

A MODEL PREDICTIVE CONTROLLER FOR SOCIAL PERSON-FOLLOWING ROBOT FOR LONG-TERM INDOOR NAVIGATION

Thesis submitted in partial fulfillment
of the requirements for the degree of

*Master of Science in **Computer Science and Engineering** by Research*

by

AVIJIT KUMAR ASHE

20172067

avijit.kumar@research.iiit.ac.in



International Institute of Information Technology, Hyderabad

(Deemed to be University)

Hyderabad - 500 032, INDIA

JUNE 2023

Copyright © ASHE, AVIJIT KUMAR 2023
All Rights Reserved

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “A Model Predictive Controller For Social Person-Following Robot For Long-Term Indoor Navigation” by Avijit Kumar Ashe, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. K. Madhava Krishna

To Science. To Love.

Acknowledgments

I would like to extend my sincere gratitude to everyone who helped me push myself during this journey and learn by doing. To everyone who shared their thoughts and advice, to all who became a part of it for any amount of time, and to every individual who was more than happy to listen to me, I am grateful. This journey took several years of which a significant time went into learning, and I wish to thank all my professors who gave me the knowledge and experience to fulfill this thesis.

First, I would thank my guardian, guide, and inspiration, Prof. K. Madhava Krishna for giving me this opportunity to be a part of IIIT Hyderabad and accepting me as a part of the Robotics Research Center (RRC) eventually. I am thankful for letting me work on various projects and helping me channelize my research career. I found a favorable and nurturing environment under his guidance and immense patience during the times I was lost or occupied with other responsibilities. I grew as a researcher and can fulfill my dream of becoming a good scientist, a good human, by continuing on his guidance. To use knowledge because it empowers is the best way forward. I was also privileged to work with Gaurav Kumar, Saket Saurav, Harit Pandya, and Bharath Gopalakrishnan during my beginning years of joining RRC. I would extend my sincere thanks to Mithun Babu for his reviews of my work and for helping me during the main course of research work. I would like to thank Alog Tech and its CEO Raghuram Nanduri for guiding me towards my problem statement of this research. I would also like to thank IIIT, Hyderabad for the financial support in carrying out my education and research in the last few years.

And, last but not least my parents, family of friends who supported me all the while. I thank my friend Ashish Kumar for his financial support during the publications. I am grateful to Lalatendu Patanaik for his strength and patience. And, above all, my lovely friend and companion Maneesa Sahoo whose unconditional love and support during the pandemic helped me complete the most critical part of this journey from research to publication, and beyond.

Abstract

The applications of shared autonomy or human-robot interaction are growing rapidly in the field of autonomous robotics. Assisting human beings in dynamically changing environments in urban areas is still an active area of research. In crowded scenarios, in a structured environment such as public places with occlusions and dynamic obstacles, moving vehicles, people and so on - is the most critical part of this challenge. And, our work focuses on developing effective control strategies using model predictive control (MPC) because it is best known for handling such uncertainty and complex system dynamics relatively easily. While the extensive use of data-driven techniques using machine learning has become the de facto solution today, the underlying physics, the model of a system, and its behaviour are necessary to develop control laws. We first design an innovative MPC controller for a social person follower that can move safely around humans. We further incorporate motion-planning, target-tracking, and social norms into a single holistic framework, being the first of its kind on a differential drive-wheeled mobile robot. To develop this robust person following behavior, we also employ path prediction using LSTM (Long-Short Term Memory) a type of recurrent neural network for supervised learning. This allows us for out-of-sight tracking and natural anticipation of a person's future state. We also develop a local-map-based early relocation (ER) strategy that can reduce oscillations in the path, maintain the field of view (FOV) for long-term indoor navigation. Thus, we move beyond trivial person following to anticipating future visit locations and following them in the present.

