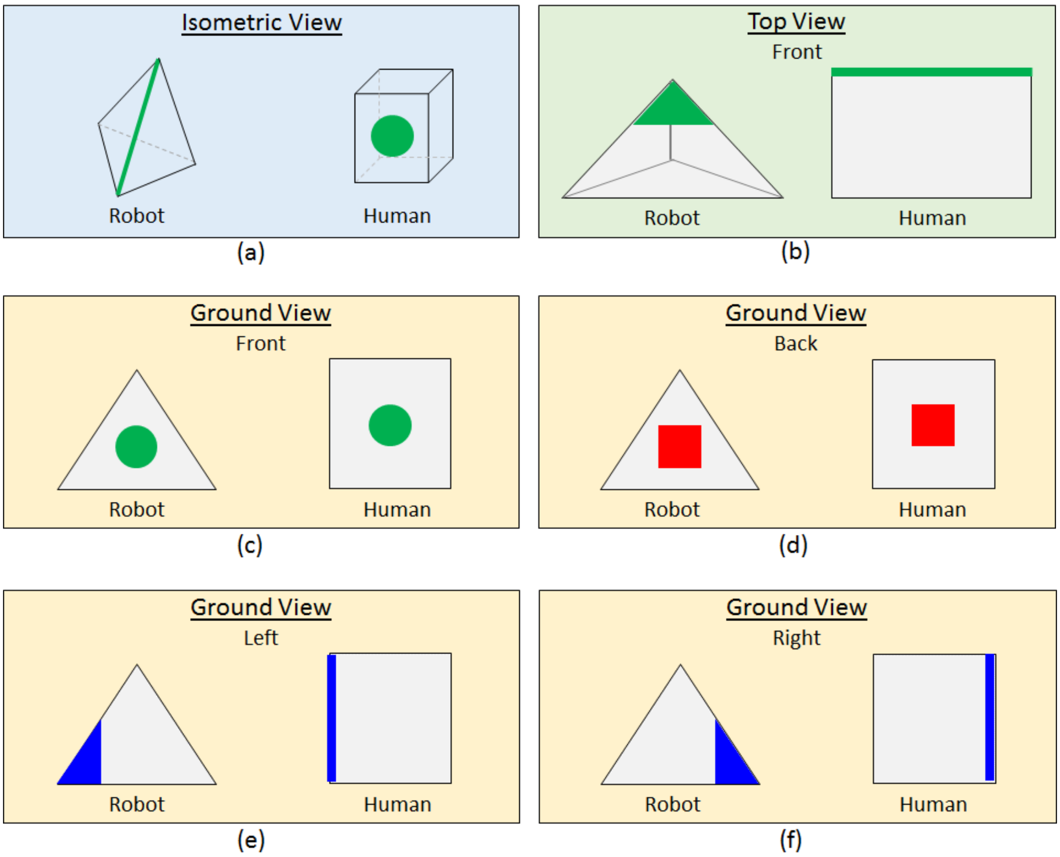


Figure 4.2: Symbolic icon representation for a robot and a human in scenario descriptions

The top view of a human is a rectangle. In the top views of agents a green color-coded region is added to illustrate the front side of an agent. In ground view illustrations, both the robot and humans retain the same (triangular and rectangular) shapes, respectively (see Figure 4.2c and d). Green-colored circular regions are used to denote the front side of both agents. Likewise, red-colored square regions are used to denote the rear side of both agents. For left- and right-side views of agents from the ground, we denote their front sides using blue-colored regions (see Figure 4.2e and f). Different color codes and shapes are used such that when we observe a two-dimensional drawing of a scenario, we can quickly deduce if agents are being viewed from the top, front, back or side. Furthermore, we will use different background colors (light blue, light green and light tan), each representing isometric, top and ground view, respectively.

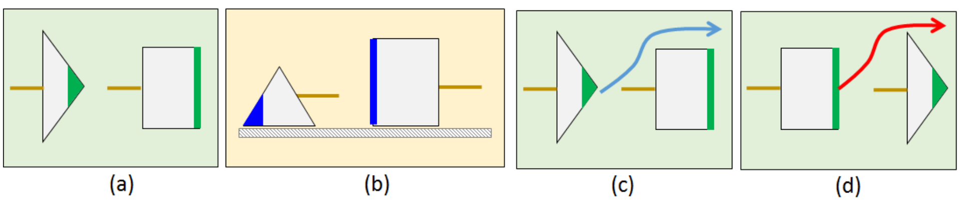


Figure 4.3: Colored trail or tail lines are used to denote past and future trajectory/heading of an agent

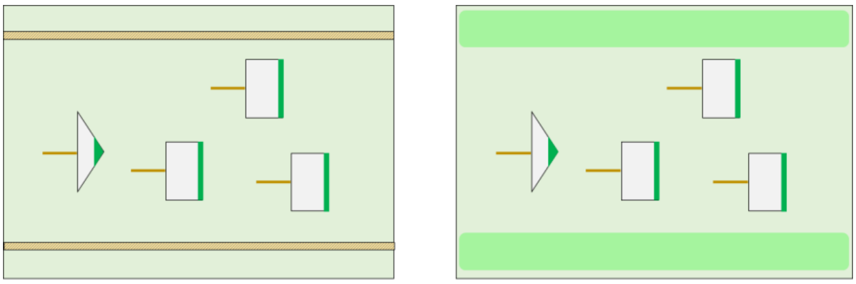
(a) Top view motion of agents moving north/top (b) Side view of agents moving left/west
(c) Blue curve for robot overtaking human (d) Red curve for human overtaking robot

In certain cases we will also include a trail (or a tail line) to an agent icon to denote its motion trajectory. In other cases the tail line is omitted since it is understood that the forward motion and heading are determined by the front side of an agent; i.e. the agent only moves forward; backward/reverse motion is not allowed. When the future motion plan of an agent needs to be explicitly notated, we will use a red-colored (for human) and blue-colored (for robot) curve with the arrow showing future heading. Such a trajectory plan illustration is limited to top-view scenario maps only. All cases are illustrated in Figure 4.3.

4.1 S1: 1D Flow

1D flow refers to test objects or *agents* (both human and robot) moving in the same direction. The traffic flow may be bounded physically by walls or fences or by social norms (e.g. walking on the sidewalk or pathway between lawns on both sides). Both cases are illustrated in Figure 4.4. The physical environment can be either indoor or outdoor. The main theme of S1 is to test a robot's ability to move along with other agents while maintaining a safe distance of separation from its immediate neighboring agents.

- *S1.1: 1D Following with the same constant speed*
  - All agents move at the same speed.
  - The test may be rerun at a different speed setting to assess a robot's locomotion profile and for humans to gauge an equivalent walking pace that a robot can sustain.
  - This is the simplest of all robotic tests since it is time-invariant (the robot sees the same humans around its personal space over time; Figure 4.5) but from a human participant's perspective, this test is insightful in understanding typical motion profiles of a robot since space occupancy of a human stride is very different from wheeled movement of a robot.
  - The test can be repeated by varying the spacing (crowd density) between agents.



(a) (b)

Figure 4.4: 1D Flow with robot following crowd scenario (without obstacles)
(a) Physical barriers/partitions and (b) Non-physical (e.g. social) barriers to bound space

- *S1.2: 1D Following with obstacles*
  - S1.1 tests are repeated with the insertion of static objects along the 1D flow path.
  - The physical space occupied by such objects is controlled such that they do not impede the flow (see S1.5 for bottlenecks). The goal is to test maneuverability around an object instead of straight-line human following movement. This scenario map is shown in Figure 4.6.
- *S1.3: 1D Following with different speeds*
  - S1.1 tests are modified with each agent assigned a unique, possibly different speed from other agents. In this scenario it is likely that certain human agents may overtake other agents. The over-take can be either marching straight ahead or followed by invisible lane change. The robot follows humans by adjusting its speed but does not overtake a human.

- Each human agent's speed is fixed/constant and does not vary over time.
- Unlike S1.1, the humans around the personal space of a robot may be changing due to over-take maneuvers by some humans. That is, from a robot's perspective, this scenario is time-varying.

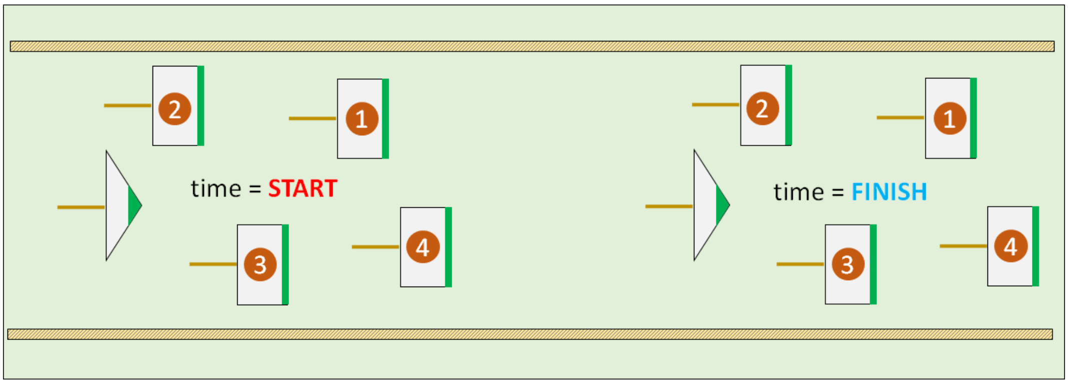


Figure 4.5: 1D Flow with all agents (both humans and robot) moving at the same constant speed

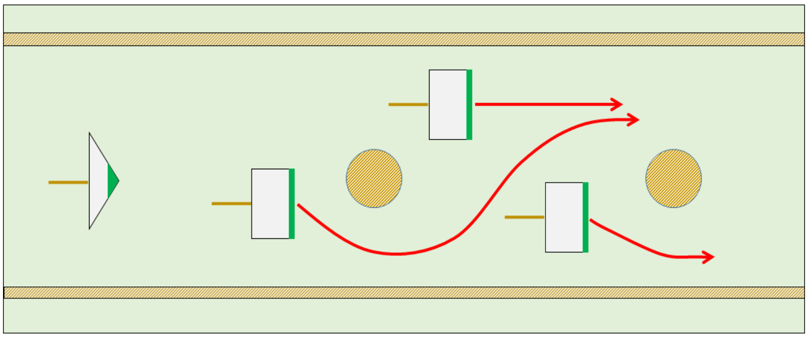


Figure 4.6: 1D Flow with robot following crowd scenario (with obstacles)
Circular objects are used to portray obstacles but any physical material and shape can be used in a test.

- *S1.4: 1D Following with different accelerations*
  - This is an advanced version of S1.3 where agents' speeds can change over time.
  - Similar to S1.3, a robot does not overtake a human agent.
  - Each human agent is assigned the same or different acceleration values.
  - A robot's speed and acceleration values are constrained by its structural design, safety margins and locomotion limits.
  - Unlike S1.1 or S1.3, this scenario may result in agent (robot or human) pausing/freezing due to acceleration/deceleration motion profiles to avoid collisions.

- *S1.5: 1D Following with bottlenecks/congestion*
  - This scenario is a major departure from S1.1—S1.4 in which flow is impeded by physical constraints or an increase in crowd density or both. Several test options exist.
  - Entry or exit doors are added to constrict 1D traffic flow. When the flow rate is increased, it resembles the general evacuation scenario.
  - Congestion is caused by narrowing of flow space; for example, the space between corridor walls is reduced or the amount of clutter from static object is increased.

- Crowd density is increased such that seamless over-taking of other agents is no longer possible. A human agent that wishes to overtake is blocked from an open path due to congestion by humans.

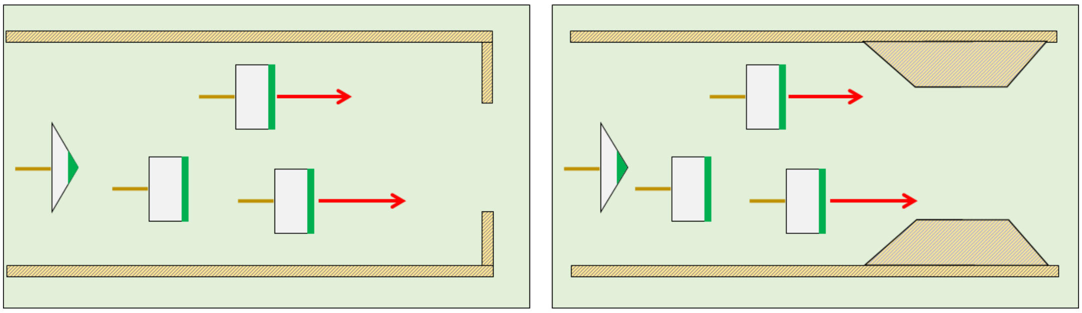


Figure 4.7: 1D Flow with congestion due to physical bounding scenarios
Doorway bottleneck (left) and narrowing of the width of partitions (right)

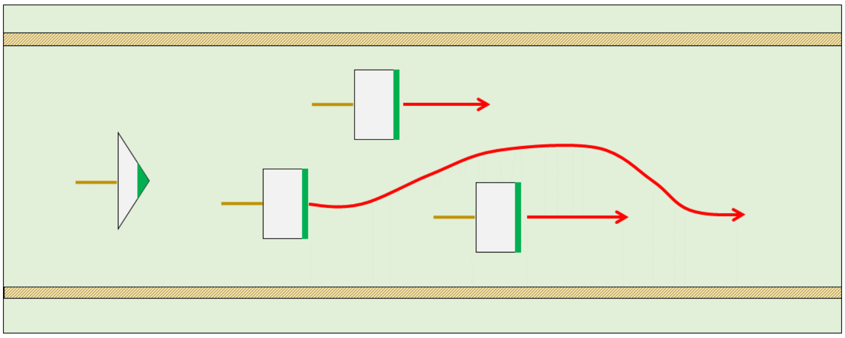


Figure 4.8: 1D Flow with lane-changing, over-taking agent scenario

The aggressive agent can stay in the lane it used to over-take or revert back to its original lane.

- *S1.6: 1D Following with over-take*
  - In this scenario the robot overtakes a human agent. Several options exist. Here the robot must maneuver around a moving object, a human agent —unlike the clutter scenario S1.2 where objects are static.
  - A robot with a higher but constant speed overtakes a human with a lower speed.
  - A robot exploits its acceleration profile to overtake a human agent.
  - An overtake may constitute moving sideways and marching ahead or this maneuver followed by another sideway motion to retain its original flow lane.
- *S1.7: 1D Following with being over-taken*
  - This is similar to S1.3 and S1.4 but explicitly requires a human agent to overtake a robot whereas in the previous two scenarios, it is mainly human-to-human overtake events.
  - The human agent may march forward after overtaking the robot or it may cut in front.
  - Both S1.6 and S1.7 result in double invisible lane changes by agents due to over-take and cut-in maneuvers. In all prior scenarios (S1.1 — S1.5) it is understood that invisible primary lane changing takes place but secondary cut-in maneuver is considered optional.

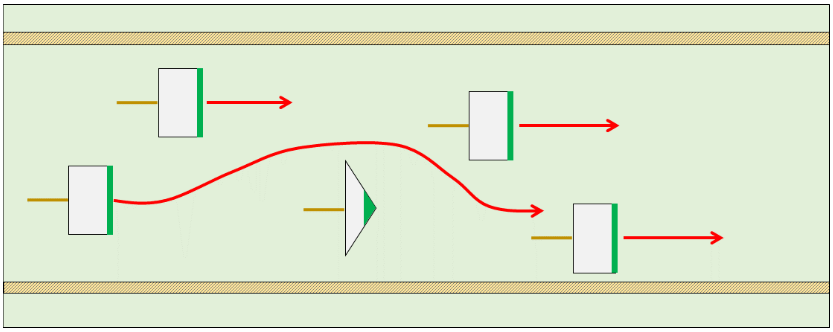


Figure 4.9: 1D Flow with human agent overtaking a robot scenario

Unlike the scenario in Figure 4.8, here the aggressive agent is required to revert back to its original lane.

