Chapter 5

C^3Arc: cBDI Based Cognitive Collaborative Control

> "Being in a band is always a compromise. Provided that the balance is good, what you lose in compromise, you gain by collaboration."
>
> Mike Rutherford

Contents

5.1 Introduction

Let us again go back to the example of Chapter 1. In that example Y is an equal collaborator. Imagine Y is an elderly person who helps you to move the table from A to B. Even though it looks like Y helps you to move the table, as Y is an elderly person so you have to take initiative and help in the task of moving the table. As in the introductory quote you are on the compromising side. Going back to the example of Chapter 1 with yet another assumption is to give the reader an impression that to be in collaboration, sometimes requires compromise.

In the context of collaboration between IW and its user, we cannot ignore the fact that human users are the agents that acts independently of the robot. As discussed in section 2.4.1of Chapter 2, in the current approach of wheelchair collaboration (control), human is always on the compromising side. We go with the view of [219]:

> .. *it is necessary that Human should no longer be on the compromising side. Robot should "equally" be responsible for any compromise, whether it is to sacrifice the shortest path to respect social norms or to negotiate the social norms for physical comfort of the person or to provide the human with the latitude in the way in the way he / she wants to be guided.*
>
> Pandey [[219], p.94]

This statement highlights responsibility of a robot. Pandey's context of the statement is not immediately applicable in our case. However his view on robot responsibility is inspiring for us. In this chapter a frame work is presented, which takes in to account how user wants to control an IW. The framework presented in this chapter is an attempt to control a wheelchair in a way the user wants to control. The cognitive enhanced control for wheelchair navigation is an attempt for *collaborative navigation* between navigation controller of an wheelchair and human user.

The framework presented in this chapter aims to apply the cBDI architecture in *collaborative navigation.* An empirical investigation of human wayfinding strategies is presented in Chapter 4. The facts of strategies learned from empirical investigation are stored as knowledge base for cBDI architecture. The architecture aims to establish cognitive collaborative control for an intelligent wheelchair. Cognitive collaborative control is the term given to the process where (navigation controller of) IW maintain collaboration with its human user by inferring user strategy and execution of control action in *help when needed* scenario.

5.2 Collaborative Navigation–What is it?

To give an idea of how concept of *collaborative navigation* is to be understood, here is an example:

Imagine Era, an elderly woman, lives with her family in an apartment. She is an IW user who can move freely from one location to another location. Imagine Era is currently in the living room. Suddenly Era start feeling tired and she feel that going to her bed room is a good idea. So she assigned an action "Move to b (from a) " to the wheelchair. Era does not move around randomly and so associated commands to IW are not random. Instead Eras' planning component takes the control and her planner devises a plan and so she gives control commands to the wheelchair.

Let's look at this scenario through the eyes of an intelligent wheelchair: Imagine instead that you are an intelligent wheelchair and you are assigned an action "Move to b (from a) ". You are equipped with intelligent sensors. When defining the task "Move to b " as an action that allows you to go to a final position starting from its initial position it is considered to be your individual action. When Era defines the task "Move to b", for you it is participatory action [220] rather than an individual action because your objective is to collaborate with Era. You are said to be collaborative with Era if you can initiate a set of control actions on her behalf in some situations or if you can correct Era's "erroneous" action. For you those actions are erroneous actions–where Era continually select and does inappropriate action under a given situation. In those cases, you apply control to prohibit Era to continue perform the inappropriate action. Most importantly, your task is to make her feel that she is controlling your actions. Your task is to intervene in Era's actions when necessary in a way that she does not even realizes that she is getting help.

Given the objective and additional safety rules that an IW can generally have, goal of this chapter is to formalize a navigation controller for an IW such that:

1. IW follows its user whenever possible
2. and in a situation where first condition may not be achievable, take the control from user.

Let us again go back to the example above: If Era's decision and its associated directive to you are correct, and then she can obtain a result that matches her goal and the situation at the time. In reality, however, due to her physical and cognitive illness, she can fail to give a proper directive to you in several ways. One of such cases may be where Era's understanding of a given situation (and thus her decision) is not correct for some reasons, such as inattention or internal distraction. Another case may be where due to her health condition she is not able to give clear directives such as "turn right" to you. In these (failure) scenarios, you need to initiate a set of actions on her behalf.

So for a controller to know when to help and initiate with control actions, controller needs to know user intent. Apart from that controller must be adaptable to its user so controller needs to have a reasonable understanding of its user. For user, there are different way to reach her goal which is based on human strategy. So a key concept for the controller is its knowledge of human centric strategies. Some time it may be possible that user and navigation controller will attempt to bring their preference. So negotiation is one of the requirements for controller of IW.

Collaborative navigation can be seen as where both control entities (user and IW controller) accomplish navigation task; and controller take initiative in situations where help is required. In collaborative navigation, the navigation controller and human are part of the same system that interacts with the environment as a single entity.

“Help when needed”– is also addressed by the work of [221]. They proposed a control architecture called ACHRIN. ACHRIN is devised to enable human user at all levels, ranging from performing set of actions for the robot to performing low level navigation. The plan that is generated by the controller considers human help for example: open a close door. Our approach is different from their approach in that the controller does not plan actions for human and only take control from the human user during different (failure) scenarios.

5.3 The Control Architecture for Collaborative Navigation

Before formalizing the control architecture certain basic requirement are considered. Focus is to provide means in order to achieve collaborative navigation. In the following subsection, we discuss components for collaborative control architecture. Suggested components are based on ideas on human-machine interaction, existing literature of collaborative control for wheelchair as well on collaborative system literature.

5.3.1 The Basic Requirements

This subsection, aims to reason about the basic requirements of the core module to establish collaboration between user and the system.

5.3.1.1 User intention predictor

In context of human-machine interaction, in [21], it is suggested that collaboration between robots and humans can be greatly improved if the robots can predict the actions of the humans. As mentioned in 2.4.1, some works contribute to user intention estimation for wheelchair shared control [136] [7] [138], [139]. So there must be some component that interprets user intentions.

5.3.1.2 User model: mechanism for adaptation

Knowing human user intention cannot make sense unless the system has a reasonable understanding of the user. In agreement with Fong [4], it is assumed that system should recognize that different users have different abilities. Any intelligent wheelchairs collaborative control can be seen as system integrating users with varied experience, skills, and training. As a consequence, the system must have mechanism for adaptation to different users. To collaborate with user of different physical and cognitive abilities, mechanisms should be available to gauge user abilities. A user model (or profile) can be seen as a set of attributes which describe a user or a group of users.

5.3.1.3 Human centric strategy library

It is quite reasonable to assume that the overall task goal, e.g. the goal position is known to both partners i.e. user and the controller. For human user, several action plans can exist to reach a goal and more likely based on the human's strategy. Therefore, a key feature (concept) of the controller is the knowledge of human centric strategy.

5.3.1.4 Negotiation

The possibility to integrate the individual intentions is a prerequisite for collective action between human and system. Under such a context it is our understanding that negotiation is a prime requirement for controller of an IW. In agreement with [4], to collaborate controller must permit negotiation on control. Negotiation is required in case the preference of individual intentions differ.

5.3.1.5 Mental Model

The design philosophy behind the proposed architecture is to provide means to collaborate as humans do, to act jointly with a human counterpart. An extension to the basic BDI architecture is based on the belief that shared plan is particularly crucial requirement to be meet for human system collaboration.

cBDI presented in Chapter 3 is extended BDI with these concepts. The cBDI agent together with strategic planner captures cognitive concept into the control architecture.

5.3.2 Underlying Principles

Robot control architecture contributed the idea that the controller should be reactive as well as deliberative. The following points are considered as the key principles for the control architecture:

- Facilitate human participation in control. User participation in the wheelchair navigation control, demands a user interface that enables human user to communicate with the IW.
- Facilitate decision capabilities in control. To facilitate deliberation in control, there must be a deliberative module to infer or recognize ongoing human activity as well as helping the IW to decide to intervene in these activities if needed.
- Facilitate reactive behaviour in control. To facilitate low level control, there must be a low level control module to provide access to incoming raw data of all sensors as well as outgoing commands to all actuators.

5.4 The C^3Arc: The Architecture

High level control architecture dedicated to *Cognitive Collaborative Control* is discussed in this section. Proposed architecture is based on the requirements investigated in the last section. As shown in Figure 5-1, it is a hybrid three layered cognitive agent based control architecture. Section 5.5 contains a through description of each layer. Before that, this section intends to reason the following four question of the control architecture.

Why it can be categorized as Hybrid?