Overall, a non-linear MPC-based control law is designed using an online optimization problem with constraints on both kinematics and dynamics, as well as social norms of safety around humans. We implement these using 2D simulations in Matlab, and in Python to test the controller performance, runtime analysis, and error analysis. We show that the MPC framework can run in real-time with an adequate margin for adapting to changing human movement patterns, and agile enough for its changing movement speeds.

Contents

Chapter	Page
1 Introduction	1
1.1 Motivation	1
1.2 Approach	3
1.3 Objectives	5
1.4 Organization	5
1.5 Contributions	6
2 Person Following Robot	7
2.0.1 In-View Tracking	8
2.0.2 Off-View or Predictive Tracking	9
2.0.3 Summary	9
3 Controller Design	12
3.1 Related Work	13
3.1.1 MPC for Collision Avoidance & Path Tracking	13
3.1.2 MPC for Person Following Behaviour	14
3.2 Novelty	14
3.3 Optimization	14
3.3.1 Kinematic Model	15
3.3.2 Cost Function	15
3.3.3 Constraints	16
3.3.4 Kinematic Constraints	16
3.3.5 Field of View Constraint	16
3.3.6 Collision Avoidance Constraint	17
3.4 Results	18
3.4.1 Simple Person Following Behaviour	18
3.4.2 Bending Around Corners	18
3.4.3 Multiple Consecutive Bendings & Corridors	20
3.4.4 A Dynamic Obstacle & Moving Person	21
3.4.5 Static & Moving Obstacles Cross Person's Trajectory	22
3.4.6 FOV Analysis	22
3.4.7 Quantitative Analysis	25
3.4.8 Ablations	27
3.5 Conclusions	28

4	Maneuvering Intersections	30
4.1	Related Work	31
4.1.1	Motion Planning for Autonomous PFR At Intersections	31
4.2	Problem Formulation	32
4.3	Dynamics of Person Following Robot	33
4.3.1	Reference Trajectory For Person Following Robot	33
4.4	OPTIMIZATION	34
4.4.1	Goal Reaching Constraints	35
4.4.2	Early Relocation Constraints	35
4.5	Intersections & Early Relocation	35
4.5.1	Choice of Early-Relocation Way-Points	35
4.6	RESULTS	37
4.6.1	Case 1: Simple T & L Junction	37
4.6.2	Case 1: Simple T & L Junction + Dynamic Obstacle	38
4.6.3	Case 3: 3-Way T-Junction At Crossroads	38
4.6.4	Case 1: FOV Analysis in Crowd & Dynamic Obstacles	40
4.6.5	Runtime Analysis	41
4.7	CONCLUSIONS	42
5	Out-of-Sight Prediction	43
5.1	Related Work	44
5.2	Problem Formulation	45
5.3	Long-Term Prediction Model	45
5.4	Results	46
5.4.1	Case 1: Simple Junctions, No Crowd	46
5.4.2	Case 2: Crowd by Moving Humans and LSTM	48
5.4.3	Runtime Analysis	50
5.5	CONCLUSIONS	50
6	Conclusions	52
6.1	Related Publications	53
6.2	Other Publications	53
6.2.1	Supplementary Publication	53