4.2 S2: 2D Flow

In the most general setting a 2D flow is not a combination of two 1D flows side-by-side in opposite directions partitioned by an invisible wall. Agents flow in both directions without partition and they interact face-to-face. Near-collision events occur among humans but rarely a contact collision since abrupt stop and sideway foot and body maneuvers are used. With human-robot interaction, contact or collision may be unavoidable. The main theme of S2 is to test a robot's ability to avoid collision with an agent moving in the opposite direction whose relative speed is the sum of absolute speeds of this agent and the robot.

- *S2.1: 1D Following with Inverse 1D Flow*
    - This is equivalent to two 1D flows moving in opposite direction. The robot belongs to one side of the flows and is following its human agents. It is well known that in such 2D flow scenarios, invisible flow lanes are created organically since agents moving in the same direction are moving one another.
    - No sophisticated collision avoidance mechanism is expected from the robot.

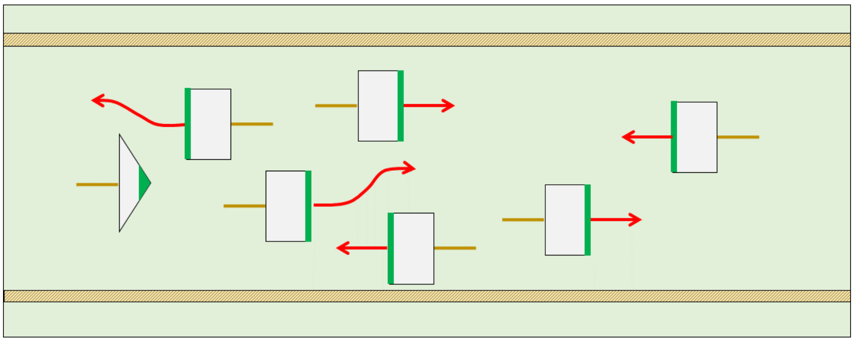


Figure 4.10: Simple 2D Flow with robot following 1D flow scenarios

- *S2.2: 1D Following with Inverse 1D Flow and Overtake*
    - This is an advanced version of S2.1 where certain human agents from the opposite flow engage in overtake maneuvers. Both humans and the robot must maneuver against these encroachers since they may (temporarily) breach into one of the invisible lanes.
    - For a robot the resulting maneuver is a combination of 1D following and sideway motion.

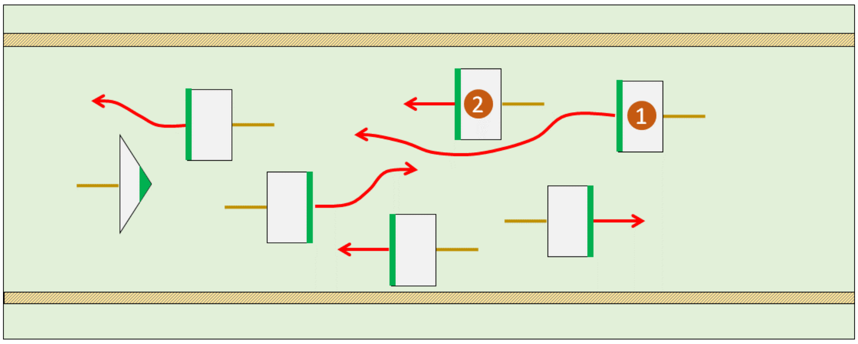


Figure 4.11: 2D Flow with over-taking agents in opposite-direction 1D flow scenario

- *S2.3: 1D Following and Overtake with Inverse 1D Flow*
  - This is the reverse scenario of S2.2. The human agent or the robot is now the encroacher since it is overtaking another human agent from the same flow while it is interacting with agents from the opposite flow.
  - Both encroachment scenarios should be tested. When a human agent engages in an over-take, it will likely cause a domino effect where the agents from the opposite direction react to its aggression. This in turn causes the robot to respond with collision avoidance maneuvers.

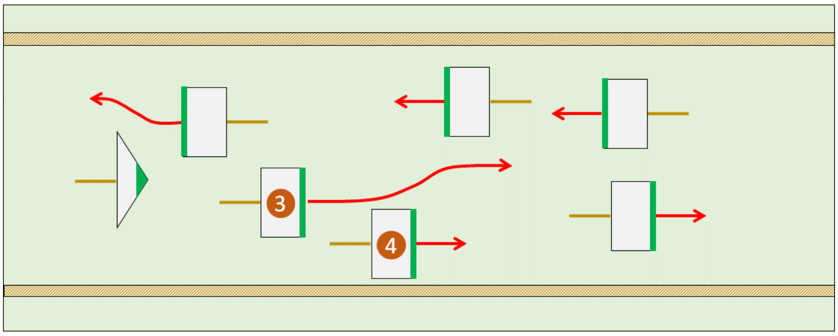


Figure 4.12: 2D Flow with over-taking agents in same-direction 1D flow scenario

- *S2.4: 1D Following and Inverse 1D Flow with Overtake in both directions*
  - This is the most complex scenario of 2D flow since accelerated motion profiles are used in both flows. This is the most realistic social model in dense urban city environments where pedestrian traffic flows in both directions (say, along a city sidewalk or underground train station hallway).
  - An advanced option to S2.4 is where certain human agents change their motion profiles: walk-then-pause, walk-then-leave the flow and join flow from the side. A related scenario where the robot changes its motion profile is covered in S4.

4.3 S3: Cross Flow

Cross flow (see Figure 4.14) refers to any movement pattern where the direction or heading between two or more agents is not limited to 0 (1D Flow) or 180 degrees (2D Flow) only. A common interpretation is that two flows cross when the difference in their heading angles is 90 degrees. This is the intersection, T-junction crossing pattern. However, we accept the interpretation of flows with heading angles that differ significantly from 0 or 180 degrees as cross flows.

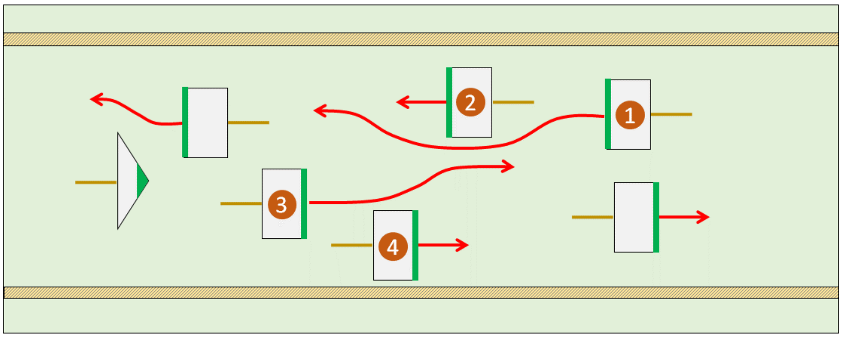


Figure 4.13: 2D Flow with over-taking agents in both directions scenario

Figure 4.14: Cross flow of human pedestrian traffic at Shibuya traffic junction; Source: [17]

With cross flows, the emphasis is different from that of S1 and S2. In both 1D and 2D flows, invisible lanes form where agents follow one another. Such lanes no longer form in Cross Flows. An agent must anticipate a gap in a cross flow for it to march through. The main theme of S3 is to test a robot's ability to understand the crossflow pattern and estimate speeds and acceleration of crossflow agents.

- *S3.1: 1D Cross Flow (1D x 1D)*
  - In this scenario two 1D flows cross path. For example, a north-to-south flow crosses with another flow going west-to-east. We assume the heading angle difference is 90 degrees.
  - Each 1D flow can have attributes defined in S1.1, S1.3, S1.4, S1.5, S1.6 and S1.7.
  - The rate of each 1D flow and its flow width (wall-to-wall space) can be the same or different although this variation is not critical to the main theme of this test.

- *S3.2: 2D Cross Flow (2D x 2D)*
  - This is a generalization of S3.1 where both crossing flows are of type 2D. We assume the difference in heading angles is 90 degrees.
  - Each 2D flow can have attributes defined in S2.1 — S2.4.

- The rate of each 1D flow and its flow width (wall-to-wall space) can be the same or different although this variation is not critical to the main theme of this test.
- Unlike S3.1, in this scenario there may not be any space gap in the cross flow to select a potential forward march option. Due to human flows from both directions severe occlusion is expected and thus the robot must navigate while inside the crowd.

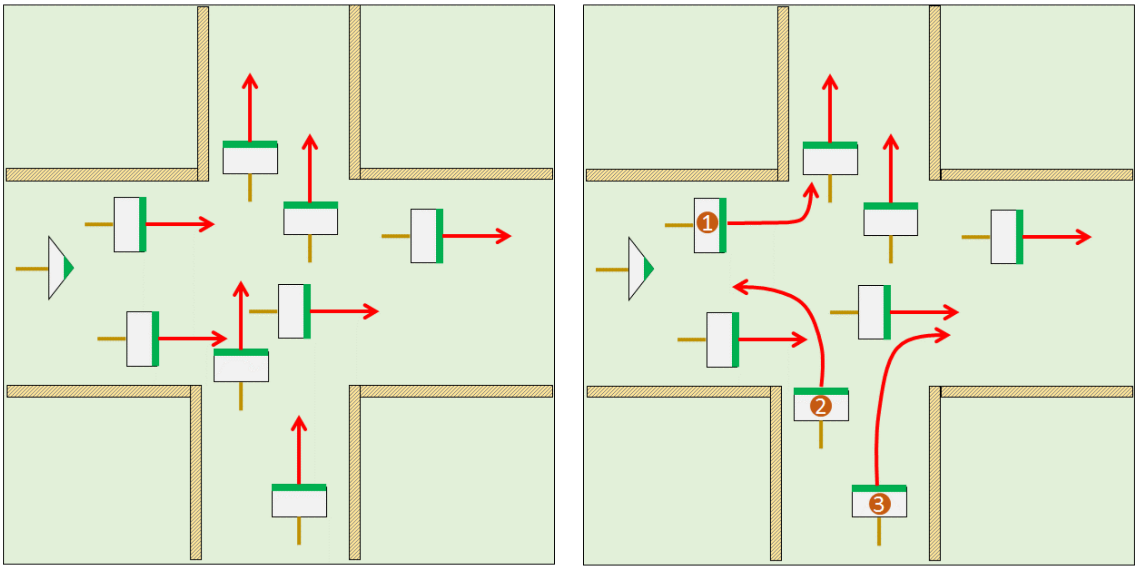


Figure 4.15: Cross Flow with two perpendicular 1D flows *(1D x 1D)* scenario

- *S3.3: Full Cross Flow (Shibuya model)*
  - The well-known zebra crossing of human pedestrian crowd at the X-intersection in the Shibuya district of Tokyo, Japan is known as the Shibuya model. This crossing consists of four street crossings and a diagonal crossing from one corner to the other. There is no zebra crossing on the other diagonal path but it is regularly used by pedestrians. Since the "walk" traffic signal is activated on all five crossings simultaneously, the Shibuya crowd (waiting at the zebra edge) bursts out into any of the six possible street-crossing flows.
  - The Shibuya model is the most sophisticated flow since it is a mix of four 2D x 2D flows plus two other 2D diagonal flows at 45 degree angle. This is the most complex flow pattern to be tested for Crowdbot navigation. A completely random particle behavior such as Brownian motion is not realistic and difficult to analyze.

From the surveillance camera views of the Shibuya crossing (Figure 4.17), it can be seen that most traffic flows along zebra crossing are conventional 2D flows, similar to scenarios from S1 and S2. The pedestrian traffic in the diagonal zebra crossings represent cross flows similar to S3.1 and S3.2 scenarios. Note, however, that pedestrian flow at waiting areas at the edges of zebra crossings cannot be accurately labeled as S1, S2 or S3.1—S3.2. This is because flow angles are neither opposite (180 degrees) nor perpendicular (90 degrees) but rather at a slanted angle. This case is represented in the scenario map of Figure 4.18 (right). Note also that in Shibuya's diagonal crossings, most pedestrians do not obey invisible (cultural norm) lanes and thereby walk as one pleases. This collective behavior is shown in Figure 4.18 (right), unlike the scenarios shown in Figure 4.16 where humans obey social norms (in these cases, moving on the right side).

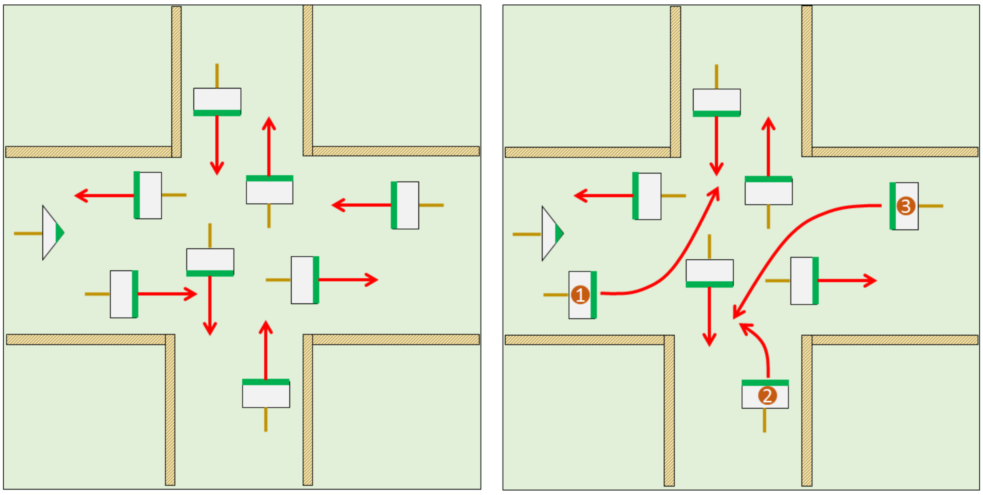


Figure 4.16: Cross Flow with two perpendicular 2D flows *(2D x 2D)* scenario

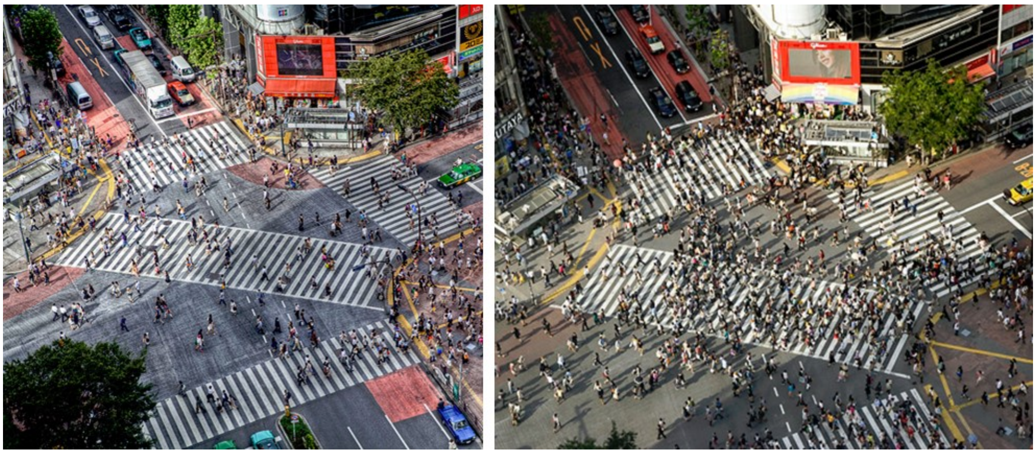


Figure 4.17: Pedestrian crowd at Shibuya crossing: light (left) and dense crowd (right)

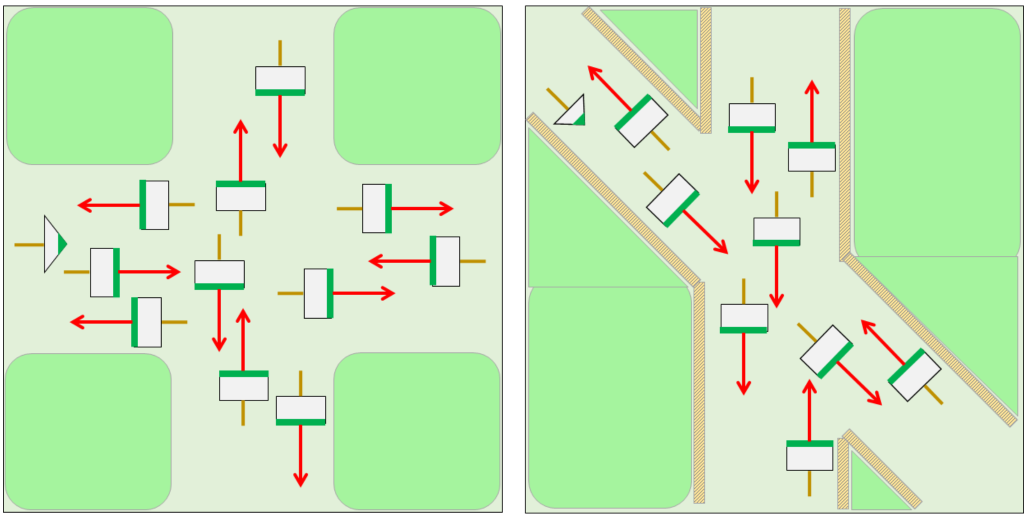


Figure 4.18: Shibuya 2D x 2D Cross Flow (left) and Diagonal Cross Flow scenario (right)

4.4 S4: Transitions

Transitions scenarios test a robot's ability to change its motion profile by varying its speed and acceleration or its heading/direction or both simultaneously. The change may be smooth such as joining or leaving a flow or abrupt such as a reactive stop or a sudden start. Such tests are applicable only to robots with high acceleration and torque capabilities.