As discuss in the section 2.3.1, while controllers cover a wide range of designs and techniques, majority can fit into one of the four basic types of control architectures. It is well accepted that entirely reactive as well as deliberative paradigm to robot control architecture is inadequate. Hybrid schemes are well accepted as robot control architecture [222]. Two widely known examples of hybrid architectures are InterRAP[223] and Touring Machine[224]. The frame work developed here follows hybrid paradigm–characterized by a layering of capabilities, where low level layers provide reactive capabilities, and high level layers provide the more computationally intensive deliberative capabilities.

Why Layered architecture?

The frame work developed here is following a layering approach. Three Layered Architecture is presented because the most popular variant on hybrid architectures is layered architectures.

Why Agent based?

We need to reason about human mental attitudes of belief, desire, and intention so that it can perform collaborative action. In order to allow control decisions that are more autonomous, reactive, proactive, and social one widely accepted approach is agent-based architecture. Work on agent based system contributes the idea that the agent based control architecture should be able to function smoothly and productively as an integrated team. We use the cBDI agent. It is an extended BDI architecture for human agent collaboration. BDI approach underline understanding of the dynamics of belief, desire, and intention and inter dependencies which is fundamental in achieving rational behaviour.

Why the term “Cognitive Collaborative Control”?

In AI and cognitive sciences community, the term “cognitive architectures” is commonly used to designate the organization of systems designed to model the human brain. As our proposed architecture is adoption of BDI architecture- a cognitive architecture, the term cognitive used here to describe our architecture. The architecture is collaborative because it is designed for collaboration based on collaborative BDI architecture.

5.5 Components of the Architecture

5.5.1 User Interface Layer

The User interface layer (UIL) facilitates human participation in collaborative control. The user issues commands for navigation task through this layer to the system. UIL allows /acquires human input (commands, other information such as goals) via a command modality. The operator’s capability is accessed as human index.

User interface layer consists of following components:

User input

The user inputs are control commands to drive the intelligent wheelchair. The user input reflects human operators’ intent in the control. The user gives command through the user interface.

User interface

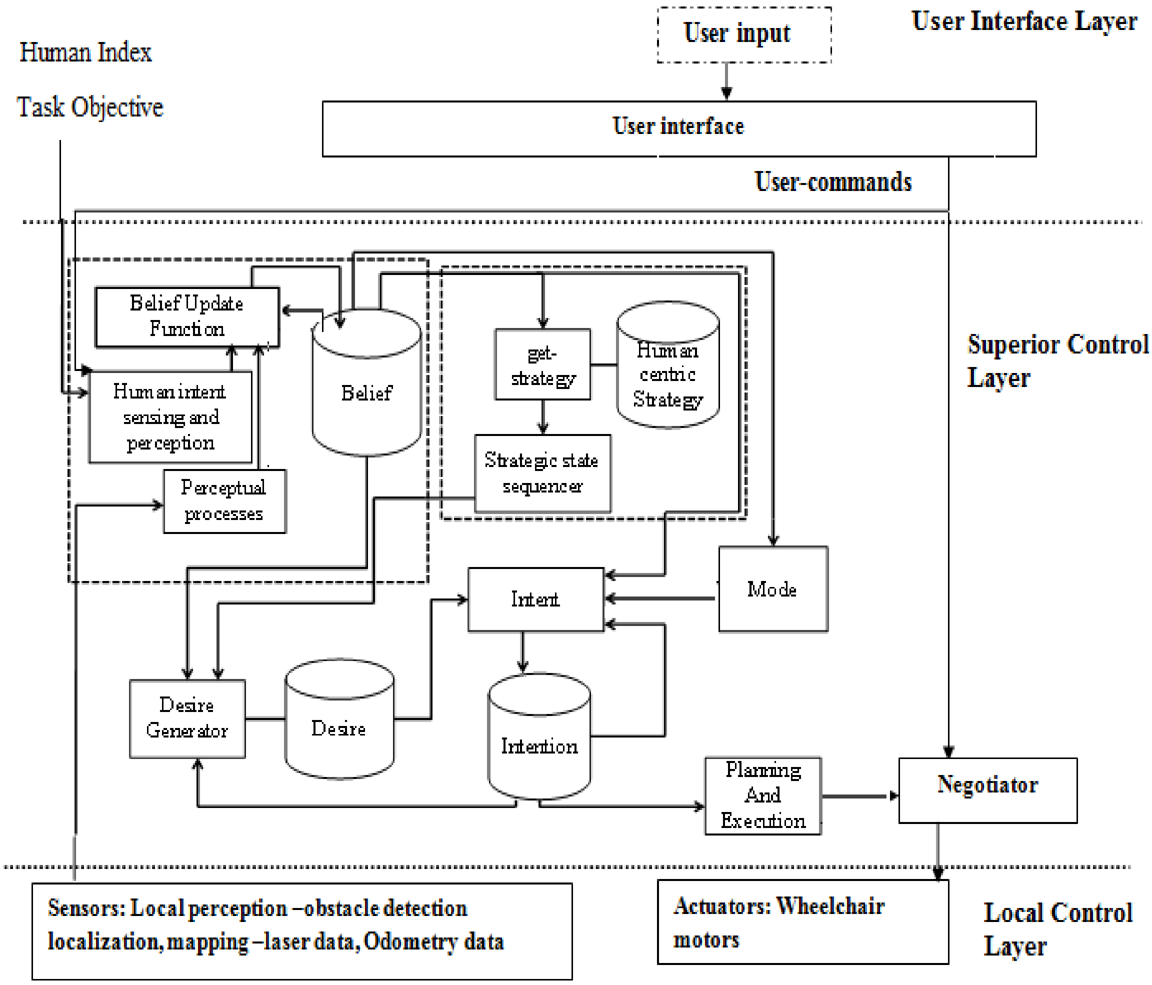


Figure 5-1: The C^3Arc Architecture

The purpose of the User interface is to sense the user input. User interface is the main module for user commanding the robot and the task.

Human index

There are different types of test to quantify user capabilities. We adopt the Folstein test or mini-mental state examination (MMSE) [225]and Instrumental activities of daily living (IADL) [226] test to it. We quantify this measure as human index; which characterizes the capability of the human user. Appendix B have further details.

5.5.2 Superior Control Layer

Superior Control Layer (SCL), also called the “deliberative layer” contains components that provide collaborative decision capabilities to the system. This layer provides the collaborative mechanism. cBDI agent and a negotiator module are the two architectural elements that makes it possible to achieve collaborative plan-

Table 5.1: Requirements and components within SCL

Requirements	Module	Comment
User intention prediction	Human intent	cBDI component
User model	Belief	Belief module of cBDI
Human Centric strategy library	Strategic planner	Strategic planner module of cBDI
Negotiation	Negotiator	
Mental model	cBDI	

ning and negotiation in control.

There are three key elements: the cBDI agent, the strategic planner together with the human centric strategy and negotiator module. In the following, these components of SCL are explained in more detail.

5.5.2.1 cBDI agent

The cBDI agent makes decisions about controller behaviour for current user capabilities. Three kinds of behaviour are possible for the cBDI agent: collaborative initiation behaviour, autonomous behaviour and inactive state. The state transition model of cBDI agent is presented in Figure 5-2. More detailed presentation is in the Table 5.2. The latter represent transfer functions.

Initially the agent is in the idle state i.e. in the inactive mode. Autonomous behaviour state is activated when agent finds measure for user capability is less than a threshold value. In autonomous behaviour, the agent does not require any specification from the user other than the goal. In this mode the agent derive a complete navigation plan.

Collaborative initiation state is activated, when agent recognizes user capability is greater than a threshold value. In this case to complete the navigation task, the agent collaborate with its user. Here cBDI agent is seen as collaborative navigator.