List of Figures

Figure	Page
1.1 Natural Person Following - Urban Real-World Scenarios. Left: Zebra intersections, Center: Warehouse, Right: Grocery shopping	1
1.2 PFR- A Real-World Simple Prototype. Left: P3DX front-view with mounted RGBD Microsoft Kinect. Right: P3DX rear-view with control computer with a target person in view.	3
1.3 Overview of MPC controller	4
1.4 A person moves in a indoor office through a narrow path between a wall and a bookcase to generate the reference recorded trajectory (blue). The PFR tries to find a collision-free path (red) with several people moving in between.	5
2.1 Studies from 1999-2021 of developing person-following robots. Evolution of human interaction vs system performance in agility and intelligent behaviour.	10
3.1 A Gazebo world snapshot. A person moves in a indoor scene with humans and traces a trajectory (black) with circular nodes as way-points. The Husky bot follows it safely (red).	12
3.2 FOV constraint for an arbitrary angle β	17
3.3 Simple piece-wise straight line/curvilinear path	19
3.4 Node-to-node transition strategy	19
3.5 Indoor setting with walls, a square room, executing two bending around corners	20
3.6 Multiple long corridors, hallways, executing consecutive turns.	20
3.7 Space-time plot of Fig 3.6 with positions of Person and PFR at $t=10, 18, 28, \text{ and } 38$. . .	21
3.8 One dynamic obstacle crosses the person, shown at $t=26$	22
3.9 Space-time plot of Fig 3.8 with positions of Person and PFR at $t=11, 22, 27, 31, 36, \text{ and } 42$	23
3.10 Scenario with both static and dynamic obstacle	24
3.11 Trajectory with 360° FOV	24
3.12 Trajectory with 30° FOV	25
3.13 Number of Obstacles vs Runtime per MPC Computation	26
3.14 Human Movement Patterns vs Trajectory Tracking Runtime	26
3.15 Violation of CA and FOV constraints vs Scene Complexity	27
3.16 Violation of CA and FOV constraints vs Movement Patterns	28
4.1 A snapshot of real-world intersections. In yellow we show L and T-junctions in a structured outdoor environment.	30

4.2	Common Y, T, L-junctions in structured urban indoor floor plans, and probable ER locations in various scenarios	32
4.3	Setting Starting Distance & Prediction Horizon	33
4.4	State-to-State Transition of PFR & Reference Person	34
4.5	Setting Starting Distance & Prediction Horizon	36
4.6	Normal Maneuver Around L-Junction Using R_f Without ER	36
4.7	Maneuver Around L-Junction Using Early Relocation & R_f	37
4.8	Simple T,L-Junction With/Without Early-Relocation	38
4.9	Large Turning Radius at T,L-Junction With/Without Early-Relocation	39
4.10	Dynamic Obstacle Course in T,L-Junction With/Without Early-Relocation	39
4.11	Simple T-Junction With/Without Early-Relocation	40
4.12	3 Dynamic Obstacles Left: Without ER, Right: With ER	40
4.13	Runtime vs Number of Obstacles for Fixed Trajectory	41
4.14	Trajectory Types vs per-MPC Plan Runtime, Effect of ER on Loss of Target Tracking	42
5.1	A target person traces a reference path in a structured environment (blue) between the narrow wall and bookshelf. The P3DX bot can follow the person (solid red) normally or it can use long-term prediction to find the unknown goal position behind the bookshelf, at a T-junction and move to it autonomously (dotted red).	43
5.2	With dynamic humans in the local map, the above trajectories get shorter. Based on the visible reference path of the target person (blue) and current robot pose (red) it predicts a path leading to human discomfort and collision. Tracking breaks as the person moves beyond the shelf, requiring LSTMs for path prediction after missing the person from FOV.	45
5.3	Closed-Loop Framework for In-FOV vs Out-of-FOV States of MPC Controller for Long-Term Prediction by LSTM and ER	46
5.4	Static Obstacle Course in T,L-Junction	47
5.5	Static Obstacle Course in T,L-Junction Diversion at 26th Node, in FOV from ER Point	47
5.6	Static Obstacle Course in T,L-Junction Another Possible Diversion at 26th Node, in FOV from ER Point	48
5.7	Dynamic Obstacle Trajectory With LSTM with 2 Moving People, and 1 Target Person. 1 PFR in Purple Track	49
5.8	Dynamic Obstacle Trajectory With LSTM for 4 Moving People, and 1 Target Person	49
5.9	Our Configuration of P3DX 2-Wheeled Drive with Mounts	51

List of Tables

Table	Page
4.1 Example of ER on Trajectory Completion	42
5.1 Ablations: Path Completion vs LSTMs	50

Chapter 1

Introduction

Robot behaviour adaptation to human social norms is a growing field of research in robotics that aims at co-existence rather than segregation. As a human in the loop, guidance and human-robot interaction are key factors that enable us to sustain this harmony. A decisive part of this process involves sensors, actuators and control laws. We design a predictive controller for a socially adept person-following robot in this work that complies with kinodynamic model of the robot as well as safety, and comfort around humans. Above all, it goes beyond trivial per-frame tracking.