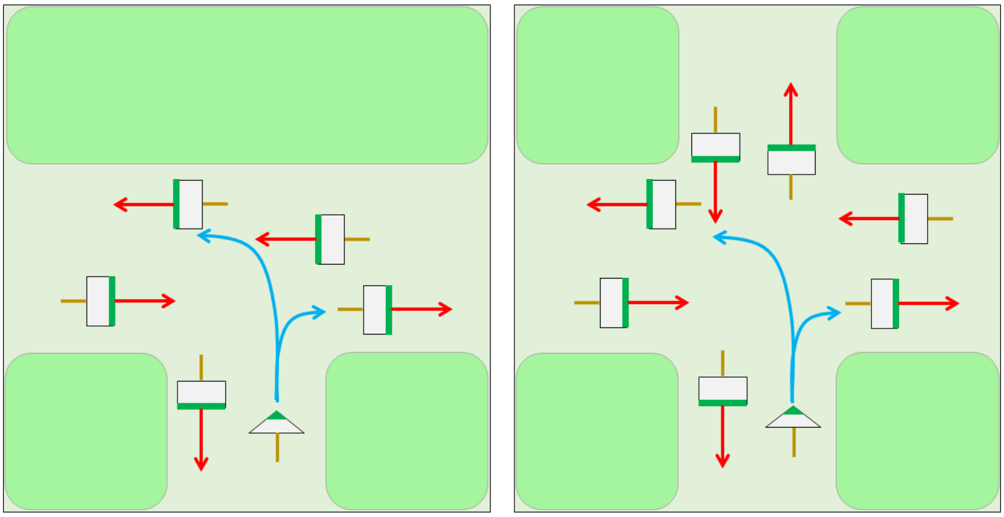


Figure 4.19: Left/Right Turn Transition Scenarios
Left: Robot joining cross flow; **Righ**t: Robot joining cross flow while dealing with opposite traffic

- *S4.1: Left/Right Turn*
    - This is similar to the pine cone car maneuver test during a driving license exam. A robot makes a left or right turn after heading straight. There may be physical bounds (walls or fences) to guide a turn. This scenario tests a robot's ability to make tight 90 degree angular turns as well as other smoother and curvy variations.
    - It also tests the accuracy of a robot turning to a specified slanted angle in an open-space floor. Unlike the previous option, here the test focuses on its onboard inertia, wheel rotation encoder and other sensors to guide its heading.
    - As shown in Figure 4.20 left, the robot must follow other agents in 1D flow but its heading is continuously changing. In Figure 4.20 right, the opposite case is tested where the robot changes its heading (to an assigned fixed angle) and no longer follows other agents.

- *S4.2: Join Flow (Stop-Go-Join)*
    - This scenario tests a robot's ability to join a flow from its initial rest state.
    - Based on the upper limit of a robot's acceleration, the speed of the ongoing flow may be adjusted such that the robot can join seamlessly.
    - This test is similar to the crossflow maneuver except that the robot is not marching through the crowd but is joining one. This requires a robot to make a quick side turn to join since it must initially be heading and resting perpendicular to the flow in order for its vison sensors (in the front) to observe and assess the flow behavior.

- *S4.3: Leave Flow (Follow-Deviate-Stop)*
    - This scenario is the opposite of S4.2. The robot, already moving along with the flow, chooses to deviate its path and exits the flow and then comes to a full stop. Several options exist for this test.

- In the simplest option the robot combines forward and sideway movements to exit from the crowd while decelerating such that it eventually comes to a stop.
- In a more advanced version a landmark or a side door at a future location is specified (its position is declared) or is discovered (e.g. vision sensing) as the final goal location. The robot must time its exit from the flow to arrive at the designated goal location.
- The difficulty of this test is dependent on the locomotion profile of the robot since it needs to move forward and sideways simultaneously to exit the flow.

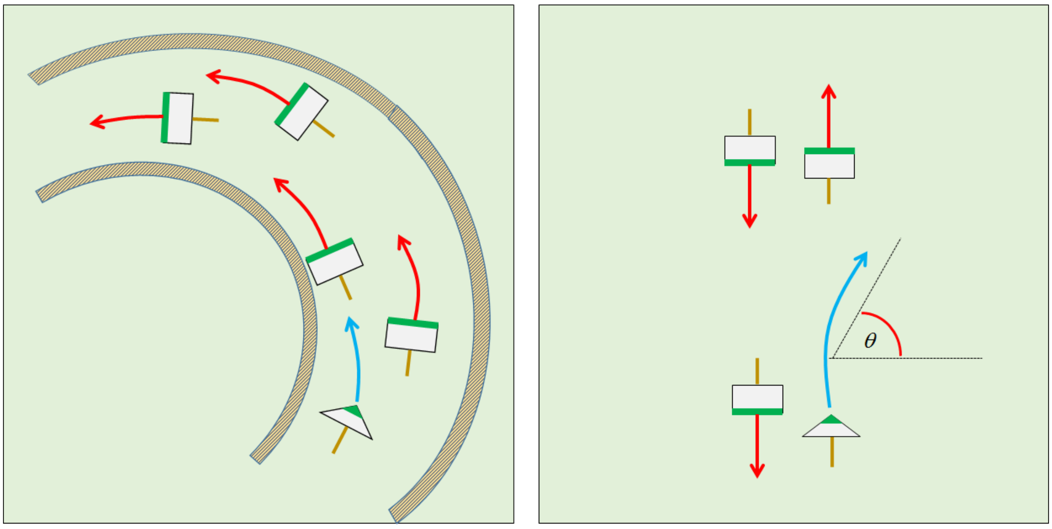


Figure 4.20: Curvy Turn (left) & Fixed-Angle (right) Turn Transition Scenarios

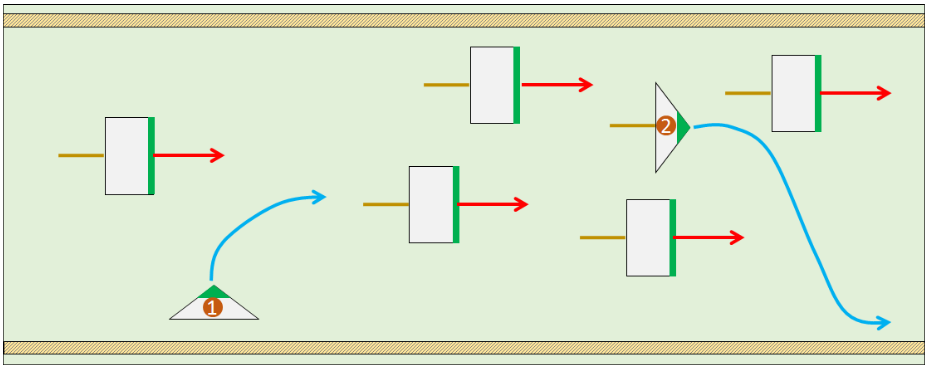


Figure 4.21: Join Flow (robot 1) and Leave Flow (robot 2) Transition Scenarios

4.5 S5: Maneuvers

The previous four scenarios S1 — S4 involve humans and traffic flows whereas S5 tests are for robots only with no human presence. The following maneuver tests assess a robot's ability to move from Point A to Point B in a safe and intelligent manner in a physical environment filled with clutter and different types of ground surfaces but void of humans.

- *S5.1: Squeeze (between objects)*
  - In this scenario the robot must march through a tight space such as a door whose width ca be varied or an area between two office desks or chairs.

- The goal is to assess the robot's ability to estimate the linear width or open space area along with knowledge of its own dimensions (footprint) and thus determine if it can march through or collide with objects while attempting it.
- The motion profile consists of a straight-line march as well as turns.
- Note that above cases are different from pine-cone maneuver tests where the wheel turn-ratios of the robot and the steering expertise of the operator are factors. Here we are judging a robot's ability to estimate its clearance.

- *S5.2: Hazards (fixed objects, holes, bumps, glass)*
  - The ground surface on which the robot moves may be littered with objects, holes and bumps (protrusion from the surface). This scenario tests a robot's ability to navigate around or over such hazards.
  - The glass door scenario is to test a robot's ability to recognize a glass surface and not crash into it.
  - These tests are also used to identify hazardous situations that cannot be overcome by the robot

- *S5.3: Ascend/Descend sloped surface*
  - In this scenario the robot moves along an inclined surface. Both upward and downward inclined surfaces will be used.
  - The test also includes a transition maneuver from a flat to an inclined surface and vice versa. Such maneuvers are routinely used by the electric wheelchair in outdoor navigation tasks (e.g. climbing up and down between the sidewalk and paved road).
  - The maximum inclination angle for safe operation of a robot is covered in S6. Here the focus is maneuverability of a robot.
  - An advanced version of S5.3 is the inclusion of hazardous objects on the surface. The robot must then detect and identify these objects while engaging in turns and sideway movements on slanted surface.
  - A variation of above sloped surface tests is where the surface is slanted to one side (a roll in the motion plane as opposed to pitch variation in up/down sloping of the surface).

.

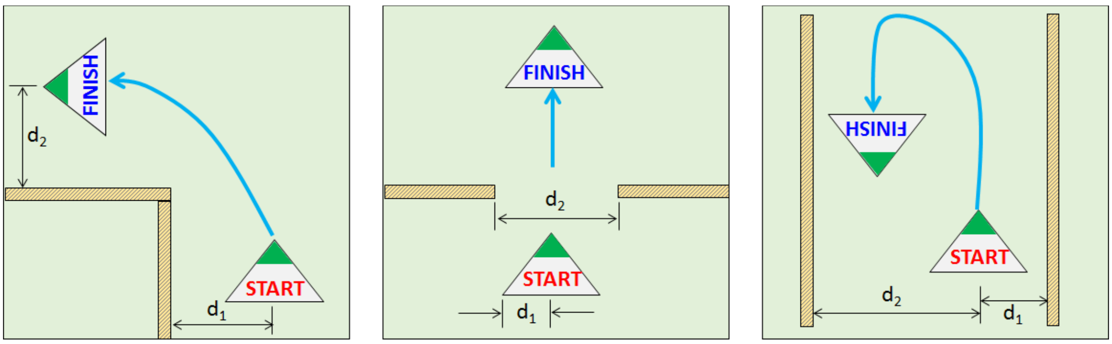
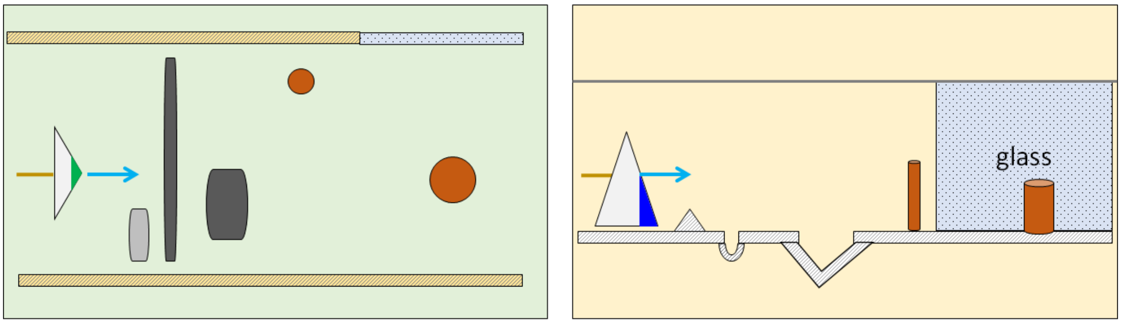


Figure 4.22: Squeeze Maneuver Scenarios

S5 tests are related to safety scenarios from S6 but their respective objectives are quite different. For example, when a robot gets stuck in a ditch, is it a hazard test (S5.2) or a malfunction test

(S6.4)? The correction interpretation is as follows: S5 tests are run such that the team is acutely aware of a robot's maneuvering capabilities as well as its technical limitations. If designed correctly, the robot will detect the ditch and proceed with the correct maneuver (move around or over it). If, however, the robot is programmed to move over it and then gets stuck, it is interpreted as a malfunction.



(a) (b)

Figure 4.23: Maneuvering around Hazards Scenarios

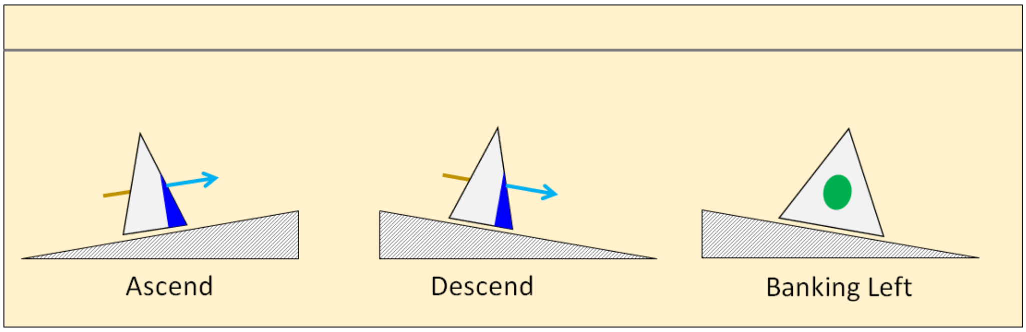


Figure 4.24: Maneuvering up and down Sloped Surfaces Scenarios

4.6 S6: Safety & Robustness

As evident from its title name, these scenarios assess a robot's ability to engage in navigation maneuvers while adhering to safety guidelines for both the robot and humans it comes into contact with. Specific safety measures and guidelines (both quantitative success thresholds as well as discoveries, see Section 5.1) for these tests will be derived from recommendations provided by the Design & Quality Control team. Robustness is the measure of its endurance (consistent operation over time) and operational resilience when its primary safety measures fail.