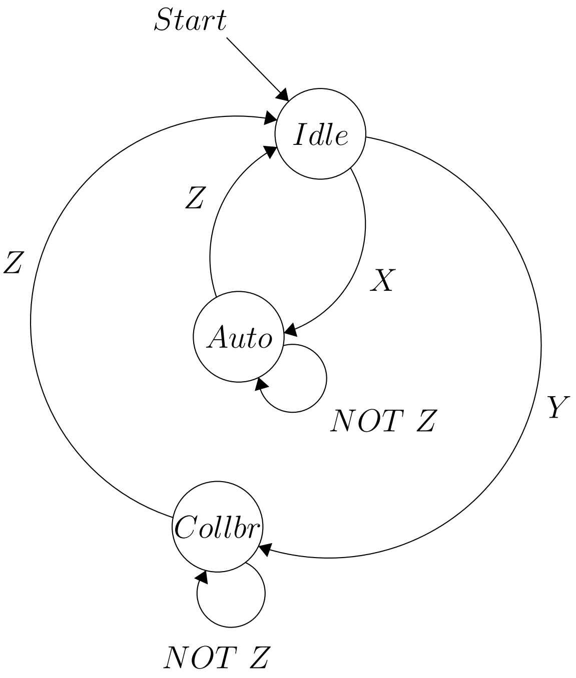


Figure 5-2: State transition model of cBDI agent

States	Process	Transfer function	Next state
inactive (Idle)	Default state ; Do nothing	User capability is less than threshold. X	autonomous
		User capacity is grater than threshold. Y	collaborative
Autonomous (Auto)	Derived a plan and execute	goal reach.Z	inactive
		NOT (Z)	autonomous
Collaborative (Collbr)	collaboration	Goal reach. Z	inactive
		NOT (Z)	collaborative

Table 5.2: State transition model of cBDI agent

5.5.2.2 Strategic planner

The Strategic planner is responsible for maintaining a human-centric strategy for navigation task. Strategic planner is finite state machine that derive and handle strategic states i.e. “adopted” goals for cBDI agent during navigation.

Strategic planner has two FSMs running. The higher-level FSM is called $Strategy_{fsm}$. This machine derives the current strategy. The lower-level state machine called $Task_{fsm}$, is for controlling the execution order of derived strategy. The derivation of a strategy proceeds in discrete step in $Strategy_{fsm}$. One of its states, called sExecute, includes $Task_{fsm}$, which handles the state of selected strategy.

The $Strategy_{fsm}$ is composed of four control states: idle, EST, $Search_{strategy}$, *sExecute*. Initially the $Strategy_{fsm}$ is in idle state. On receipt of any information leading to belief update, then $Strategy_{fsm}$ transit to EST state. When a strategy is detected in state $Search_{strategy}$, then a signal is sent to the *sExecute*. *sExecute* controls the states of the selected strategy. The state transition model of $Strategy_{fsm}$ is presented in Figure 5-3 and more detailed presentation in the Table 5.3 . The latter represent transfer functions.

$Task_{fsm}$ activate on receipt of the selected strategy. $Task_{fsm}$ handle execution of the selected strategy. In $Task_{fsm}$ for each stategic state a signal as a "adopted goal" is sent to cBDI agent's desire.

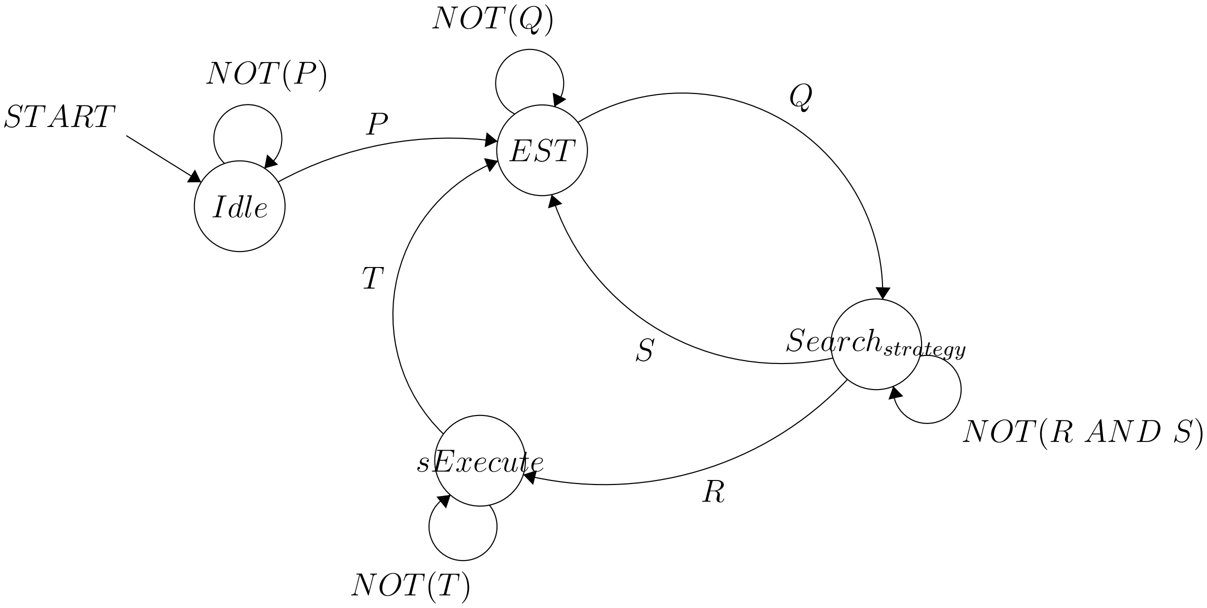


Figure 5-3: $Strategy_{fsm}$: States of Strategic planner for strategy control

States of Strategic planner in execution of strategy is presented is in Figure 5-4 and more detailed presentation in the Table 5.4. The latter represent transfer functions.

State	process	transfer function	next state
Idle	Default state	Get belief. P	EST
		NOT(P)	Idle
EST	Estimate current state	Current state received. Q	$Search_{strategy}$
		NOT (Q)	EST
$Search_{strategy}$	Search and select appropriate strategy	Selection is done. R	sExecute
		No appropriate state is found. S	EST
		NOT (R AND S)	$Search_{strategy}$
sExecute	Execute the states of the strategy	All needed state of selected strategy is executed. T	EST
		NOT(T)	sExecute

Table 5.3: $Strategy_{fsm}$: States of Strategic planner for strategy control

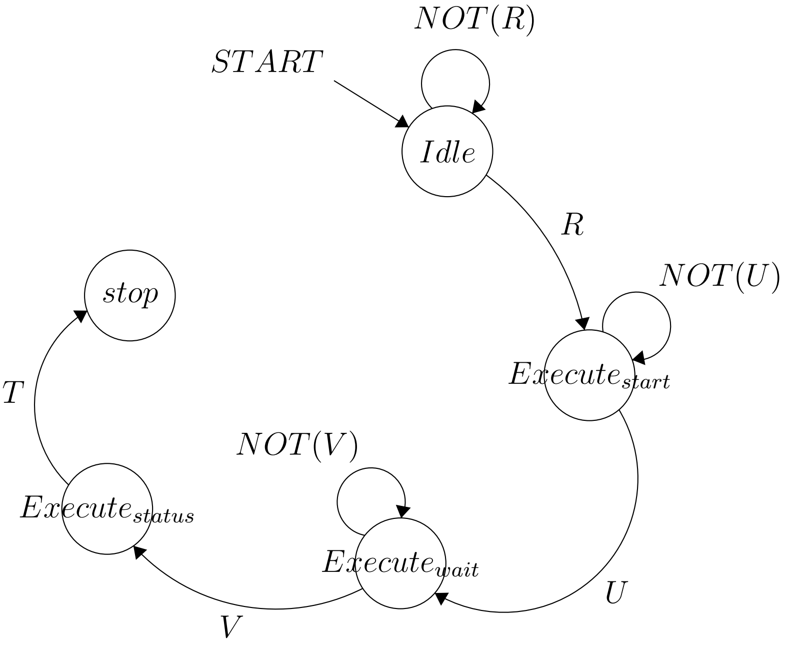


Figure 5-4: $Task_{fsm}$: States of Strategic planner for strategy execution

State	process	Transfer function	next state
Idle	Default state	Strategy received. R	$Execute_{start}$
		NOT(R)	Idle
$Execute_{start}$	Execution of each state of a strategy starts. signal sent to desire	Execution is started. U	$Execute_{wait}$
		NOT (U)	$Execute_{start}$
$Execute_{wait}$	Wait for execution done event	Execution done event received. V	$Execute_{status}$
		NOT (V)	$Execute_{wait}$
$Execute_{status}$	Sending status to sExecute	T	Stop

Table 5.4: $Task_{fsm}$: States of Strategic planner for strategy execution

5.5.2.3 Human centric strategy

This is a repository that contains human centric strategies. It describes what the cBDI agent is able to know about human strategy for navigation. Knowing the human (wayfinding) navigation strategies, cBDI agent is able to synthesize acceptable plans for collaborative navigation.

Based on the study in Chapter 4, the observable strategies of a human during navigation are categorized in to four distinctive categories:

- Least angle strategy (LSA): Human user clearly tries to maintain constant heading to the goal.
- Central point strategy (CP): Human user follows direction travel for some time. Reverse route to starting or previously visited location.
- Trajectory based strategy (TB): Human user follow path. At intersections prefer less meandering path. If paths are equally offset chose straighter route. Do not reverse route to starting location.
- Summary scanning strategy (SS): Human user, who is undecided about what to do next; that is a user neither expressing least angle strategy, route planning nor central point.