1.1 Motivation

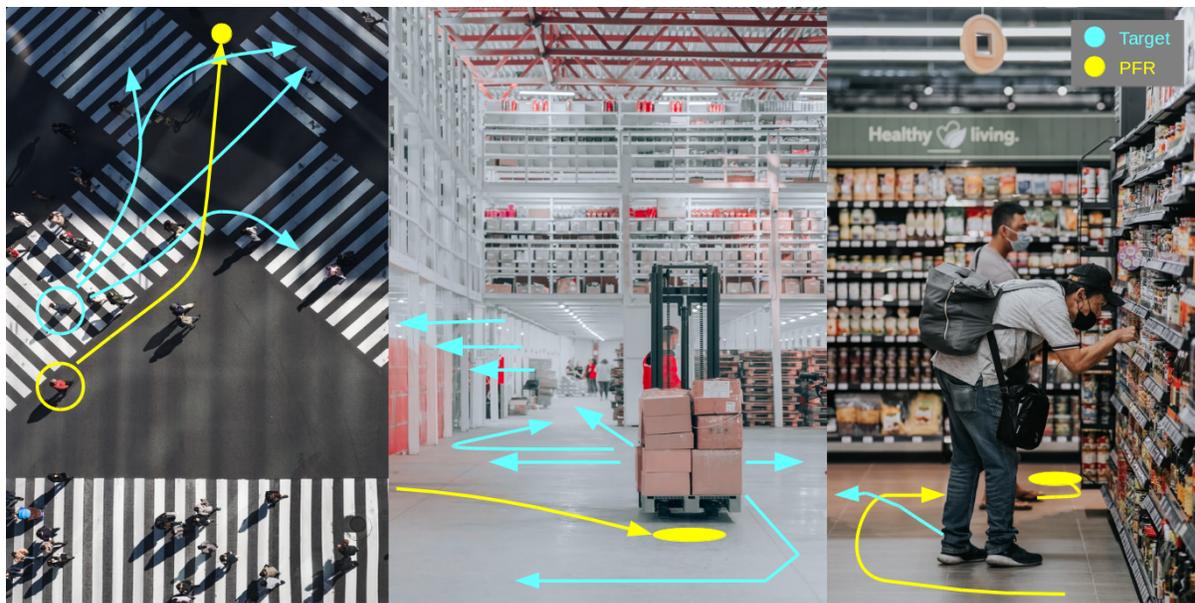


Figure 1.1: Natural Person Following - Urban Real-World Scenarios. Left:Zebra intersections, Center:Warehouse, Right: Grocery shopping

Consider the above three different urban world scenarios. All of them have one thing in common. The environment is structured or artificial, and predictable because of patterns, signs, and consistency. This allows us to follow certain rules or social norms for streamlining our motion. For example, using the pedestrian crossing, and walking along aisles. On the other hand, humans tend to follow moving objects (or persons) using visual tracking without needing any explicit motion model. For example, catching an incoming ball. Now, a kid following their parent on a footpath includes both - visual tracking and rules for walking on marked signs. So, even if the parent is not always in the FOV of the kid, it can keep following using such knowledge.

Now, translating this behaviour to a mobile robot here serves as our strongest motivation. While working on an autonomous cart for warehouses traditionally built for human workers, we realized that a semi-autonomous policy with human-in-the-loop allows for a more scalable and reliable strategy to solve the navigation problem, of moving from point A to B.