- *S6.1: One-time Contact/bump*
    - There are several parts to test in this scenario. A robot typically uses a combination of sensors (proximity, contact, vision) to perceive physical contact with an object or human.
    - The first is a physical contact test where a robot's responsiveness (time from contact to its response) is measured. Both soft and hard surface materials can be used as test objects.
    - Since contact sensors are on/off switches, no force or pressure exerted on the object by the robot is measured. This scenario is covered in S6.3.
    - The second is the dead-zone test where a robot's sensor suite does not register a contact even though one has occurred. This is the missed detection test.

- The third test is the opposite of missed detection: a false-positive test where the sensor suite registers contact even though no physical interaction occurred.

- *S6.2: Continuous Contact/Scrape*
    - This scenario is an extension to S6.1. It assesses a robot's ability to register contact that is ongoing after the first occurrence. Hence it is known as the scrape test where a robot has continuous contact with an object while in motion.
    - Proximity sensors are calibrated to generate a "true" output when it detects an object below a distance threshold. Thus it is not a true continuous contact sensor. Certain physical contact sensors provide continuous readings while others provide one-shot readings only.
    - We note that the continuous-contact (scrape) scenario (Figure 4.25) is different from the pressure-sensing test of S6.3 and Figure 4.26. In the former, the robot maintains contact with an external object while moving whereas in the latter, motion has ceased after contact (such as front or rear collision) but pressure is still exerted on the external object because the robot has not retracted or its motors/wheels are still engaged, resulting in an ongoing pressure on the external object.

- *S6.3: Force/Pressure*
    - Unlike contact based scenarios of S6.1—6.2, here the robot must be able to measure the impact of such contact using force/pressure sensors. The robot must be able to differentiate a strong force contact from that of a gentle touch.
    - The goal of force/pressure tests is not necessarily to measure physical exertion on the robot by the contact object but rather its reciprocal effect: the pain/force caused by the robot on the human.
    - Note, however, the test is requires attach force/pressure sensors on the robot but not on the contact object even though the goal is to measure force/pressure exerted by the robot.

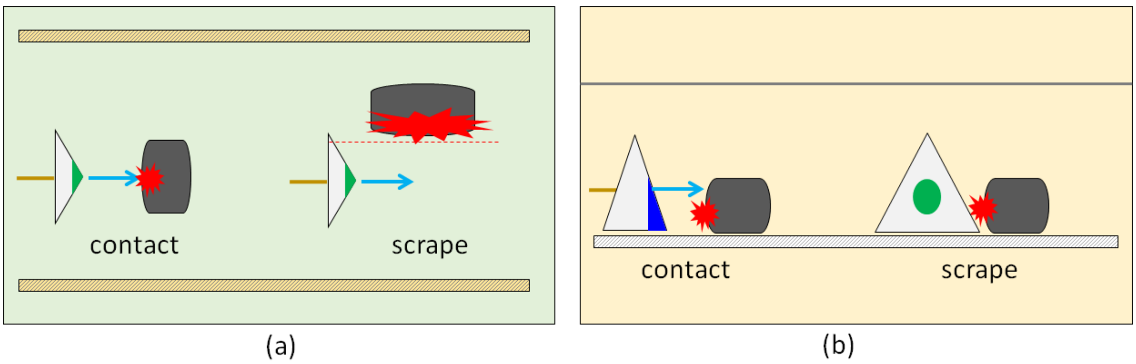


Figure 4.25: Safety Contact Scenarios (one-time and continuous scrape)

In the marketplace, currently available sensors to detect contact are very different from those dedicated to measure force or pressure. Some contact sensors detect "on" (contact) state only whereas other types detect both "on" and "off" states. Scenarios S6.1—6.3 are defined to collectively assess proper design and effectiveness of a robotic system in comparison to the touch-and-feel and pain sensing capabilities of humans.

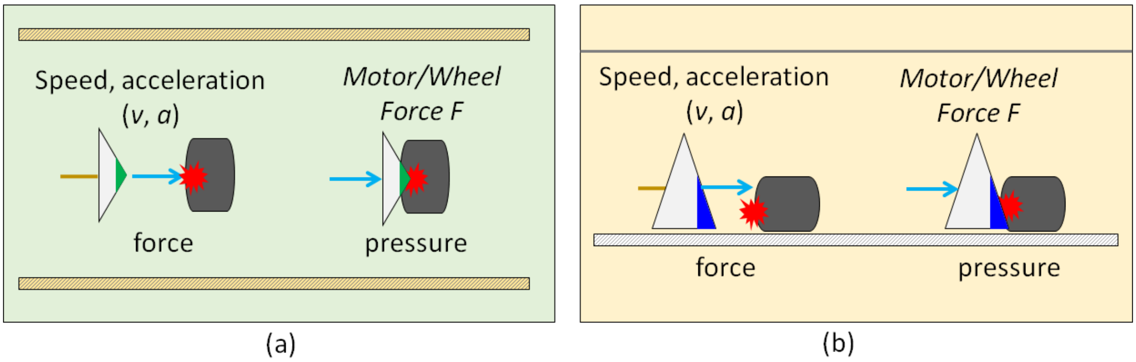


Figure 4.26: Safety Contact Scenarios (force and pressure sensing)

- *S6.4: Malfunction*
  - Malfunction scenarios are any kind of tests that expose the weakness, design flaw or software bug of the navigation system.
  - A red team can be formed to brainstorm such tests to break the system.
  - In such tests no changes are made to the robot but the physical environment and behavioral and motion profiles of humans can be modified to cause a robotic malfunction. Examples are selection and placements of objects to cause miss detection or false positives and human dynamic activities that cause confusion to its navigation plan.

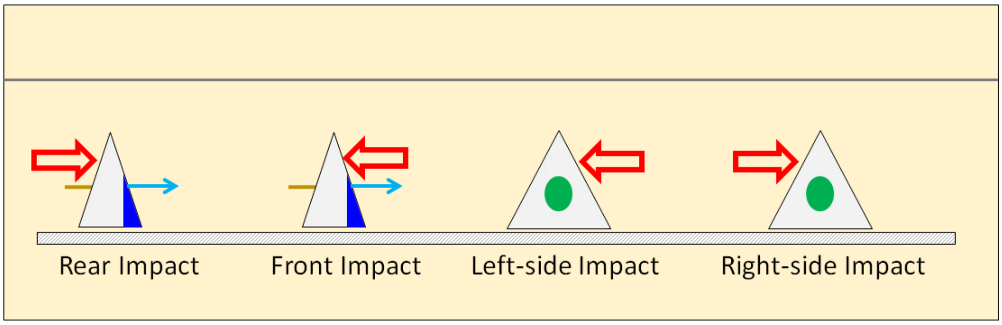


Figure 4.27: Safety Scenarios (fall inducement or detection)
Force impact from an external object to a different side of a robot

- *S6.5: Fall*
  - This scenario explores marginally safe conditions and triggers that may cause the robot to fall. It may not apply to all robots, especially those with default anti-tilt/fall features.
  - Due to safety and liability concerns, the actual tests may not cause a real physical fall (i.e. impact on the floor). The robot may be secured with harnesses to prevent a physical fall or held and controlled by a test team member.
  - The requirement objectives of these tests are two-fold: first, to document borderline and corner cases and trigger mechanisms that lead to instability to its base and second, if it falls, to document how it falls (face down, backwards or sideways) and the body parts suffer from the greatest force(s).
  - Potential bodily injury to a human due to a robot fall is not investigated in this test. Such kind of test requires strict ethical and legal approvals and are considered outside the scope.
- *S6.6: Blind*

- This scenario refers to cases where a sensing module of the robotic system becomes inactive. That is, it is no longer providing its readings to the processor (i.e. a blind sensor).
- The requirement objective is to assess the robustness of a robot against partial system failure or component outage.
- Here we are not testing the system architecture which can always be made more robust by adding redundancies and backup systems. The test scope is limited to sensors. For example, we are interested in the reactive behavior of the navigation system if it loses vision information from its color camera or readings from a subset of its proximity sensors.
- This test will identify critical sensor components that must be active and in good working order for the robot to continue its safe and intelligent navigation tasks.

- *S6.7: Endurance/Time Invariance*
  - The scenario is used to assess a robot's ability to maintain the same performance regardless of its length of operation. A robot's sensing and computing modules may degrade or improve as a function of their "on" time. Ranging and proximity sensors are said to suffer from drift where a bias to measured data grows as the length of operation increases.
  - Another endurance test assesses the management of onboard computing and memory storage resources. In a well-designed and managed system, performance degradation due to computer resource overload, buffer overflows and data loss/congestion at bus controllers and interfaces should not occur regardless of the length of operation.
  - This test does not measure the energy storage capacity of onboard batteries or the total operational time of a robot. That kind of endurance (i.e. longevity) is outside the scope of Crowdbot tests.

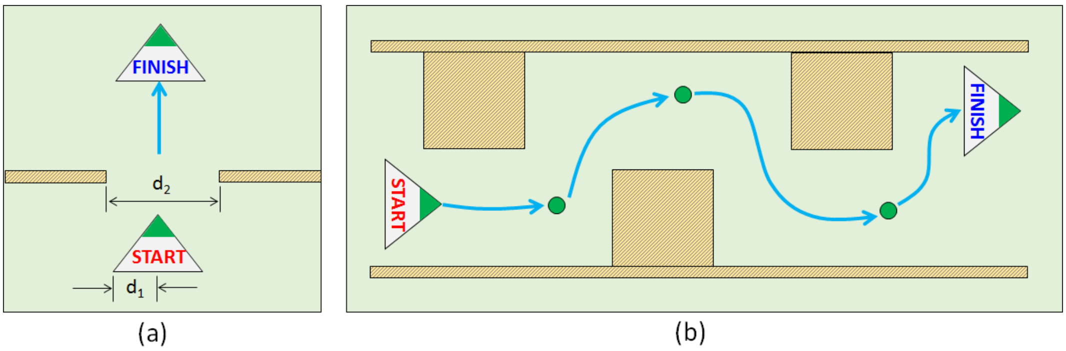


Figure 4.28: Robustness Scenarios (Time-invariant robot operation without data drift)
(a) Distance accuracy and (b) Goal location accuracy test scenarios after large time elapse

- *S6.8: Kill*
  - As a safety measure, every robot under test must be equipped an emergency kill switch —physically present on the robot or activated remotely via a communication link or both. This scenario is to test that such safety feature works as designed without any flaws.
  - An example of a flaw is where the physical kill switch is activated and it causes the robot to fall or during its activation, the human operator suffers from an injury.
  - If a remotely activated kill switch is the only safety measure, then the test team must demonstrate a backup plan to disable the robot in case the primary option fails.

Figure 4.29: Location of physical red button kill switches on Crowdbot robots

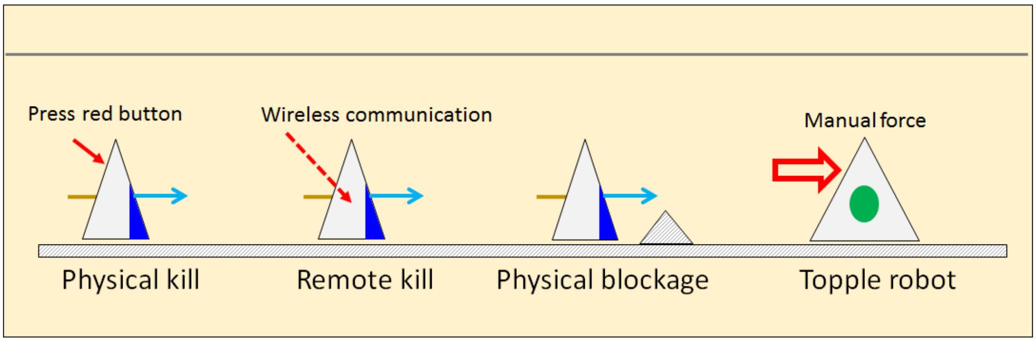
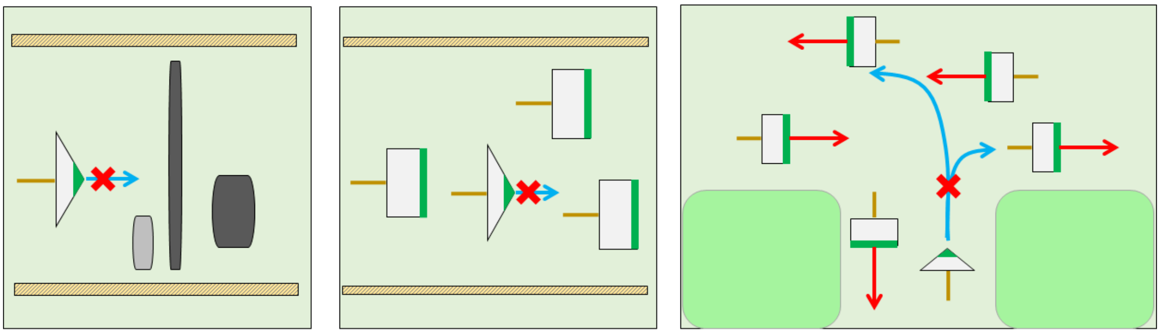


Figure 4.30: Safety & Robustness Scenarios (suspension of robot operation with a kill switch)

- *S6.9: Freeze/Stall*
  - This scenario refers to abnormal behavior of a robot exhibited during its system development and laboratory test phases. A robot may stall or freeze on its own due unknown reasons (See Figure 4.31). Note that we are not referring to conventional freezing due to collision avoidance; that is acceptable behavior. In contrast, we are concerned with abrupt loss of motion due to a computer crash, loss of power or some other reasons. Such observations must be recorded and reported as discovery-type requirement items.
  - In general, such anomalies where the root cause is not known are not easily reproducible in a controlled test environment but their realization cannot be ignored even if their frequency of occurrences is low.
  - If the cause is known or a suspect is identified, then this anomaly must be rectified and thoroughly tested before physical tests with humans are resumed.
  - Another issue closely related to freeze/stall is the *dead-zone* phenomenon. This occurs when a robot's sensors cannot detect or perceive objects in its vicinity or their fused results lead to ambiguity. (See Section 4.8 for a discussion on near- and far-field spaces.)
  - Identifying and reporting dead zones of a robot is not a requirement since this phenomenon is highly dependent on the number of sensors used and their placement on the robot. Exact measurement of dead zones is also a lengthy and tedious process with the need for precision instrumentation and setup and thus is not considered a valuable use of test time.

In automotive [33] and aviation [34] industries a malfunctioned and misbehaving automobile or aircraft cannot continue to operate. It is recalled, grounded or short-term mitigation options are proposed via urgent action bulletins. Since the Crowdbot team is pursuing scenarios where no prior work and results are available or unpublished, we take our test cases concerning safety very seriously. Test cases that result in robot malfunction or stall/freeze will be well-documented and its root causes investigated. Lack of reporting or under-reporting because the malfunctions clear up after a restart or reboot is not advised since these may be critical features for safe navigation.



(a) (b) (c)

Figure 4.31: Safety & Robustness Scenarios (reporting of abnormal robot operation such as stalling)

4.7 S7: Social Navigation

The following scenarios test socially aware navigation strategies on the commercial platform Pepper and thus are specific to the Pepper robot only. Social navigation is an emerging field of study for robotics now that they are proliferating among human-only spaces. The following scenarios are first-round attempts of social navigation specifically tailored for a humanoid such as Pepper using its visual and audio cues for interaction with humans. We already know that social interaction already takes place between a wheelchair operator and other humans. However, the emphasis of S7 is targeted at autonomous robots without human intervention. This requirement excludes the human-operator smart wheelchair or any tele-operated robot.