From these four types of strategies, cBDI agent exhibiting human centric strategy is modelled. If observed that human is using least angle strategy, agent is also going to apply least angle strategy. If state of observed strategy is central point, then agent is going to maintain central point strategy. If observed strategy state is trajectory based strategy, then agent is in route planning strategy. Otherwise, agent will employ summary scanning strategy. In summary, the cBDI agent could behave in each state as follows:

- $Collb-with-L$: The agent also approaches to goal using least angle strategy.
- $Collb-with-T$: The agent also approaches to goal using route planning strategy.
- $Collb-with-C$: The agent also moves towards goal using central point strategy.
- $Collb-with-S$: The agent tries to reach goal using summary scanning strategy.

A state transition that tries to relate the agent's strategy to reach the goal location is presented in Figure 5-5 and Table 5.5.

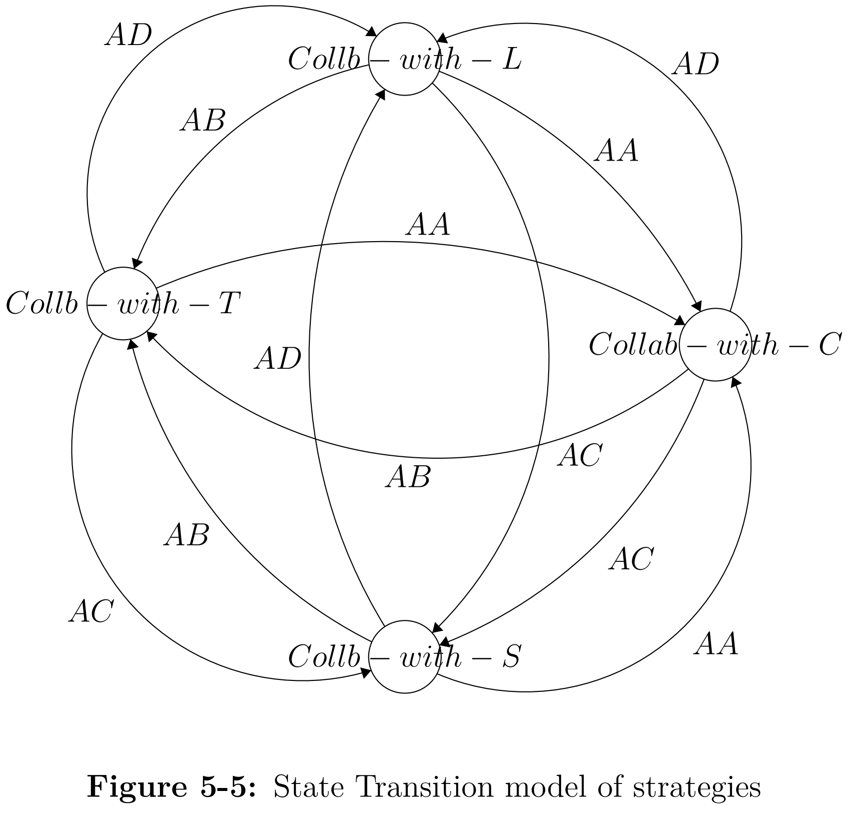


Figure 5-5: State Transition model of strategies

State	process	Transfer function	next state
$Collb-with-L$	Execute states that satisfy of Least angle strategy	When user intention is CA. AA	$Collb-with-C$
		When user intention. is TB. AB	$Collb-with-T$
		When user intention. is SS. AC	$Collb-with-S$
$Collb-with-C$	Execute states that satisfy of Central point strategy	When user intention is LSA. AD	$Collb-with-L$
		When user intention. is TB. AB	$Collb-with-T$
		When user intention. is SS. AC	$Collb-with-S$
$Collb-with-T$	Execute states that satisfy of Trajectory based strategy	When user intention is LSA. AD	$Collb-with-C$
		When user intention. is CA. AA	$Collb-with-T$
		When user intention. is SS. AC	$Collb-with-S$
$Collb-with-S$	Execute states that satisfy of Summary scanning strategy	When user intention is LSA. AD	$Collb-with-C$
		When user intention. is CA. AA	$Collb-with-T$
		When user intention. is TB. AB	$Collb-with-S$

Table 5.5: State Transition model of Strategies

5.5.2.4 Negotiator

The Negotiator provides a means of command arbitration and determines the final steering command in collaborative mode. It is based on the user intent and takes into account candidate direction proposed by the cBDI agent. Negotiator is to resolve conflicts over control action.

The command request are handled by negotiator through the finite state machine illustrated in Figure 5-6. The negotiator is composed of three control states: operative, act_{human}, act_{agent}. Initially, the negotiator is in operative state. If a conflict over control action is measured between the user input and suggested control action of the agent, then negotiator transit to act_{human} mode. If the value of erroneous behaviour is greater than a threshold, then negotiator is in act_{Agent}

mode–give the agent command.

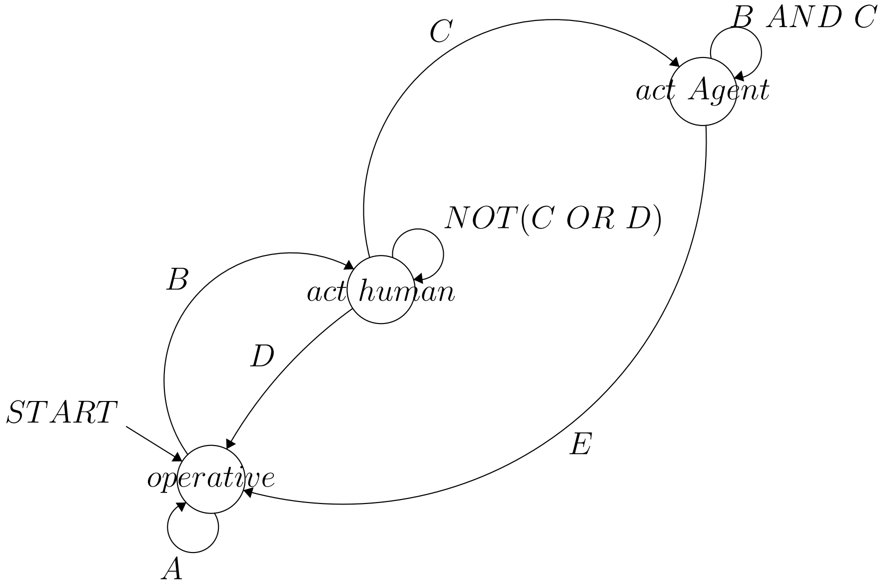


Figure 5-6: State Transition Model of Negotiator

State	Process	Transfer function	Next state
operative	Execute Command	Commands from the user or agent. A	operative state
		Conflict of command. B	act_{human}
act_{human}	Execute user command	Error in execution exceed a threshold value.C	act_{Agent}
		No error in execution. D	operative state
		NOT (C OR D)	act_{human}
act_{Agent}	Execute agent command	Execution done event.E	operative state
		B AND C	act_{Agent}

Table 5.6: State Transition Model of Negotiator

5.5.3 Local Control Layer

The Local Control Layer (LCL) also called the "Reactive layer" takes responsibilities of the low level control of the system hardware and provide access to incoming raw data of all sensors as well as outgoing commands to all actuators.

5.6 Description of "help when needed" scenario

This section aim to give a description of how C^3Arc "help" in user (failure) scenario. During the navigation, sometimes the human might not be able to drive the wheelchair in a way it's required to be drive. Such situations have been shown in Figure5-7.

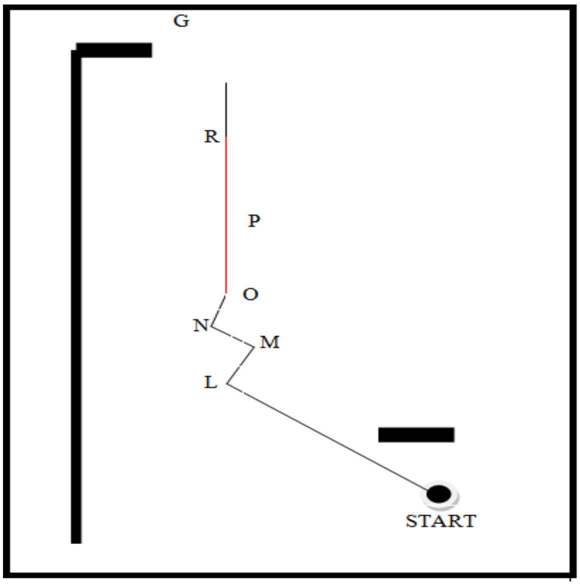


Figure 5-7: Help in "when needed" scenario: Controller behaviour during collaborative navigation. START represents the initial position. **G** represents the goal. L through R represents the positions of wheelchair at some specific time instance.