In Fig. 1.1, the paths in cyan denote a human (target) walking naturally. The paths in yellow denote a person (or a robot) following the cyan target. On the left, we have a multi-zebra crossing with an X-cross intersection. The target can execute 4 different cyan paths in the future. As the yellow follower has no clue, it can either follow it every step (per visual frame) from a few steps away, or it can be smart and choose a vantage location (yellow dot) and monitor the progress of the target person. This gives it added advantage of not meticulously tracking the person in every visual frame, saving energy and time, and also being robust to occlusions due to short-sightedness. The vantage point offers a wider field of view (FOV) on all possible cyan future paths, the target person can take. So, by reaching the location ahead of time, it can resume tracking once the person has progressed in any one of the certain directions.

At the center, we have a warehouse scenario where a person and a forklift remote-operated vehicle are moving together, the vehicle following the person. There are several aisles, intersections, and return points. While the person can trace many possible cyan paths in the future that can be infeasible for a non-holonomic system to follow strictly. It makes more sense to station it at a vantage point (yellow dot) and wait till the target person is on one certain path (or certain distance) where per-frame tracking can be resumed without any chance of getting stuck. This helps avoid the freezing problem in mobile robots. The physical limitations make this even more sensible. Finally, on the right, the grocery shopping scenario is a very common application of assisted robots. The yellow dot at the intersection of the aisles of the grocery floor is a great spot for someone to wait, to follow the cyan target once it has executed a certain maneuver.

In all the above scenarios, the presence of consistent structure helps us design a natural following behaviour. Therefore, not following the target moving object in every frame allows for off-view tracking using prediction (from knowledge of social norms and the structure of the environment) and dealing with partial occlusions and uncertainty in future reference trajectories. In this work, we design an MPC controller that can handle such scenarios as naturally as a human would, or at least close enough.

1.2 Approach



Figure 1.2: PFR- A Real-World Simple Prototype. Left: P3DX front-view with mounted RGBD Microsoft Kinect. Right: P3DX rear-view with control computer with a target person in view.

We presented above three most common applications. To simplify the above settings we bring the robot from following a person on a busy street to an indoor office-like environment with corridors and pathways around rooms and cabins as junctions on the street and a P3DX robot as the person-following robot (PFR). And, we assume the static map is available. A P3DX is a popular 2-wheel drive robot that can be mounted with a stereo camera, and RGBD sensors running ROS to operate it. It translates well to any modern warehouse robot, a service robot for home applications, and so on. To allow the PFR to track the person we need a controller that sends velocity commands to the wheels of the robot at each time step through the reference trajectory provided by the target person. By formulating it as a model predictive controller (MPC) framework, we design an MPC routine that optimizes for the best control commands to orient and drives the robot by following several constraints. These include the physical limitations on velocity, acceleration of the robot, the goal-reaching behaviour, and the safety factor of operating within a comfortable space around the person/person(s) involved as dynamic actors in the

map. Fig. 1.2 shows a simple PFR prototype in a lab setting to visualize how such a semi-autonomous strategy may work.

In a crowded street, at shopping malls, at a nursing center, in a library, at a museum records room, in search and rescue, and even in defense, the role of a personal assistant is evidently prominent. And, we have covered them in detail in Chapter 2 in the literature survey. Meanwhile, they [1,2,3,4,8] force our attention to the concept of shared autonomy with a human in the loop so that in all such applications a PFR can offer assistance. It stands somewhere in between complex complete autonomy and trivial remote-controlled passive operation. Here, the primary task of a PFR is not just to track a specific person but also autonomously navigate from one goal to another goal location, perform path and motion planning on-the-fly and have a long-term operation. On the social side, it must be able to interact with its human companion, with the other humans around it, be able to foresee conflicting scenarios, and plan ahead of time to avoid them. That is, perform dynamic collision-avoidance as well while continuously updating its strategy based on the current scenario, state, and map information. This brings us to a simpler configuration of maneuvering intersections in an indoor setting while still complying with social norms. Fig. 1.3 shows the overview of our PFR’s MPC controller.