Returning to social navigation tests for Pepper, we focus on the following factors:

a) Safety: No physical harm
b) Comfort: Absence of annoyance and stress for humans
c) Naturalness: Similarity between robots and humans behaviour patterns
d) Sociability: Adherence to explicit high-level socio-cultural conventions

If the scenarios are carried out for other robotic platforms, they need to be adapted accordingly. As stated in Section 2.3, social navigation scenarios exploit certain human-machine interaction technologies which may or may not be supported by a particular robot platform.

- *S7.1: Guiding Human (or Passing Alone) in a Narrow Corridor*

The robot's task is to guide a person from point A to point B passing through a narrow corridor. The robot could encounter people coming in the opposite direction, going in the same direction, standing in the corridor, or no people at all. The sub-scenarios under S7.1 assumes space clearance for the robot only but not for the human it is guiding.

- *S7.1.1: Robot Passing by Humans in 1D Flow*
    - People/pedestrians are moving in the same direction as the robot. No human flow from the opposite direction.
    - This scenario assumes there is enough space to pass by the side of the people.

- If human speed < preferred speed of the robot, the robot passes by the side of the people, sets minimum personal distance to 0.7m.
- If minimum personal distance <0.7m, the robot announces: "Excuse me" and reduces its speed. After passing the people, it adjusts to preferred velocity.
- Otherwise, the robot could continue at the same velocity.

Figure 4.32: Pepper passing by non-blocking humans

- *S7.1.2: Robot Flowing Along with Humans in 1D Flow*
  - This scenario assumes there is not enough space to pass by the side of the people.
  - The robot adjusts its speed to match that of one of the people.
- *S7.1.3: Robot Passing by Stationary Humans (who are standing but not blocking)*
  - This scenario is a variation of S7.1.1 but with non-moving humans.
  - This scenario assumes there is enough space to pass by the side of the people.
  - The robot passes by the side of the people (depending on cultural information if possible).
  - It sets minimum personal distance to 0.7m.
  - If minimum personal distance <0.7m, it announces "Excuse me" and reduces its speed. After passing the people, it adjusts to preferred velocity.
- *S7.1.4: Robot Passing by Stationary Humans (who are standing and blocking)*
  - A variation of S7.1.3 with standing humans who are now blocking the robot's path.
  - This scenario assumes there is not enough space to pass by the side of the people.
  - The robot reduces its speed when distance = 2m.
  - It stops at a distance of 1.2m from the people.
  - The Robot gazes in the direction of people's faces and announces "Excuse me, I would like to pass."
  - It repeats similar speech every 20 secs until there is enough space to pass.
  - If successful, it passes by the space let by the people.
  - After passing the people it adjusts to preferred velocity.

Unlike Figure 4.32, there is no clearance behind humans for Pepper to pass by. Hence it engages with humans and requests permission to proceed.

- *S7.1.5: Robot Passing with Non-Blocking Humans in the opposite Flow*

  - This scenario is a variation of S7.1.3; now with people moving in the opposite direction. This is simplified 2D flow with no humans moving along with the robot. That is, all human flows are from the opposite direction.
  - It is assumed there is enough space to pass by the side of the people.
  - The robot takes the right side of the corridor (depending on cultural information).
  - It sets its minimum personal distance to 0.7m.
  - If minimum personal distance <0.7m, it reduces its speed and announces: "Excuse me."
  - After passing the people it adjusts to preferred velocity.
- *S7.1.6: Robot Passing with Blocking Humans in the opposite Flow*
  - This is a variation of S7.1.5 with humans flowing from the opposite direction blocking the robot's path.
  - There is not enough space to pass by the side of the people.
  - The robot reduces its speed and takes the right side of the corridor (depending on cultural information).
  - If minimum personal distance <0.7m, it stops.
  - If people are blocking for more than 10 sec., the robot announces: "Excuse me, I would like to pass."
  - It repeats similar speech every 20 sec. until there is enough space to pass.
  - After passing the people it adjusts to preferred velocity.

Figure 4.33: Pepper trying to pass by blocking humans

- *S7.2: Guiding Human (or Passing Alone) in Open Space*

This scenario is a variation of S7.1 where the physical environment is modified from a bounded indoor corridor space to a wider outdoor open space. The scenarios outlined in Sections 4.1—4.4 (1D Flow, 2D Flow, Crossflow, and Transitions) are adapted with social navigation principles from scenarios S7.1.1—S7.1.6 for an outdoor open space environment. The robot should follow as much as possible the planned path given by the navigation module, this could be achieved by penalizing the robot when moving away from the planned path. For this purpose, the robot could have a virtual space covering the planned path.

No specific sub-scenarios are defined for S7.2. If the team is successful in conducting S5.1—5.4 and S7.1.1—S7.1.6 and time allows, then the test can proceed with any of the possible sub-scenarios noted in S7.2. In passing maneuvers proposed in S7.1, it is implicitly assumed that the robot is able to measure distance (to determine clearance) for it to pass by humans. In an indoor office space environment, the wall, furniture and other office objects serve as physical boundaries. However, in outdoor or open space environments, the physical constraints are likely to be humans and the ground terrain (e.g. unevenness, rough pavement or social norms).

- *S7.3: Interactive Navigation*

A robot could be guiding or being guided by a human or simply navigating around, and at any moment there could be interaction between the robot and the human or other persons for a number of reasons. A human who is guiding or being guided by the robot is called the *user*. Other humans that the robot may interact while navigating are known as *people*. Some pertinent interactive scenarios while navigating are:

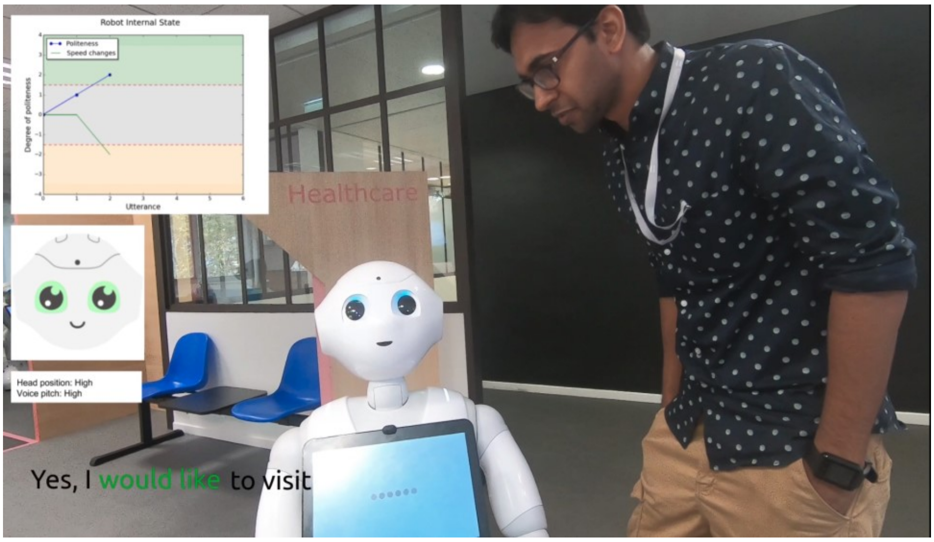


Figure 4.34: Pepper engaging in a conversation with its user

- *S7.3.1: User asking robot for information*
    - The user conducts an interactive communication session with the robot while navigating. Both the robot and user may be stationary or moving.
    - For example, the user may ask for the location of a specific place.
- *S7.3.2: User orders the robot to stop*
    - In this scenario the user commands the robot to stop and end its navigation plan.
    - For example, the user changes its mind or has other things to do.
- *S7.3.3: User orders the robot to pause*
    - In this scenario the user commands the robot to pause to let other people pass by.
    - Unlike S7.3.2, the robot will continue its navigation plan and continue moving once the user gives permission or it determines that it is safe to start moving again using the same principles and margins outlined in S7.1.
- *S7.3.4: User casually talking with robot*
    - The robot and user communicates. Unlike S7.3.1, no factual information is relayed from the robot to the user. Here it is merely a chit-chat.
    - The robot will use this interaction to modulate its behaviors or to better understand the social context or situation.

- *S7.3.5: Robot is proactively talking with people*
  - The robot interacts and engages with people in a more natural manner, or to get some information.
  - This scenario assumes that there are people in its vicinity and that the robot obeys its social navigation principles outlined in S7.1.
  - This scenario can be an enhance version of S7.1.3 S7.1.6 where the robot communicates beyond the announcement of "Excuse Me" to pass by humans. Once it passes by, it can also announce "Thank You" and so on.

- *S7.4: Shared Control Navigation*

This scenario is designed for people with special needs such as elderly or blind people. The user may lean on the robot in order to walk and at the same time control the direction of the robot if needed. Note that the term "Shared Control" in robotics has many categories and different interpretations. In this context for the Pepper robot, it refers to ad-hoc, as-needed navigation guidance provided by the user (a blind or an elderly person) to Pepper. This differs from the shared control navigation task of a wheelchair user where its guidance and control via the HMI device is continuous. If the user no longer exerts control of the HMI device (e.g. joystick), the wheelchair comes to a halt whereas in the Pepper case, it continues it navigation maneuvers until the user sends an interrupt command such as a pull, touch or some other means for communication.

- *S7.4.1: Robot assisting blind people*
  - Given that the user is blind, this scenario assumes that the robot is the primary controller of navigation.
  - The robot should be aware of the presence of the user. It can use any available onboard technologies (e.g. touch or force sensors, voice communication, etc.) to detect and monitor the presence of the user.
  - The robot can pause or end the navigation if the user is no longer leaning on it.
  - When moving, the robot should adapt to the pace of the user. A robot may use the force exerted by the user or some other technology to determine an acceptable pace.
- *S7.4.2: Robot assisting elderly people*
  - Given that the user is an elderly but receptive, this scenario assumes that the user is the primary controller of navigation.
  - This scenario is a variation of S7.4.1 with a different kind of user but everything else remains the same. The fact that the elderly user has vision can be exploited for human-robot interaction.
  - As in S7.4.1, the robot should be aware of the presence of the user and move at a pace acceptable to the user.

- *S7.5: Reactive Positioning in cluttered environments*

This scenario addresses the personal space that a robot must maintain while navigating. In order to preserve the social conventions of equi distances, the robot should be able to assess the current environment dynamically and react by keeping a similar distance between all the obstacles detected in the scene. We show some examples of the desired reaction from the robot (triangle) to the presence of obstacles (squares).

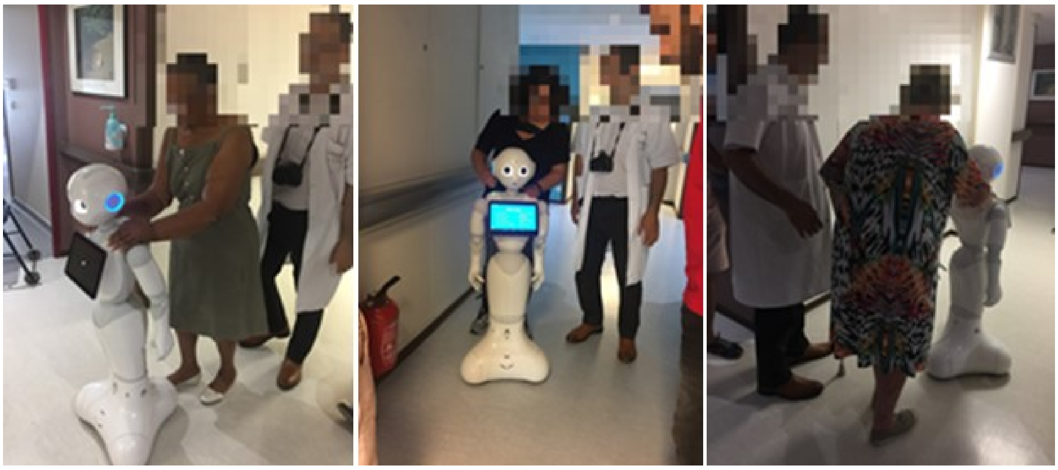


Figure 4.35: Pepper leading or guiding in a shared control navigation scenario

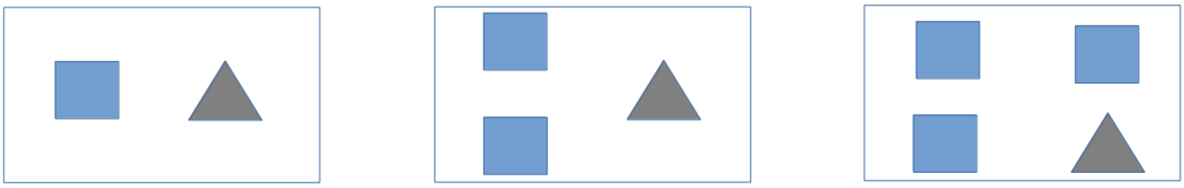


Figure 4.36: Personal space between robot and humans that preserve social conventions

- *S7.6: Social meeting*

This scenario addresses the maneuver a robot should make before engaging with a human. In general a robot could approach a person for different reasons, for example to greet, to inform, to deliver an object, etc. This task requires that the robot plan and execute a path to approach a person in a social manner. There have been some studies related to this scenario and they state that the robot should approach making a curve instead of a straight line. Also the robot should maintain a social distance according to the proxemics. This proxemics can be modified in case of the need to deliver an object to the user, in such case the robot should stand at a position where the user could receive the object.

No specific sub-scenarios are defined for S7.6. If the team is successful in conducting social navigation scenarios in S7.1—S7.5 and time allows, then the test can proceed with any of the possible sub-scenarios noted in S7.6.

4.8 Crowd Density & Personal Space

This section provides a short summary of techniques and methodologies used to measure or estimate (human) crowd size in a well-defined physical environment. Rigorous treatment of crowd analysis can be found in these excellent references [20, 21, 22]. Sizing of a large and dense crowd is mostly science but sometimes a combination of art and science. Figure 4.37 provides two cases where the crowd size is measured using a computer algorithm-based human tracker (left) or by using an estimation technique based on typical crowd size in a unit area (i.e. crowd density) and exposed ground surface area (with no human occupancy).

(a) (b)

Figure 4.37: Measure of large, dense crowd at a (a) marathon event [18] (b) Presidential Inauguration [19]

For scenario descriptions our scope is limited to definitions and quantitative measures of human crowd size as identified or perceived from a robot's sensing perspective. Two critical factors in quantifying crowd size are:

- *Dynamics*: As evident from the runners in motion at a marathon event (Figure 4.37a), the crowd size is time and location dependent if humans are moving. This is similar to the thermodynamic effects of particles in a contained space or volume.
- *Clustering*: As evident from the attendees at the presidential inauguration event (Figure 4.37b), the distribution of crowd in a contained space is uneven. Some areas are packed more than others along this space. This is the clustering phenomenon. It can occur in both static and dynamic crowd behavior scenarios.

Both factors complicate precise definition and quantification of the term *crowd density*. We provide our approach and methodology in the next two sub-sections, limiting the scope to operational and test scenarios of interest detailed in this report.

4.8.1. Various Measures of Density

Crowd density is a measure of people (humans) present in a well-defined and contained planar space. In a multi-story building the fact that crowds exist in a different floor is irrelevant to people on the same floor because density is the opposite metric of free or personal space. If people in a different floor or same floor but different room do not impact a person's personal space, then they are irrelevant and thus are not accounted for in density measure. Hence, both containment and planarity of space are critical constraints in defining which type of people are accepted for head counting in density measure. Furthermore, the physical space itself must be well-defined; that is, we must state the geometry (i.e. length and width) of the physical space in which the presence of humans is counted. Using this setup we can measure crowd density in various parts of the lawn (Figure 4.37b) by bounding the area of interest. The bounding technique addresses the clustering problem. Likewise, by bounding human flow over a strip of marathon track path, we can measure the number of runners flowing in and out of this area over time. This addresses the dynamics problem. The exact method used to measure crowd size in this bounded area is outside of the scope of this report; it can be computer-vision tracker based, manual head counting or some other means. The only remaining topic to address is the quantitative measure of crowd density.