Suppose the user reach M and apply MOVE LEFT action. So the wheelchair is at N. Again the user applies MOVE RIGHT. So wheelchair reaches position O. In the meantime agent detect that user is using least angle strategy and the cBDI agent suggested action is MOVE FORWARD. At position O user apply again MOVE LEFT action. The negotiator detects the user control command. The negotiator also detects the agent suggested action. The negotiator infers that the user's suggested action of MOVE LEFT may not be perfect as the erroneous behaviour value rise to a threshold value. So negotiator passes the control command of the cBDI agent. So wheelchair reaches at position P and continually negotiator pass the suggested action of cBDI agent. At the position R, as human suggested action is same as that of the agent action, the negotiator pass the human action of MOVE FORWARD.

Figure 5-8 depicts the sequence diagram for collaborative mode of the controller. In collaborative mode, the procedure starts with user's knowledge and continues as seen in Figure 5-8. During collaboration, there is a negotiation between agent's command and the command specified by the user.

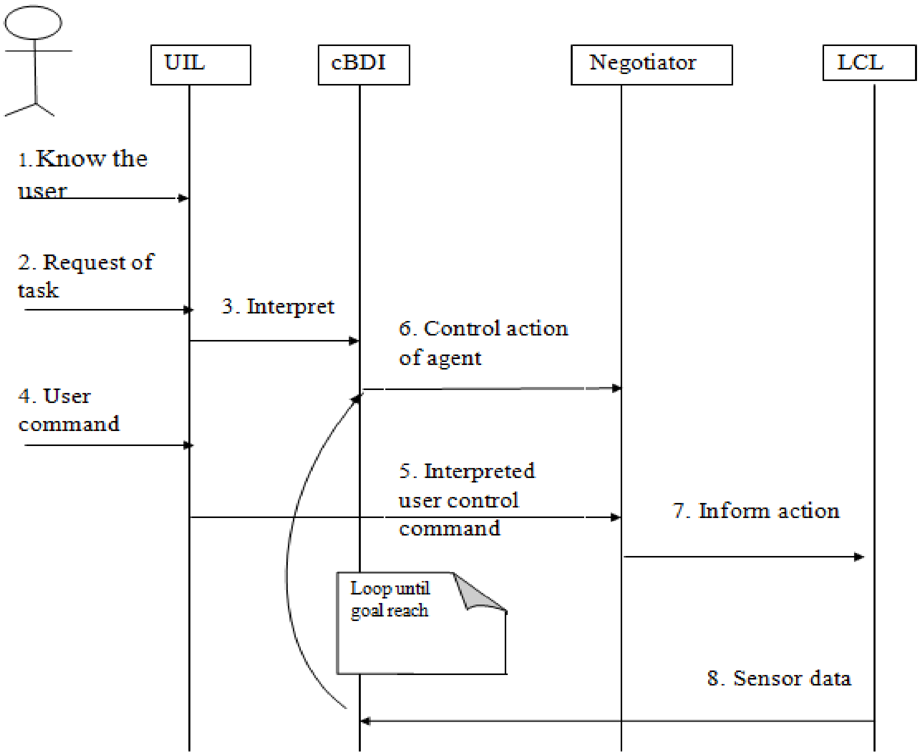


Figure 5-8: The sequence diagram for Collaborative mode

The controller can also be in autonomous mode. When controller is in autonomous operation, the procedure also is started with user's knowledge and continues as seen in Figure 5-9. In the autonomous mode, the user can only interact with controller by posting goals (request of task).

5.7 On assessment of C^3Arc architecture

This section presents a proof of concept evaluation of the control architecture C^3Arc. In assessment of our proposed architecture, USARSim (Unified System for Automation and Robot Simulation) and ROS (Robotic Operating System) have been used. ROS-USARSim is a combined framework for robotic control and simulation [227]. Simulation of C^3Arc is discussed briefly in the following subsections.

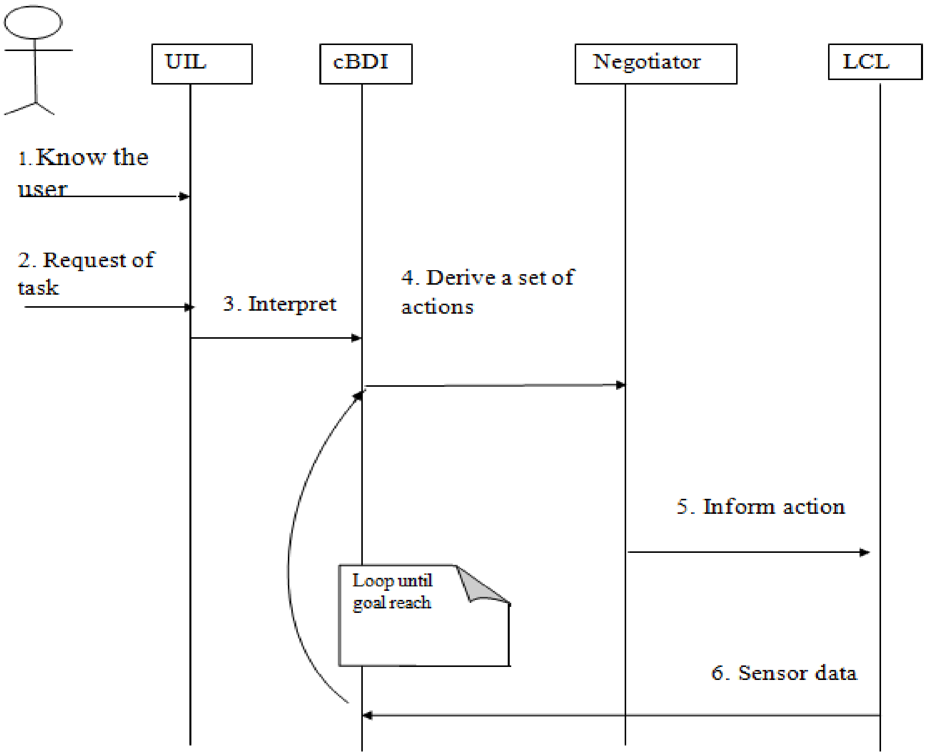


Figure 5-9: The sequence diagram for Autonomous mode

5.7.1 ROS-USARSim Simulation

ROS

ROS or Robot Operating System is an open source flexible framework for writing robot control [228]. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms. ROS is not an operating system in the traditional sense of process management and scheduling; rather, it provides a structured communications layer above the host operating systems [228]. We are interested in certain concepts of ROS which facilitates implementing collaboration between human and the machine. One of the key facility that ROS provides is called publish/subscribe anonymous message passing. A node in ROS environment is defined as a process of computation [229] and for every functionality a node can be constructed. A ROS topic is a communication channel between two or more nodes and a ROS node that is interested in a certain kind of data will subscribe to the appropriate topic [229].

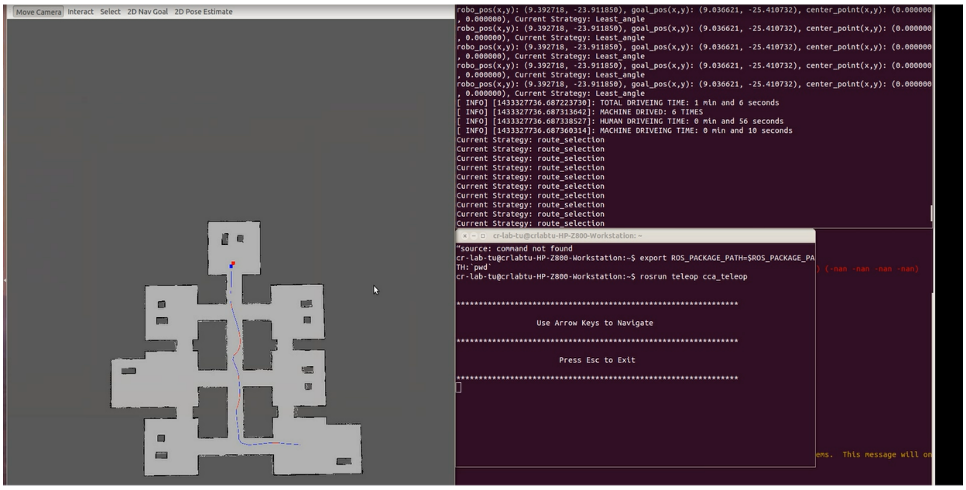
USARSim

Unified System for Automation and Robot Simulation (USARsim) is a simulator of urban search and rescue (USAR) robots and environments [230]. USARsim offers the possibility of providing a valid tool for the study of basic robotic capability in 3-D environment. USARsim provides users with the capabilities to build their own environments and robots. Here, USARSim is used to define an indoor environment. The robotic wheelchair system is based on pioneer P3AT.