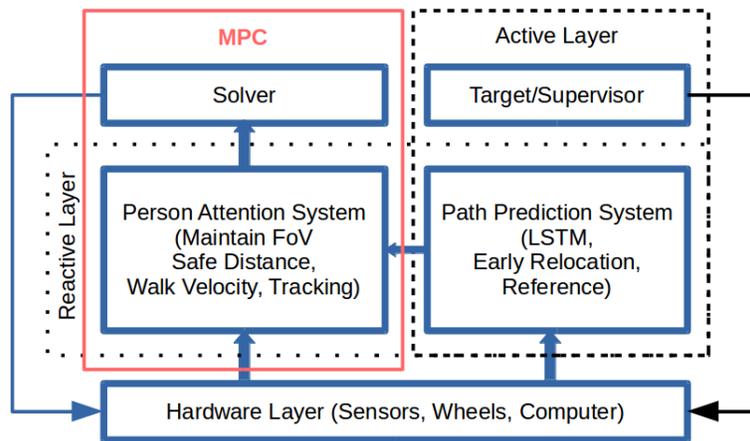


Figure 1.3: Overview of MPC controller

The controller, explained in detail in Chapter 3, has a layered structure. It has an autonomous layer and a semi-autonomous layer with human interaction. This is called the reactive layer because it reacts to movements by the target person. There is an active layer that autonomously navigates to vantage locations in a pre-defined map (e.g. of a warehouse, or office). Finally, the hardware layer communicates over ROS protocol to drive the vehicle around.

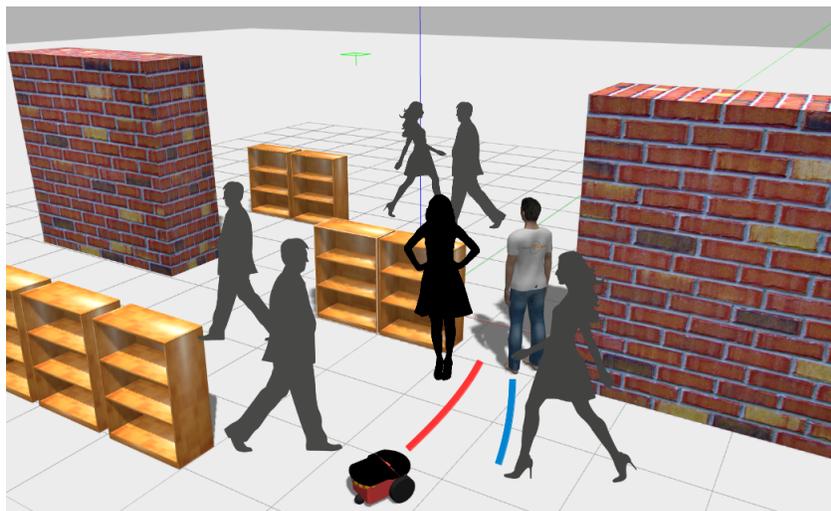


Figure 1.4: A person moves in a indoor office through a narrow path between a wall and a bookcase to generate the reference recorded trajectory (blue). The PFR tries to find a collision-free path (red) with several people moving in between.

1.3 Objectives

The primary objective is to design a new control law using MPC in an integrated manner using non-linear control of non-linear systems, often with non-linear constraints. A differential-drive wheeled mobile robot with non-holonomic constraints poses a great challenge in terms of control, and we address this in this work. While vision-based solutions [5,6,9,10,11] of tracking and identification, mapping, and localization are in abundance, designing controllers that respect such requirements is scarce. We address this for real-world problems [1, 2, 3, 4, 6, 12, 37, 39, 42, 43] for shared autonomy with human interaction with robots so that such controllers can be implemented in such vehicles, resulting in better performance tailored to their operational needs.