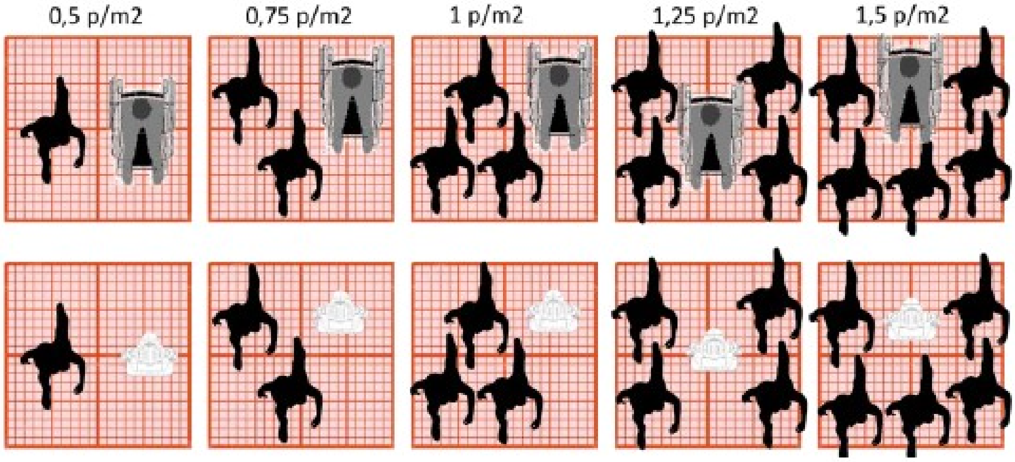


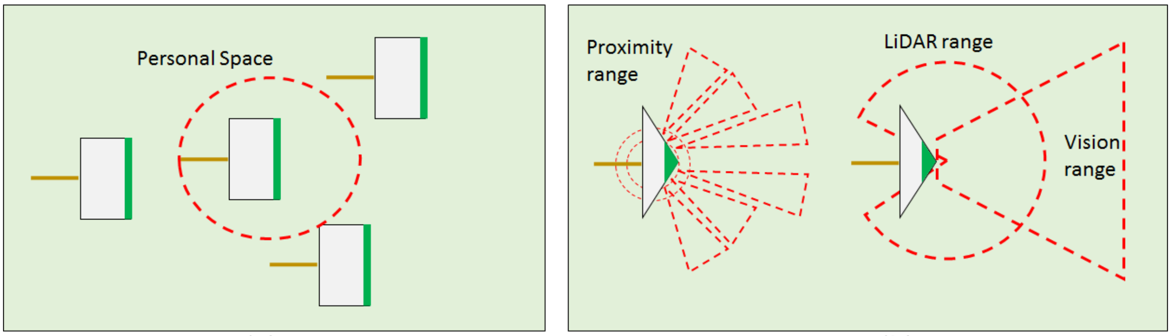
Figure 4.38: Illustrations of different levels of crowd density in terms of people per 2-meter grid [23]

The general rule-of-thumb used frequently in the crowd study community is the number of people in a planar space of 1 meter squared (1m × 1m). Scenarios with different density values are shown in Figure 4.38 [23]. Each display grid is (2m × 2m). In the lowest density case there are two agents in the grid and thus 2 people per 4 meter squared or 0.5p/m2. The top illustrations are portrayed with the smart wheelchair in the center while the bottom ones use a smaller footprint robot such as Pepper. The midpoint 1p/m2 is generally considered as the threshold above which the crowd is labeled "dense."

4.8.2. Personal Space (near-field and far-field spaces)

From our discussions in the previous section, it is obvious that density and personal space are two opposing measures of the distance between an agent and others around it. If the agent of interest is a robot, we must also factor in the limited range and angles of view of its sensing technologies. It is less important (or not meaningful) to characterize and quantify human crowd size at physical locations that cannot be "seen" by a robot. This leads to the concept of personal space for robots. We have already introduced in term "personal space" in regards to social navigation in S7 and an illustration of personal space in Figure 4.36. There we are highlighting a measure of distance separation between a person and its immediate neighbors (Figure 4.39a) from a social context. In contrast, here we are referring to free space around a robot such that collision-free maneuvering and short-term navigation are possible (Figure 4.39b). Equivalently, we can interpret the personal space of a robot as its safety buffer zone. The exact dimension of this zone is highly dependent on a specific robot platform, its onboard sensors, its locomotion profile and its safety margins for contact and collision.

To complete this discussion on personal space for a robot, we introduce two additional classifiers: near-field and far-field spaces. The near-field personal space refers to the immediate region surrounding a robot for which higher accuracy or fuller knowledge of its environment is achievable. In the far-field space the robot either has less accuracy, less knowledge or switches to a different operating mode. Note that there is not a single radius or distance threshold at which the robot's perception switches from near- to far-field. In fact, this threshold value is specific to a type or class of sensor. For example, a spinning LiDAR has a wide field of view in detecting obstacles whereas a vision-sensor camera has a narrower field of view, generally on the front side. Hence, a robot equipped with a LiDAR and a vision camera has a near-field space on the front side where both sensor data set can be fused for higher accuracy for proximity estimation.



(a) (b)

Figure 4.39: (a) Personal Space of a human vs. (b) near- and far-field spaces of a robot

To summarize, the following crowd density definitions and quantifiers are critical to scenario developments and requirements tracing:

- *Density of Human Flow*: In all our scenarios (except for those in social navigation S7), humans are in motion (with possibly temporary pausing). Therefore crowd density is measured in terms of humans flowing in and out of a bounded, well-defined planar floor space (see Figure 4.40 left). Immobile (i.e. stationary) humans are treated as static objects/obstacles for navigation purposes and thus are not counted as crowd units. Of course, in a real-world scenario we cannot predict who will remain immobile (for a long time) and who is pausing to resume walking shortly. In contrast, in our controlled test environment, all humans are part of the crowd, all are counted and each is assigned a unique motion profile. Humans are not used as static objects or obstacles in our tests.

  In certain scenarios with congestion/bottleneck or dense and clustered crowds, it is possible that flow-in and flow-out rates may differ with respect to which bounded region is used and at what time the crowd density is measured (see Figure 4.40 right). The key takeaway is in a number of scenarios, crowd density is dynamic and must be measured and expressed as function of time and region of interest.
- *Near-Field and Far-Field Spaces*: As crowd density increases around a robot, its navigational operations may fail due to next-step confusion and ambiguity, sensor data errors or some other form of malfunction. A well-documented behavior is the freeze. Whenever applicable in all our proposed scenarios, we plan to measure the threshold between near-field and far-field spaces that trigger robot malfunction or poor execution.

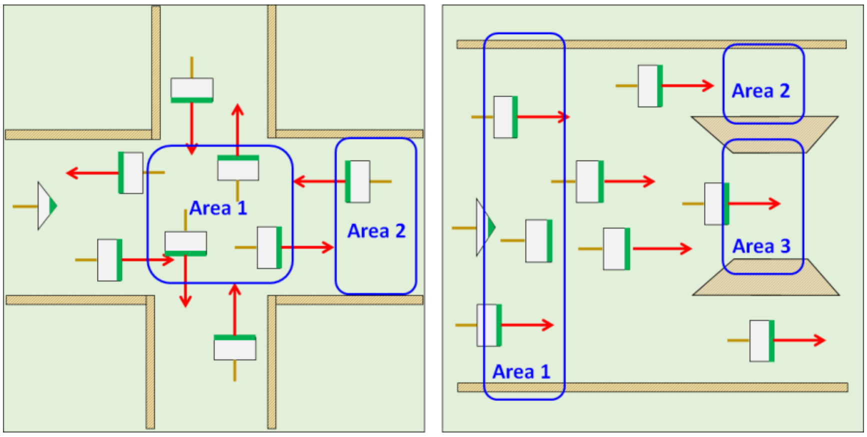


Figure 4.40: Region/area of interest and human flow for crowd density measurement

5. Specification of Requirements

This section serves the main theme of this document: specification of requirements for all test scenarios that are planned for the Crowdbot project. Note that both test scenarios (Section 4) and requirements list (Table 5.1) are not set in stone. The expected date for the first phase of testing is Dec. 2019. Any revisions to test scenarios will be made (with justifications provided) by the Test & Evaluation team, before this date based on recommendations from the Technology & Design teams and from feedback from stakeholders. See Section 6 for a complete description of the scenario update process.

As noted in Section 4, there are seven main test scenarios among which some have additional sub-scenarios for further testing. Each requirement is notated using the convention R x.y for easy referencing (x is the main scenario ID). For example, R 4.2 is the 2nd requirement item from the 4th main scenario. Additional details associated with each requirement item are:

a. *Robot*: Humanoid (H), Wheelchair (W), Service (S) or All (A)
b. *Metric*: see Section 5.1 for complete details
c. *Venue*: Physical (P), Simulation (S) or both (B)
d. *Type*: Mandatory (M) or Optional (O)

Some other constraints for testing and requirements fulfillment:

1. Since the Crowdbot team has access to three different types of robots (say, A, B and C), each requirement item must state which platform(s) it applies to. If the main scenario is applicable to robot type A only, then its derived requirements (sub-scenarios and requirement items) are not applicable to other platforms B or C.
2. The physical test venue is likely to be indoor such as a partner's laboratory facility, a gymnasium, sporting facility or some other large room or warehouse. Office hallways and corridors can also be used to test 1D and 2D flow scenarios. Outdoor testing is optional and limited to the wheelchair.
3. Physical tests are carried out under the leadership of Work Package 1 and 5 (WP1 & 5). Test cases are strictly limited to availability of human and robot resources, access to a physical venue and regulatory clearance from local and national authorities. In contrast, computer-based simulation tests have no such restrictions except for the team's ability to complete

and run high-fidelity simulation models and tools. Simulation based tests are conducted by the leadership of Work Package 4 (WP4).

4. Since test cases may be conducted at different partner sites, the same test setup (physical space, size and type of humans and object clutter) at one test site may differ from another, resulting in different test outcomes. This issue in test data discrepancy, if it exists, will be investigated, recorded and clarified in the Test Report (TR).
5. If a requirement is labeled as "mandatory", then the test scenario associated with this requirement must be completed, corresponding test results must be published and shared with stakeholders. In case a mandatory requirement is not fulfilled, the Test Report (TR) must provide an explanation.
6. Optional requirements are listed as "to do" items if there is additional time for completion or the scenario is an advanced extension of a mandatory requirement. If completed, optional requirements and their test outcomes are included in the test report. If not, they are simply omitted from the Test Report (TR).

An example of a requirement item to be populated in Table 5.1:

Item No.	Description	Robot	Metric	Venue	Type
R 0.1	Robot must move on wheels.	A (All)	B	Both (B)	Mandatory (M)

5.1 Performance Metrics

Each requirement item is labeled exclusively as one of the following three metric types:

1. <u>Success Threshold (Quantitative or Type A)</u>: Whether a requirement from a test scenario is met or not is assessed by comparing a measured quantity to a success threshold. For example, the execution time to reach from Point A to Point B is measured in seconds. If a robot achieves the task under a pre-defined total time elapse value, then this requirement is met. In most cases, due to the lack of prior experimental data, the requirement item may not explicitly state the threshold value but it is left to the Technology and Test & Evaluation teams to propose an acceptable threshold before actual tests commence.
2. <u>Success Threshold (Qualitative of Type B)</u>: In certain test scenarios, it may not be possible to measure a robot's performance quantitatively or with a meaningful metric. For example, in the same example as above, in moving from Point A to Point B, a robot may traverse via many different paths with no clear winning path. However, if a robot maneuvers in a motion profile deemed to be unsafe or unpredictable, then this requirement is unmet. Typical metric values are Yes/No, Pass/Fail or a numeric scale (1 to 10).
3. <u>Discovery (Measured Data & Assessment or Type C)</u>: Since Crowdbot is exploring new robotic features and test scenarios that have never been documented nor replicated, in certain cases, we do not have a baseline to gauge the success level of a test outcome. For such test scenarios, we simply report our findings (test set up, collected data, analysis and assessment) and they serve as discoveries/outcomes that meet this requirement.

All requirement items are listed in Table 5.1. Based on test results and recommendations from external stakeholders, its content will likely be revised in *D 1.3: Requirements Update*.

Table 5.1: Test Scenario Requirements List

Item No.	Description	Robot	Metric	Venue	Type
R 0.1	Robot must move on wheels.	A	B	P	M
R 0.2	Robot navigation test is conducted with indoor ambient (natural or artificial electric) lighting.	A	B	P	M
R 0.3	Wheelchair navigation test is conducted with outdoor ambient natural lighting.	W	B	P	O
R 0.4	Wheelchair navigation test is conducted on rough terrain.	W	B	P	O
R 0.5	Applicability and adaptability of a technology profile used for navigation to another robot in the same class (humanoid, service or wheelchair) must be presented.	A	A	P	O
R 0.6	Human-Machine Interface device and communication method used for the robot must be specified.	A	A	P	M
R 0.7	Two-way wireless communication method, if used, must be specified.	A	A	P	M
R 0.8	Robot must be equipped with a kill switch that deactivates and stops its forward/backward/sideway motion.	A	B	P	M
R 0.9	Structural augmentation of robot must be specified.	A	A	P	M
R 0.10	Electrical/electronic augmentation of robot must be specified.	A	A	P	M
R 0.11	Additional computing resources used must be specified.	A	A	P	M
R 0.11	Changes to dimensions, weight and locomotion of robot, if any, must be specified.	A	A	P	M
R 0.12	Removal or disabling of embedded sensors, processors and electronic devices or modules must be specified.	A	A	P	M
R 0.13	Power supply and energy source modifications to accommodate system augmentation must be specified.	A	A	P	M
R 1.1	Robot follows humans in 1D flow in a straight line path; all agents have the same speed; no over-taking	A	B	B	M
R 1.2	Robot follows humans in 1D flow in a straight line path; all agents have the same speed but a different value from a prior test; no over-taking	A	B	B	O
R 1.3	Robot follows humans in 1D flow in a straight line path; Some human agents have different but constant speeds; over-taking may occur.	A	B	B	M
R 1.4	Robot follows humans in 1D flow in a straight line path; Human agents increase acceleration and robot must adapt its motion profile to continue following; no over-taking	A	B	B	M
R 1.5	Robot follows humans in 1D flow in a straight line path; Human agents decrease acceleration and robot must adapt its motion profile to continue following; no over-taking	A	B	B	M
R 1.6	Robot follows humans in 1D flow in a non-straight curve line path; all agents have the same speed; no over-taking	A	B	B	O