5.7.1.1 C^3Arc as a ROS Node

Simulation of C^3Arc consists of three steps [1]:1. Generating virtual map for the robot in USARsim and map in ROS. 2. Spawn a P3AT robot in the simulated environment and 3. Bring up the implemented C^3Arc as ROS node.

P3AT imbibes C^3Arc. P3AT robot is treated as intelligent wheelchair. The 3D indoor environment created in USARSim simulator is seen as a 2D map using rviz: the ROS visualization tool. Figure 5-10 shows map of simulated environment generated by SLAM in ROS including the path of P3AT robot during collaboration.



```
robo_pos(x,y): (9.392718, -23.911850), goal_pos(x,y): (9.036621, -25.410732), center_point(x,y): (0.000000
, 0.000000), Current Strategy: Least_angle
robo_pos(x,y): (9.392718, -23.911850), goal_pos(x,y): (9.036621, -25.410732), center_point(x,y): (0.000000
, 0.000000), Current Strategy: Least_angle
robo_pos(x,y): (9.392718, -23.911850), goal_pos(x,y): (9.036621, -25.410732), center_point(x,y): (0.000000
, 0.000000), Current Strategy: Least_angle
robo_pos(x,y): (9.392718, -23.911850), goal_pos(x,y): (9.036621, -25.410732), center_point(x,y): (0.000000
, 0.000000), Current Strategy: Least_angle
robo_pos(x,y): (9.392718, -23.911850), goal_pos(x,y): (9.036621, -25.410732), center_point(x,y): (0.000000
, 0.000000), Current Strategy: Least_angle
[ INFO] [1433327736.687223730]: TOTAL DRIVEING TIME: 1 min and 6 seconds
[ INFO] [1433327736.687313642]: MACHINE DRIVED: 6 TIMES
[ INFO] [1433327736.687338527]: HUMAN DRIVEING TIME: 0 min and 56 seconds
[ INFO] [1433327736.687360314]: MACHINE DRIVEING TIME: 0 min and 10 seconds
Current Strategy: route_selection
Current Strategy: route_selection
Current Strategy: route_selection
Current Strategy: route_selection
Current Strategy: route_selection
Current Strategy: route_selection
Current Strategy: route_selection
Current Strategy: route_selection
Current Strategy: route_selection
Current Strategy: route_selection
```

```
"source: command not found
cr-lab-tu@crlabtu-HP-Z800-Workstation:~$ export ROS_PACKAGE_PATH=$ROS_PACKAGE_PA
TH:`pwd`
cr-lab-tu@crlabtu-HP-Z800-Workstation:~$ rosrun teleop cca_teleop


**************************************************************

                  Use Arrow Keys to Navigate

**************************************************************

                       Press Esc to Exit

**************************************************************
```

```
) (-nan -nan -nan -nan)
ems.  This message will on
```

Figure 5-10: Simulated environment generated in ROS-USARSim. Left hand side shows the environment for navigation including path taken by the P3AT robot during collaboration. Right hand side of the figure lists the strategy with other environment variables.

Blue line represents the path where human user controls P3AT through keyboard under teleoperated mode. The figure highlights the segments (the red colored lines) where C^3Arc is driving the P3AT even though the robot is supposed to be controlled through teleoperation.

---

[1] Mohamad Arif Khan, Biomimetic and Cognitive Robotics Lab, Tezpur developed the simulation of C^3*Arc*

5.7.1.2 Evaluation

Experimental method

To evaluate the viability of the navigation controller C^3Arc, a human subjects study was performed. We describe a preliminary user study using 12 number of participants. Participants were grouped into two different cognitive score group (low cognitive score and high cognitive score); and were made to individually drive the wheelchair to a goal location.

5.7.1.3 Procedure

All participants were asked to use the USARSim system to explore an environment. Prior to starting the task, participants were given verbal instructions on the objectives, and a demonstration of the controls. All subjects were required to confirm an understanding of the task and the controls by maneuvering through a training environment for several minutes. A questionnaire (inspired from MMSE [225] and IADL [226]) was prepared to estimate the cognitive score of each participant (Detail is in Appendix B). In this experiment all the participant had to drive the chair to the goal location in the environment shown in Figure 5-11. Participants were asked to control the robot in a safe and asked to control the robot in a safe and effective manner as possible.

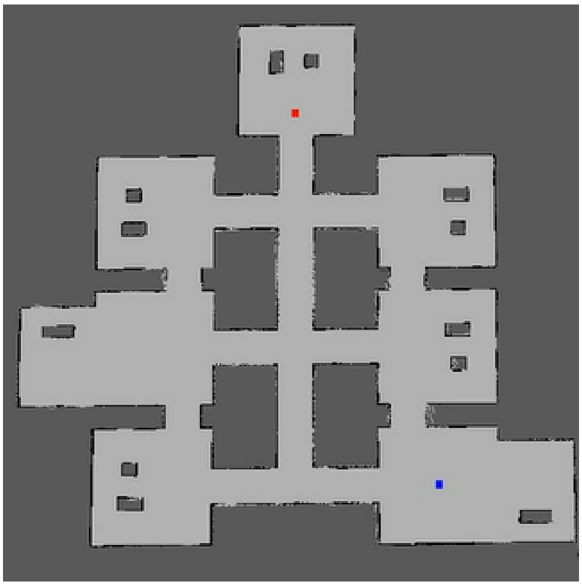


Figure 5-11: Environment for navigation task where blue box is start location and red box is the goal location.

5.7.1.4 Design

Participant belongs to either low cognitive score group or high cognitive score group. Unbeknown to them, the participants control the wheelchair movement in two different robot control method. Each of the robot control methods are detailed below:

- **No assistance mode:** In no assistance mode, participant control the robot movement manually. The participants drive the P3AT robot using arrow keys. In the no assistance mode the wheelchair would not get any assistance from the C^3Arc for any task at higher level.
- **Assistance mode:** In assistance mode, unbeknown to the participants, they had the opportunity to collaborate with the wheelchair. In assistance mode, the wheelchair through C^3Arc would perform higher level navigation task.

After completion of both the robot control round, participants were asked to answer the following:
Question Q_5: What is your overall reaction to the driving of the wheelchair? Do you "feel in control" of the wheelchair in both the round?

5.7.1.5 Participants

Twelve able bodies participants (Mean Age = 32.09, S.D = 8.63) were selected; all of them being either researchers or employed at Tezpur University. Participants were naive to the purpose of the experiment.

5.7.1.6 Apparatus

The robotic WC was controlled using arrow keys on a standard 101 computer key board. To run the simulation two PCs were used: 1. USARSim was administered on a HP computer equipped with Core i3 and 12 GB RAM and a 21-inch monitor. 2. ROS Fuerte version was administered on another HP computer equipped with Core i3 and 12 GB RAM and a 21-inch monitor.

Dependent Measures

Participant's performance assessment was based on following metrics:
Finish Time: Total driving time to reach the goal location.[2]

---

[2] Dependency measure–finish time is used to evaluate ease of driving of the wheelchair in different mode and for different group of participant.

Safe margin: Total time a participant cross a *minimum safe margin value d1.*[3]

Hypotheses

Based on the dependent measures described above, four main hypotheses in this experiment were as follows:

H_6^0: For low cognitive score group, finish time in two modes is not different.

H_7^0: In assistance mode, there is no difference in finish time for two group of participant.

H_8^0: In assistance mode, there is no difference in safe margin time for two group of participant.

H_9^0: Participants do not sense difference in two control methods of wheelchair driving.

5.7.1.7 Results

As each participants had to drive the wheelchair in two control modes. Here, the trajectories within two different control modes are discussed first.

5.7.1.8 Experimental trajectory

Experimental trajectories give a scope for user behaviour evaluation.

Experimental trajectory in no assistance mode:

Figure 5-12 shows a sample path taken by a participant, who has low cognitive score, during no assistance mode. All participants take routes nearly similar to this sample trajectory. From the figure, it is clear that participant with low cognitive score shows erratic style of driving.