R 1.7	Robot follows humans in 1D flow in a straight line path. A human agent that the robot is following over-takes another human. The robot then follows another human.	A	B	B	M
R 1.8	Robot follows humans in 1D flow in a straight line path. A human agent behind the robot over-takes the robot.	A	B	B	M
R 1.9	Robot follows humans in 1D flow in a straight line path. The robot over-takes the human that is in front of the robot.	S, W	B	B	M
R 1.10	Robot follows humans in 1D flow in a straight line path. Flow congestion is induced by adding human agents to flow.	A	B	B	M
R 1.11	Robot follows humans in 1D flow in a straight line path. Flow congestion is induced by narrowing the passage.	A	B	B	M
R 1.12	Robot follows humans in 1D flow in a straight line path. Flow congestion is induced by having all agents exit a door at the end of passage.	A	B	B	M
R 1.13	Robot follows humans in 1D flow in a straight line path; obstacles (physical objects) are added along the passage. All agents have the same speed; no over-taking	A	B	B	M
R 1.14	Robot follows humans in 1D flow in a non-straight curve line path; obstacles (physical objects) are added along the passage. All agents have the same speed; no over-taking	A	B	B	O
R 2.1	Robot follows humans in 1D flow in a straight line path; another 1D flow in the opposite direction also exists; all agents have the same speed; no over-taking; opposite flow only on one side of the robot.	A	B	B	M
R 2.2	Robot follows humans in 1D flow in a straight line path; other 1D flows in the opposite direction also exists; all agents have the same speed; no over-taking; opposite flows on both sides of the robot.	A	B	B	M
R 2.3	Robot follows humans in 1D flow in a straight line path; obstacles are added along the passage; other1D flows in the opposite direction also exists; all agents have the same speed; no over-taking; opposite flows on both sides of the robot.	A	B	B	M
R 2.4	Robot follows humans in 1D flow in a straight line path; another 1D flow in the opposite direction also exists; over-taking in the opposite flow is induced; opposite flow only on one side of the robot.	A	B	B	M
R 2.5	Robot follows humans in 1D flow in a straight line path; another 1D flow in the opposite direction also exists; over-taking of the robot by a human in the same flow is induced; opposite flow only on one side of the robot.	A	B	B	M
R 3.1	Robot marches forward in a 1D x 1D, 90 degree cross flow. All human agents also march forward using social norm.	A	B	B	M

R 3.2	Robot marches forward in a 1D x 1D, 90 degree cross flow. Some human agents march forward using social norm and some make left or right hand turns in front of robot.	A	B	B	O
R 3.3	Robot marches forward in a 1D x 1D, 90 degree cross flow. All human agents also march forward using social norm. Cross flow human density is high such that the forward path is partially occluded.	A	B	B	O
R 3.4	Robot marches forward in a 1D x 1D, 90 degree cross flow. All human agents also march forward using social norm. Cross flow human density is high such that the forward path is fully occluded.	A	B	B	O
R 3.5	Robot marches forward in a 2D x 2D, 90 degree cross flow. All human agents also march forward using social norm.	A	B	B	O
R 3.6	Robot marches forward in a 2D x 2D, 90 degree cross flow. Some human agents march forward using social norm and some make left or right hand turns in front of robot.	A	B	B	O
R 3.7	Robot marches forward in a 2D x 2D, 90 degree cross flow. All human agents also march forward using social norm. Cross flow human density is high such that the forward path is partially occluded.	A	B	B	O
R 3.8	Robot marches forward in a 2D x 2D, 90 degree cross flow. All human agents also march forward using social norm. Cross flow human density is high such that the forward path is fully occluded.	A	B	B	O
R 3.9	Robot marches forward in Shibuya 2D x 2D 90 degree cross flow. All human agents also march forward without social norm.	A	B	B	O
R 3.10	Robot marches forward in Shibuya 2D x 2D, 90 degree cross flow. All human agents also march forward without social norm. Cross flow human density is high such that the forward path is partially occluded.	A	B	B	O
R 3.11	Robot marches forward in Shibuya 2D x 2D, 90 degree cross flow. All human agents also march forward without social norm. Cross flow human density is high such that the forward path is fully occluded.	A	B	B	O
R 3.12	Robot marches forward in Shibuya 2D x 2D 60 degree cross flow. All human agents also march forward using social norm.	A	B	B	O
R 3.13	Robot marches forward in Shibuya 2D x 2D, 60 degree cross flow. All human agents also march forward using social norm. Cross flow human density is high such that the forward path is partially occluded.	A	B	B	O
R 3.14	Robot marches forward in Shibuya 2D x 2D, 60 degree cross flow. All human agents also march forward using social norm. Cross flow human density is high such that the forward path is fully occluded.	A	B	B	O

R 3.15	Robot marches forward in Shibuya 2D x 2D 60 degree cross flow. All human agents also march forward without social norm.	A	B	B	O
R 3.16	Robot marches forward in Shibuya 2D x 2D, 60 degree cross flow. All human agents also march forward without social norm. Cross flow human density is high such that the forward path is partially occluded.	A	B	B	O
R 3.17	Robot marches forward in Shibuya 2D x 2D, 60 degree cross flow. All human agents also march forward without social norm. Cross flow human density is high such that the forward path is fully occluded.	A	B	B	O
R 4.1	Robot make a left or right hand turn at the junction of a 1D x 1D 90 degree cross flow. All human agents also march forward using social norm.	A	B	B	M
R 4.2	Robot make a left or right hand turn at the junction of a 1D x 2D 90 degree cross flow. Cross flow traffic is 2D. All human agents also march forward using social norm.	A	B	B	M
R 4.3	Robot make a left or right hand turn at the junction of a 2D x 2D 90 degree cross flow. All human agents also march forward using social norm.	A	B	B	O
R 4.4	Robot follows humans along a curved corridor in 1D flow. All agents march at the same speed.	A	B	B	M
R 4.5	Robot first follows and then leaves 1D flow at a specified angle and marches forward.	A	B	B	M
R 4.6	Robot first follows and then leaves 2D flow at a specified angle and marches forward. Robot must cross over incoming flow from opposite direction.	A	B	B	O
R 4.7	Robot joins 1D flow from initial rest position.	A	B	B	M
R 4.8	Robot leaves 1D flow from initial following motion and then stops at any location outside of flow area.	A	B	B	M
R 4.9	Robot leaves 1D flow from initial following motion and then stops at pre-defined location outside of flow area.	A	B	B	O
R 5.1	Robot squeezes between two boundaries without colliding either of them.	A	A	P	M
R 5.2	Robot makes tight 90 degree left or right turn without colliding boundaries.	A	A	P	M
R 5.3	Robots makes 180 degree U turn without colliding boundaries.	A	A	P	M
R 5.4	Robots makes 180 degree U turn in a corridor without colliding boundaries and using forward motion only.	A	A	P	M
R 5.5	Robot maneuvers over or around a protruding obstacle on the ground plane.	A	A	P	M

R 5.6	Robot maneuvers over or around a small indented obstacle on a ground plane.	A	A	P	M
	Robot maneuvers around a large indented obstacle (a ditch) on a ground plane.	A	A	P	M
R 5.7	Robot maneuvers around a thin pole obstacle on the ground plane.	A	A	P	M
R 5.8	Robot maneuvers around a thick pole obstacle on the ground plane.	A	A	P	M
R 5.9	Robot maneuvers around a glass object on the ground plane or on the side as a wall or partition.	A	A	P	M
R 5.10	Robot identifies and stops at a frontal cliff edge.	A	B	P	O
R 5.11	Robot identifies and avoids a cliff edge on the side of path.	A	B	P	O
R 5.12	Robot moves along an inclined (uphill) surface.	A	A	P	M
R 5.13	Robot moves along an inclined (uphill) surface and stops at pre-defined localized position.	A	A	P	O
R 5.14	Robot moves along an inclined (downhill) surface.	A	A	P	M
R 5.15	Robot moves along an inclined (downhill) surface and stops at pre-defined localized position.	A	A	P	O
R 5.16	Robot moves along side-inclined surface while maintaining forward march.	A	A	P	M
R 5.17	Robot moves along side-inclined surface and stops at pre-defined localized position.	A	A	P	O
R 6.1	Robot can detect one-time contact with an external object.	A	B	P	M
R 6.2	Robot can detect non-contact with an external object after engaging in one-time or continuous contact.	A	B	P	O
R 6.3	Robot can detect continuous contact with an external object while at rest.	A	B	P	O
R 6.4	Robot can detect continuous contact with an external object while moving.	A	B	P	O
R 6.5	Robot can measure force impact during contact with an external object.	A	A	P	O
R 6.6	Robot can measure pressure exertion after contact with an external object.	A	A	P	O
R 6.7	Robot can measure change in pressure exertion after maneuvering to and from an external object.	A	A	P	O
R 6.8	Robot can detect multiple contacts originating from different directions.	A	B	P	M
R 6.9	Robot can simultaneously detect multiple contacts originating from different directions.	A	B	P	O
R 6.10	Robot can detect and differentiate multiple contacts originating from different directions.	A	B	P	O
R 6.11	Robot can simultaneously detect and differentiate multiple contacts originating from different directions.	A	B	P	O
R 6.12	Any navigation malfunction during test must be reported.	A	A	P	M

R 6.13	A red team is formed to deliberately trigger a malfunction.	A	A	B	O
R 6.14	Robot's margins before falling are measured by injecting force from the side.	A	A	P	O
R 6.15	Robot's margins before falling are measured by injecting force from the front.	A	A	P	O
R 6.16	Robot's margins before falling are measured by injecting force from the back.	A	A	P	O
R 6.17	By tilting the robot from the side (if possible), its force exertion of the external object is measured.	A	A	P	O
R 6.18	By tilting the robot from the front (if possible), its force exertion of the external object is measured.	A	A	P	O
R 6.19	By tilting the robot from the back (if possible), its force exertion of the external object is measured.	A	A	P	O
R 6.20	Robot navigation is assessed after disabling a sensor module via electrical, software kill or physical blind fold.	A	A	P	M
R 6.21	Robot navigation is assessed after disabling a processor module via electrical or software kill.	A	A	P	O
R 6.22	Robot's navigation accuracy is tested after long time-elapsed operation by repeating a squeeze test (through two boundaries) with collision.	A	A	P	O
R 6.23	Robot's navigation accuracy is tested after long time-elapsed operation by repeating a multi-waypoint localization and navigation test.	A	A	P	O
R 6.24	Robot's navigation accuracy is tested after long time-elapsed operation by repeating a relevant test procedure.	A	A	P	O
R 6.25	Robot is deactivated using onboard kill switch while moving.	A	B	P	M
R 6.26	Robot is deactivated using remote kill option if exists.	A	B	P	M
R 6.27	Robot is deactivated using at least two kill switch options.	A	B	P	M
R 6.28	Any abnormal robot behavior such as stall or freeze is reported.	A	A	P	M
R 7.1	Robot passes by stationary and 1D flow moving human and uses audio-visual cues and verbal announcements as needed.	H	B	P	M
R 7.2	Robot passes by stationary and 1D flow moving human and uses audio-visual cues and verbal announcements as needed. If passage is blocked, robot asks for clearance via verbal communication until passage is cleared.	H	B	P	M
R 7.3	Robot passes by stationary and 1D flow moving human and uses audio-visual cues and verbal announcements as needed. If passage is blocked, robot uses its gaze in the direction of humans that are blocking and asks for clearance via verbal communication and also until passage is cleared.	H	B	P	M
R 7.4	Robot passes by stationary and 2D flow moving human and uses audio-visual cues and verbal announcements as needed.	H	B	P	M

R 7.5	Robot passes by stationary and 2D flow moving human and uses audio-visual cues and verbal announcements as needed. If passage is blocked, robot asks for clearance via verbal communication until passage is cleared.	H	B	P	M
R 7.6	Robot passes by stationary and 2D flow moving human and uses audio-visual cues and verbal announcements as needed. If passage is blocked, robot uses its gaze in the direction of humans that are blocking and asks for clearance via verbal communication and also until passage is cleared.	H	B	P	M

5.2 Validation & Verification of Tests

The following order of execution are planned for each round of tests:

1. *System-Level Test Plan (STP):* A month prior to test events, the T&E team in collaboration with other internal teams (Technology Development, System Integration, Design and Quality Control) prepares a test plan document. It provides a list of test cases planned during one- to two-month long test event. Test cases are labeled as:
   Example: **T1.1**: 1D Flow test with all agents moving at same constant speed
   T1.2: 1D Flow with agents moving at different constant speeds and so on…
   Note that specific details of each test case (time length of test, number of repetitions, number of human agents, speed value, etc.) may not be stated in STP since it is a system-level document. Actual values of test parameters used in each test case is recorded by the test execution team. A copy of STP is distributed (prior to the test event) to stakeholders to solicit their feedback and recommendations.
2. *STR Mapping*: Each test case is a representative of a test scenario detailed in Section 4. There may be not a one-to-one equivalence between a test scenario and a test case. A test case may be a subset of a test scenario but it may contain features and constraints not stated or considered in a test scenario. The purpose of STR mapping is to clarify the relationship between a test case **T** x.y, its best-matching test scenario S_m and its associated requirement items R i.j, R m,n, ⋯. The STR map is prepared by the T&E team during or after the test event since it must witness the test cases themselves to understand what are being tested and what are omitted. The STR map and its related information are documented in the main body of the Test Report (TR).
3. *Test Phase & Data Collection*: The test event duration is from one to three months. See Figure 6.1 for timelines. First- and second-round tests are planned for M22—25 and M39—42, respectively. Test cases can be spread across these months based on availability of the test team, a specific robot platform, human participants, test facility and other logistics and physical resources. Test cases can also be spread across different partner locations. STP will provide high-level information of each test case (date, location, test team, etc.) but test details are gathered and documented in the Test Report.
4. *Test Evaluation & Reporting*: Test data evaluation and preparation commences from the end of the test event and lasts exactly one month. A total of two test reports—TR1 and

TR2—will be completed by end of project months M25 and M42, respectively. TRs are also released and distributed as official deliverables D1.4 and D1.5, respectively.

6. Scenario Specification Updates

This report D1.1 is the first of two requirements documents generated by the Test & Evaluation team for the Crowdbot project. Its follow-on D1.3: *Specifications Update* is published after the completion of 1st round physical tests and its accompanying evaluation report (see Figure 6.1 for details of the timelines). This assures that D1.3 takes into account lessons learned from the 1st round tests as well as feedback from stakeholders.

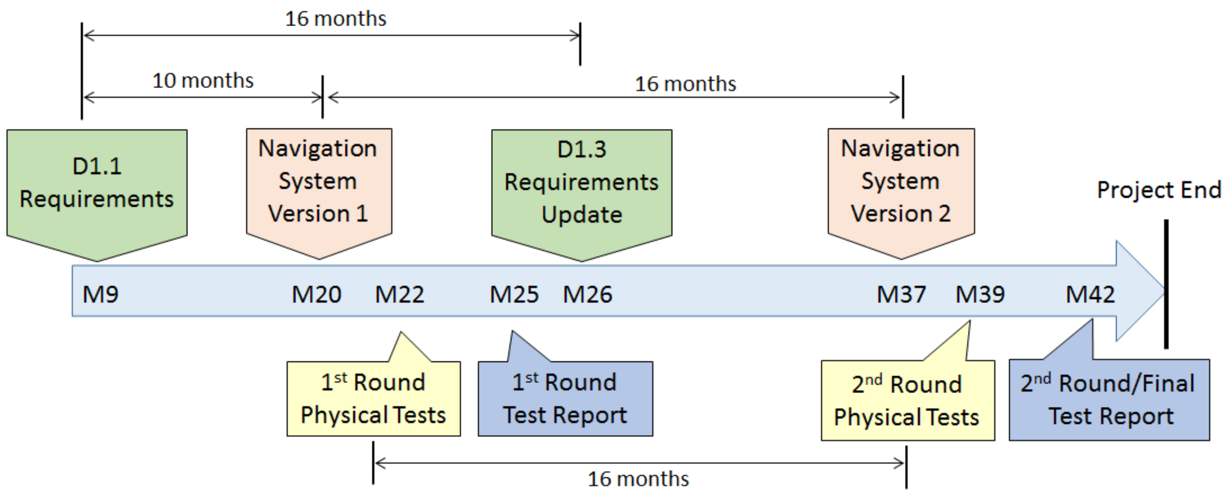


Figure 6.1: Timelines & dependencies among requirements, tests and reports

A one-to-two month time lag is expected from the time of completion of tests to the publication of the test report. The report will cover scenarios tested, applicable requirements, test outcomes, post-processing and assessment by the team, and feedback from stakeholders.

From Figure 6.1, we see that there are two rounds of physical tests with a time gap of roughly sixteen months in between for technology enhancements. Likewise, the time gap between completion dates for 1st and 2nd navigation systems is also sixteen months. The 1st version of the navigation system is expected to be ready by month 20-21 (Aug/Sept 2019), which is roughly ten months from the publication of this report (month 9). The physical test commences roughly two months after the delivery of the navigation system.

Due to the dependency of D1.3 on the completion of physical tests and the evaluation report, any slippage in the delivery of the navigation system and/or a delay in running and completing tests will push the publication date of D1.3 to the right.

6.1 System-Level Test Plan (STP)

Two rounds of physical tests are planned on months 22 and 39 (October 2019 and March 2021) over the 42-month project time span. However, no specific dates are set in the original proposal for the delivery of a test plan. Furthermore, no action item is assigned to a Crowdbot team member as a Work Package task to prepare a test plan. This ambiguity is resolved in this section: the Test & Evaluation (T&E) team will take the lead in the preparation of test plans for both

1st and 2nd round tests. It will work closely with the remaining internal teams (Technology Development, System Integration and Design & Quality Control) teams to script specific tests to be carried out in each round. Since there is a two-month gap between the delivery of the navigation system and the commencement of physical tests, the T&E team will use this time window to prepare the test plan. Specifically, the T&E team will publish the test plan within one month from the delivery date of the navigation system, thus allowing a one-month period for preparation and coordination before the commencement of physical tests.

6.2 Simulation Tests

Another topic not covered in the proposal is the role of computer-simulation based navigation tests in fulfilling scenario requirements. The Work Package 4 (WP4) team is in charge of crowd simulation but this effort may not extend to robotic navigation system simulations. Two deliverables are anticipated for the crowd simulator: month 20 (August 2019) for the intermediate version and month 36 (December 2020) for the final version. No specific dates are assigned for the delivery of a robotic navigation simulator.

The proposed solution by the T&E team is as follows: The developer of the crowd simulator (INRIA) will work closely with other Crowdbot teams (Technology Development, Design & Quality Control and Robot Integration) to jointly develop a robotic navigation system simulator. Since the simulator itself will not undergo the process of independent validation and verification, it cannot be used to fulfill requirement items marked as “Quantitative Success Threshold.” Its contribution is limited to the fulfillment of qualitative success and discovery requirement items only. More importantly, the simulator can be used as an aid in debugging software modules and robot integration efforts before the commencement of physical tests. Further details on the System Level Test Plan (STP) can be found in Section 5.2.

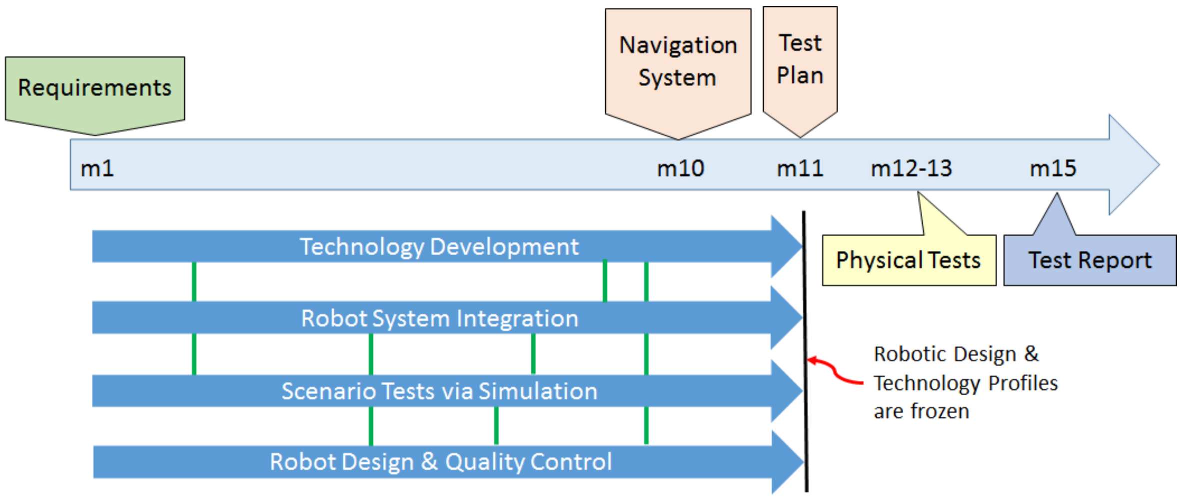


Figure 6.2: Fifteen-month Timeline & Activities for each round of Physical Tests

A summary of timelines and Work Package activities for each round of tests is provided in Figure 6.2. During the 15-month time span from the publication of the requirements document to the commencement of physical tests, parallel efforts are underway among Crowdbot teams and at certain project months, integration takes place. This is illustrated using vertical green lines in Figure 6.2 signifying team-to-team collaboration. Approximately two months before the commencement of tests, the Technology Development team releases its Navigation System (a technical report plus software code). The Test & Evaluation team then prepares a test plan to outlines specific test scenarios and relevant requirement items. The specific robot design and navigation technology profiles are frozen before the preparation of the test plan. Within two months from the completion date of tests, a test report is published.

6.3 Stakeholder Engagements (SHENG)

As detailed in Figure 2.2, there are several types of external stakeholders that the Crowdbot team will continually engaged with and seek feedback regarding project progress and milestone achievements. In this section we limit our scope to the following stakeholders most pertinent to scenario development and test evaluation: general public, governmental and enterprise customers. These stakeholders are collectively known as the *user community* since they all play critical roles realizing in feasibility and implementation of Crowdbot technologies for real-world practical use. An outline of how we use stakeholders' feedback in improving test scenarios and technology enhancements are illustrated in Figure 6.3. Specific details of the process and procedures used for stakeholder engagement is provided in the following sections.

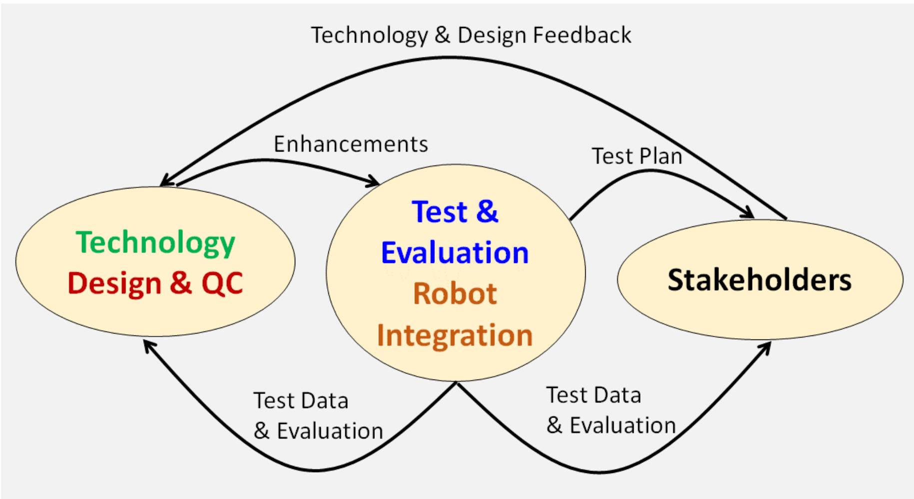


Figure 6.3: Feedback Cycles for Updating Test Scenarios

Involving stakeholders in CROWDBOT is crucial for defining and refining scenario requirements. The CROWDBOT project investigates safe robot navigation in dense crowds. The aim of the stakeholder engagement is to understand the type of interactions between robots (smart wheelchairs or humanoids) and crowds of people in order to define concrete scenarios in which to evaluate and test technologies developed within the project scope. UCL has submitted an ethics approval proposal for the CROWDBOT stakeholder engagement studies.

6.3.1. Recruiting Participants

Participants will be recruited from the general public, wheelchair users, medical practitioners, wheelchair manufactures, assistive technology experts, and building managers. The recruitment process involves sending out invitations for participation in either a structured interview or focus group. The invitations can be written or verbal and a consent form must be signed by each participant prior to participating in any study.

6.3.2. Procedure

This study follows two main procedures, namely the structured interview and focus group study. A structured interview occurs in a public place and a participant is asked a fixed set of questions with his responses video/audio recorded. Each participant is asked exactly the same set of questions for consistency. A structured interview is expected to last, at most, 60 minutes. The interview recordings will be outsourced to an external company for transcription. In the interest of safety and transparency, the researcher will never be alone in a room with a participant. All structured interviews will be conducted in the presence of other people. A focus group involves a group of six to eight participants discussing wheelchair scenarios. The discussion is moderated by the researcher who will also be taking notes during this process. A focus group is expected to last, at most, 90 minutes. Reasonable travel costs will be reimbursed for both the structured interview and focus group studies.

6.3.3. Ethics/Privacy

Every structured interview and focus group will be video/audio recorded, however the participant can choose whether or not he wishes to be video recorded. All audio recordings will be anonymized in transcription and video recordings will not be shared or viewed by anyone outside the ethics approval, unless prior consent is given by the participant. Where consent is given, the video will be used for presentations at conferences or teaching material and will be anonymized. Apart from contact information (name, email address, telephone number, address) and recordings, no other personal information will be collected. Furthermore, contact information of participants will remain in the country where it was collected and will be deleted upon finalising this study.

The anonymized data collected will only be disclosed to the principal investigator, project partners, and researchers involved in this project. The results will be published in journals or conference proceedings.
In summary, stakeholder engagements are planned twice: before 1st and 2nd round tests. Feedback from each engagement will be used to tailor and enhance test cases in each round of test event.

6.4 Technology Enhancements

From Figure 6.2 one can infer that technology enhancements are part of ongoing R&D activities. New features are added after successful collaboration among different sub-teams. Milestones serve as time markers for major achievements in the project but fine-grained enhancements are targeted at monthly or bi-monthly intervals. These are not official deliverables but internal reports/documents and software builds. A major technology upgrade is expected after the release of first-round test reports and the publication of scenario updates D1.3. This includes the use of new sensing and computing devices, development of new test scenarios and modification of navigation algorithms and software code. Enhancements continue up until the commencement of 2nd round tests in month 39. By then the entire team shifts focus to test planning, preparation and execution. All technology enhancements are reported in the first and final test reports.

Figure 6.3 summarizes all entities involved in each round of tests. The Crowdbot team consists of four sub-teams: Technology Development (TD), Robot Design & Quality Control (QC), Test & Evaluation (T&E) and Robot System Integration (SI). The external entities are various stakeholders (see Section 2.1 and Figure 2.2 for additional details). The two combined teams (TD and QC) develop technologies applicable to each robot platform. Each version of design and software build is submitted to the SI team where each robot undergoes technology enhancements. Based on current state of the resulting Crowdbot robot features, the T&E team prepares a test plan and lays out relevant test scenarios and applicable requirement items. Beyond internal circulation the test plan is shared with certain stakeholders as a courtesy. After tests are completed, T&E evaluates test data and prepares a report which is shared with all other Crowdbot teams and certain stakeholders. Feedback from stakeholders is incorporated into further technology enhancements. The above process occurs in two cycles, once for each physical test.

References

[1] Starship Technologies, https://www.starship.xyz/

[2] Domino's Pizza delivery robot, https://qsrmedia.co.uk/technology/news/dominos-and-starship-launch-eu-pilot-program-autonomous-robots

[3] https://www.seattletimes.com/business/technology/delivery-robots-are-showing-up-on-city-sidewalks/

[4] Emergency waiting area in hospital, http://brazosportregional.org/healthcare_services/emergency_services.aspx

[5] Paris Gare du Nord train station: https://parisbytrain.com/wp-content/uploads/2013/02/gare_du_nord_platforms_eurostar_thalys_optimized.jpg

[6] Crowdbot Project D6.1: *Overview of Risks when using Robots in Crowds*, available in Dec. 2018.

[7] ANYmal Robot by ETH Zurich, http://www.rsl.ethz.ch/robots-media/anymal.html

[8] Savioke Robot, http://www.savioke.com/

[9] Savioke Relay robot for hotel room service, https://wtvox.com/robotics/relay-robot/

[10] TUG Robot, https://aethon.com/

[11] TUG Robot for luggage delivery, https://ktla.com/2018/02/19/luggage-robots-sheraton-hotel-san-gabriel/

[12] Quickie Electric Wheelchair, https://www.quickie-wheelchairs.com/

[13] SoftBank Robotics, https://www.softbankrobotics.com/emea/en/robots/pepper

[14] Willow Garage PR2 Robot, http://www.willowgarage.com/pages/pr2/overview

[15] Crowdbot Project D2.1: *Sensor Specifications*.

[16] Crowdbot Project D5.1: *System Integration*.

[17] Cross flows at Shibuya traffic junction: https://www.blogto.com/city/2007/10/could_we_walk_across_bay_and_bloor_diagonally/

[18] Tracking of large dense crowds: http://crcv.ucf.edu/projects/crowd/

[19] Presidential Inauguration Crowd, https://www.pbs.org/ image taken from live video coverage.

[20] Helbing, Dirk, Anders Johansson, and Habib Zein Al-Abideen. *"Dynamics of crowd disasters: An empirical study."* Physical review E 75.4 (2007): 046109.

[21] Plaue, Matthias, et al. *"Trajectory extraction and density analysis of intersecting pedestrian flows from video recordings."* Photogrammetric image analysis. Springer, Berlin, Heidelberg, 2011. 285-296.

[22] Liddle, Jack, Armin Seyfried, and Bernhard Steffen. *"Analysis of bottleneck motion using Voronoi diagrams."* Pedestrian and evacuation dynamics. Springer, Boston, MA, 2011. 833-836.

[23] *Safe Robot Navigation in Dense Crowds*, CROWDBOT Project, EU Horizon 2020 proposal to European Commission, H2020 ICT-25 2017 RIA.

[24] A. L. Barriuso et al, "*Agent-based Intelligent Interface for Wheelchair Movement Control*", Sensors, May 2018, vol. 18, 1511.

[25] Sunrise Medical Power Wheelchairs and Accessories; http://www.sunrisemedical.com/

[26] United States Department of Labor, Occupational Safety & Health Administration, https://www.osha.gov/law-regs.html

[27] Sheridan, T. B., & Verplank, W. L. (1978). *Human and computer control of undersea teleoperators*. Massachusetts Inst. of Tech., Cambridge Man-Machine Systems Lab.

[28] Groceries delivery via robots, https://bgr.com/2017/01/31/robot-delivery-pizza-starship-technologies/

[29] MediaMarkt package delivery robot, http://www.mediamarktsaturn.com/en/press/press-releases/pilot-project-launched-media-markt-test-delivery-customer-delivery-robot

[30] Domini's Pizza Robot crossing tram track in the Netherlands, https://www.nu.nl/119076/video/dominos-test-pizzabezorging-met-robot-in-amsterdam.html

[31] World Health Organization, *Fact Sheet on Wheelchairs*; Geneva, Switzerland, 2010.

[32] United Nations. *Standard Rules on the Equalization of Opportunities for Persons with Disabilities*; San Francisco, CA, USA, 1994.

[33] National Transportation Safety Board, https://www.ntsb.gov/_layouts/ntsb.recsearch/RecTabs.aspx

[34] Federal Aviation Authority, https://www.faa.gov/regulations_policies/

[35] Cuybot robot, Locomotec; http://www.locomotec.com/en/