Figure 5-13 shows a sample path taken by a participant, who has high cognitive score, during no assistance mode to reach the goal position. All participants with high cognitive score take routes nearly similar to this sample trajectory. As in Figure 5-13 path taken by the participant (belonging to high cognitive score group) is smooth, which is not the case for participant with low cognitive score.

Experimental trajectory in assistance mode

Figure 5-14 shows a sample trajectory taken by a participant with low cognitive score during assistance mode. Blue colored path is where the participant is controlling the wheelchair. The red colored lines are where the wheelchair initiated

---

[3] Dependency measure–safe margin is used to evaluate wheelchair behaviour "in help when needed" scenarios.

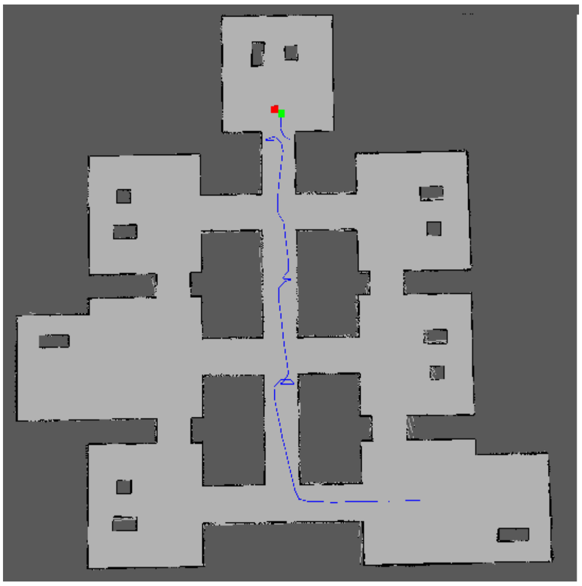


Figure 5-12: Figure displays a sample trajectory taken by participant with low cognitive score during no assistance mode.

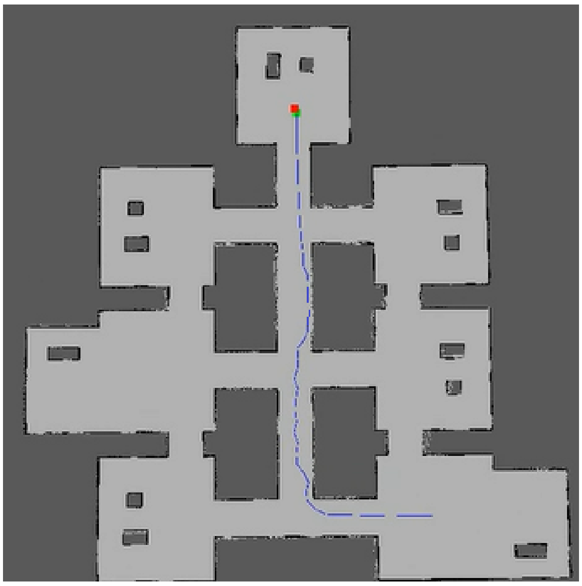


Figure 5-13: Figure displays sample trajectory taken by participant with high cognitive score during no assistance mode.

control. To give an illustration on wheelchair behaviour i.e how controller C^3Arc initiate control in assistance mode, path taken by participant (participant number 12) is analyzed as follows:

The black circles show the continuation of the human controlling process (how user wants to control the chair), which also depict supporting behaviour of the chair. As in Figure 5-14, the participant has shown erratic style of driving yet within acceptable limits and robot did not show any deviation; instead shows support. The yellow circles represent the part of trajectory where quality of driving is not up to the mark as safe margin increases above a threshold; controller even-

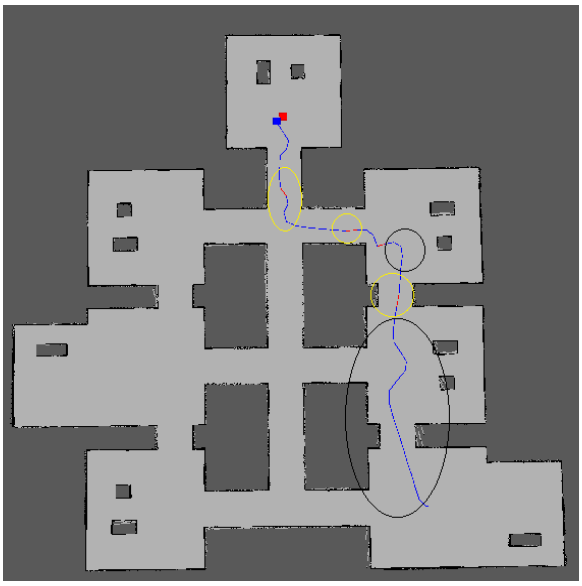


Figure 5-14: Figure displays trajectory taken by participant number 12 during assistance mode

tually has decided to take initiative. Controller C^3Arc took over the control of the chair. Following three point are clear from this illustration:

- Wheelchair initiate help when needed.
- There is no unnecessary deviation of the path before and after transfer of control between wheelchair and participant.
- It is expected that the wheelchair adapt the navigation strategy of user when it overrides control. From followed path it is clear that wheelchair use the same strategy of user after overriding.

Figure 5-15 shows the case where robot did not show any initiative action, because participant was successfully able to control the chair.

5.7.1.9 Assessment of participant performance

Table 5.7 and 5.8 provides descriptive statistics for the dependent measures. In no assistance mode, participant belonging to low cognitive score group took a mean of 153.48 sec to complete the task. This decreased to 78.68 sec, when performing the same task in assistance mode. In no assistance mode, participant belonging to high cognitive score group took a mean of 89.65 sec to complete the task. This is marginal decrease to 66.67 sec, when performing the same task in assistance mode.

In no assistance mode, participant belonging to low cognitive score group crossed a minimum safe margin value with mean of 47.60 sec. This decreases

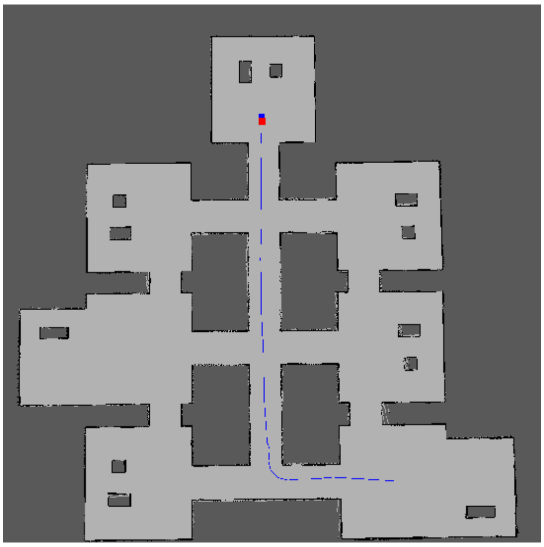


Figure 5-15: Figure displays trajectory taken by participant (number 3) with high cognitive score during assistance mode

to 13.4 sec in assistance mode. In no assistance mode, participant belonging to high cognitive score group showed a mean of 23.14 sec in safe margin to complete the task. This decreased to 3.42 sec in assistance mode.

Table 5.7: Mean (± Standard deviation) for finish time in seconds.

Mode	Group	
	Low cognitive score	High cognitive score
No assistance mode	153.48(32.42)	89.65(21.89)
Assistance mode	78.68(33.12	66.67(8.30)

Table 5.8: Mean (± Standard deviation) for safe margin in seconds

Mode	Group	
	Low cognitive score	High cognitive score
No assistance mode	47.60(12.81)	23.14(15.45)
Assistance mode	13.4(11.26)	3.42(2.49)

Figure 5-16 shows finish time under each control method. Figure 5-17 shows safe margin time under each control method. In assistance mode participant with low cognitive score performed nearly same as to the participant with high cognitive score.

Null Hypothesis (H_6^0): For low cognitive score group, finish time in two modes is not different.
Alternative Hypothesis (H_6^a): For low cognitive score group, finish time in two modes is different.

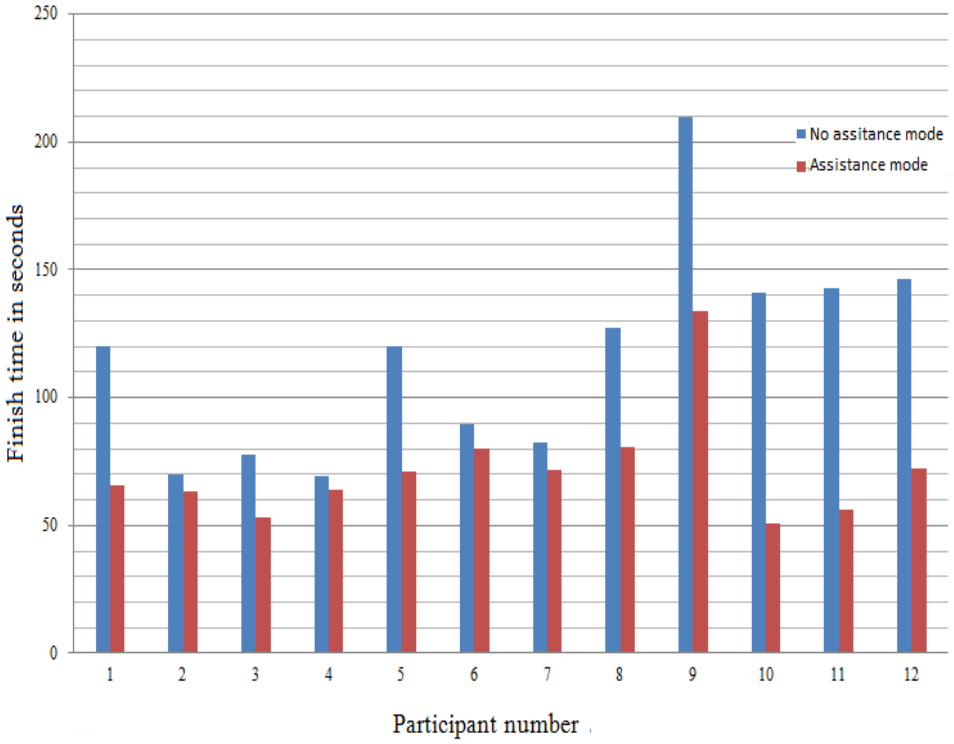


Figure 5-16: Figure displays finish time (in seconds) for each participant under assistance as well as no assistance mode

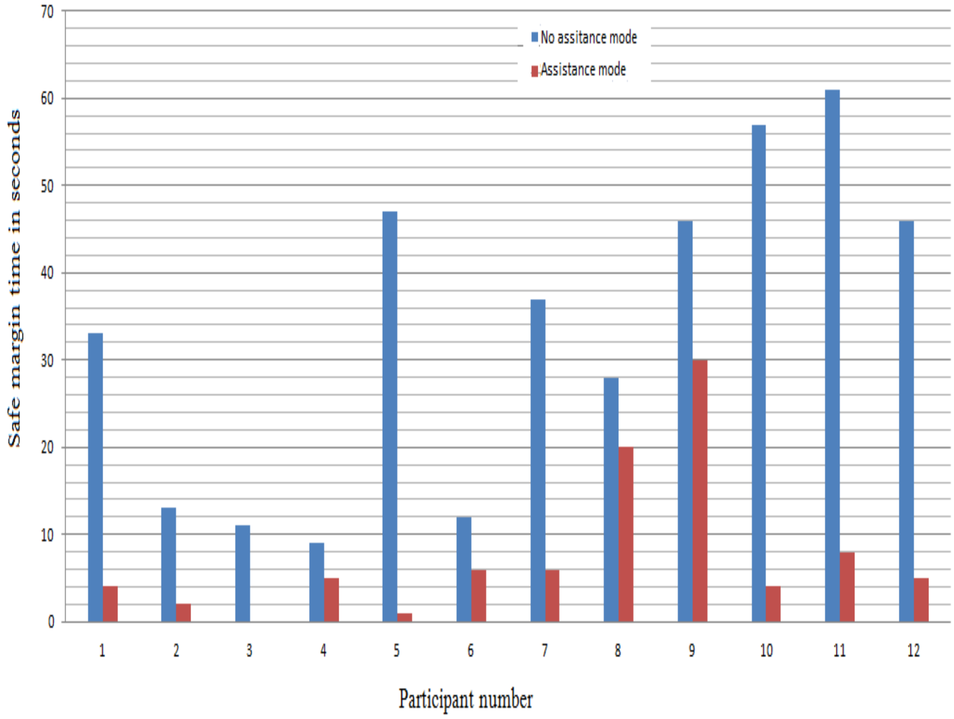


Figure 5-17: Figure displays safe margin time (in seconds) for each participant under assistance as well as no assistance mode

To test H^{0}_{6}, ANOVA was conducted. For low cognitive score group, interaction between finish time and modes was significant, $\mathrm{F}(1, 9) = 13.022, p = 0.007$. So the null hypothesis (H^{0}_{6}) was rejected. Alternative hypothesis (H^{a}_{6}) was accepted. We conclude, for a 95 percent confidence interval, that for low cognitive score group finish time in two modes is different.

Null Hypothesis (H^{0}_{7}): In assistance mode, there is no difference in finish time for two group of participant.
Alternative Hypothesis (H^{a}_{7}): In assistance mode, there is a difference in finish time for two group of participant.

To test H^{0}_{7}, ANOVA was conducted. In assistance mode interaction between cognitive score group with finish time was not significant, $\mathrm{F}(1, 11) = 0.833, p = 0.383$. So the null hypothesis (H^{0}_{7}) was accepted. Alternative hypothesis (H^{a}_{7}) was rejected. We conclude, for a 95 percent confidence interval, that there is no difference in finish time for two group of participant in assistance mode. However, in no assistance mode interaction between cognitive score group with finish time is significant, $\mathrm{F}(1, 11) = 16.77, p = 0.002$.

Null Hypothesis (H^{0}_{8}): In assistance mode, there is no difference in safe margin time for two group of participant.
Alternative Hypothesis (H^{a}_{8}): In assistance mode, there is a difference in safe margin time for two group of participant.

ANOVA reported a significant interaction between safe margin time in assistance mode and cognitive score groups, $\mathrm{F}(1, 11) = 5.342, p = 0.043$. So the null hypothesis (H^{0}_{8}) was rejected. Alternative hypothesis (H^{a}_{8}) was accepted. We conclude, for a 95 percent confidence interval, that there is a difference in safe margin time for two group of participant in assistance mode.

5.7.1.10 Assessment of participant feedback

In answer to the question Q_5, 83.33% of participant reported that the wheelchair driving in two control method is same.

Null Hypothesis (H^{0}_{9}): Participants do not sense difference in two control methods of wheelchair driving.
Alternative Hypothesis (H^{a}_{9}): Participants sense difference in two control methods of wheelchair driving.

Participant with low cognitive score assessed the wheelchair driving in two control method as same. While five out of seven participants with high cognitive score reported that the wheelchair driving in two control method as same. Two participants with high cognitive score reported that assistance mode is easier than no assistance mode. A Mann-Whitney U test show there was no significant difference between the cognitive score group and their assessment of robot control ($U = 12.50, z = -1.254, p = 0.210$). So the null hypothesis (H_9^0) was accepted. Alternative hypothesis (H_9^a) was rejected. We conclude, for a 95 percent confidence interval, participants do not sense difference in two control methods of wheelchair driving. From this result we can assume that assistance mode was able to give "feeling in control" sense in wheelchair driving.

Table 5.9: Summary of hypothesis testing-II

Testing of hypotheses	Hypotheses considered
Hypothesis H_6^0	H_6^a is accepted
Hypothesis H_7^0	H_7^0 is accepted
Hypothesis H_8^0	H_8^a is accepted
Hypothesis H_9^0	H_9^0 is accepted

5.7.1.11 Conclusions

From the results we make following conclusions:

1. Navigation behaviour of person from the low cognitive score group improved through assistance from the collaborative control architecture.

2. Behaviour of the collaborative controller in assistance mode is independent of person's cognitive score.

3. Even though there exist a difference in safe margin time in assistance mode for two group of participants, as wheelchair took initiative in many situations, participant with low cognitive score performed nearly same as to the participant with high cognitive score.

4. Assistance mode was able to give "feeling in control" sense in driving of the wheelchair, participants did not realize difference in control methods.

5.7.2 Justification of the term "cognitively enhanced"

From the study we conclude that the C^3Arc exhibit cognitively enhanced control behaviour. Within the framework cBDI agent induce cognitive capability of decision making and makes controller to behave as cognitively enhanced. From the study we justified our claim that a cognitively enhanced control framework for wheelchair navigation can improve collaboration between wheelchair user–machine.

5.8 Chapter Summary

In this chapter, a framework for collaborative navigation is presented. The framework presented in this chapter applies the cBDI architecture in *collaborative navigation.* The architecture addressed cognitive collaborative control for an intelligent wheelchair. Before formalizing the architecture basic requirement are considered. Chapter described the importance of negotiation. We have presented help in "when needed" scenario. We have presented proof of concept evaluation of the proposed framework.