1.4 Organization

The prelude in Chapter 2 consists of the literature survey into person following behaviour and mobile robots, and controllers used to implement such behaviour. The scarcity of complete end-to-end systems is alarming given the numerous immediate applications of such robots. We organize the later part of the report into 3 sections based on the progression of this research work and its publication in online media. Chapter 3 deals with dynamic target tracking and collision avoidance behaviors of a person following robot and the baseline for single-step MPC. The next Chapter 4 deals with the utilization of local map information for developing a strategy called Early Relocation (ER) and n-step-MPC to improve the tracking by prioritizing potential corner cases like intersections, and blind corners. This is important because these T, L-junctions are the main reasons for losing sight of a person, and more

so during crowd. The final section and Chapter 5 speak about using recurrent neural network models like LSTM (Long-Short Term Memory) to learn to predict the trajectory of a person(s) a few seconds into the future so that the MPC can follow the predicted goal locations and avoid missing the person in crowd and corner cases like above. This shows a better way to deal with such scenarios than ER when map information is unavailable.

1.5 Contributions

The contributions of this work are as follows.

1. The multi-layered MPC allows it to abandon tracking the target person in every frame, autonomously navigate to vantage locations using ER, and meet the person again to resume normal tracking. It is the first of its kind attempt in literature in realizing a natural person following behaviour. It reduces computational cost as MPC is an online optimization algorithm.
2. We developed a robust person following behaviour inbuilt into a single holistic MPC controller. It complies with social norms, and kinematics of a differential-drive P3DX, and performs in-view tracking in real-time by setting constraints of the above optimization.
3. Our formulation can deal with static obstacles, constrained spaces like corridors, single dynamic obstacles, multiple dynamic obstacles, and various types of complex maps with low complexity.
4. Introduced LSTMs as social path prediction models for a long-term person following in crowded dynamic environments, and track a predicted path rather than an actual path for coupled motion-planning and target-tracking. It achieves off-view tracking for short durations during occlusions and abandons in autonomous mode.
5. The MPC can track human movement patterns of various shapes S, L, Z, maneuver intersections like L, X, T, blind corners, and adjust to varying human walking speeds, in real-time.

Chapter 2

Person Following Robot

In this chapter, we delve deep into the literature around person following robots and controller design for the same. We cover the latest works spanning mostly between 2011 and 2021, and the current state of the art.

The applications of autonomous robotics have a niche field involving human and robot interaction. A great number of attempts have been made in developing social vehicles such as wheeled mobile robots for autonomous carrying systems [1,3] in public places like shopping malls, assistance for elderly [2, 4] and disabled [8], standing in queue [3] and many more. These person following robots [1, 2, 3, 4, 5, 6] must perform two tasks beautifully. First, tracking the target person, and second, avoiding collisions on the way. It means the person following robot or PFR has to do localization, mapping, motion planning, or navigation all at the same time. But, without controller hardware and software, it is not possible to control the system or vehicle. Such a controller must have some definitive properties.

1. It can handle multiple inputs (e.g. distance, velocity, waypoints) and multiple outputs(e.g. optimal velocity and bearing), with interactions between the inputs and outputs.
2. An increase in inputs and outputs should not increase the number of controllers.
3. It can handle constraints (e.g. PFR's speed limit, bounds for acceleration).
4. It has preview capability to update controller performance using new data (e.g. upcoming curve, occlusion of target person, and collision).

Assisting human beings in dynamically changing environments in urban areas is still an active area of research. And, we focus our work on urban structured indoor environments only, but it can be extended to outdoor environments as well with relative ease. The most popular control, which MPC is, is the current best way to control processes while satisfying constraints. Using PID or proportional integral differential controllers is challenging because they can't handle such needs. For example, in multiple input-output systems, they work independently and don't allow interactions. A lot of works [12, 13, 14, 15, 16] involve MPC for dynamic obstacle avoidance, path tracking, and dealing with non-linear constraints.

MPC is a feedback control algorithm that uses a model (e.g. non-holonomic differential drive two-wheeled mobile robot) to make predictions about future outputs (e.g. position, velocity, orientation) of a process by iteratively solving an online optimization problem subject to some constraints at each time step. Because of this, MPC requires a powerful fast processor with large memory. It is a multivariable controller that takes into account interactions between input and output system variables. It even incorporates future information to improve current controller performance on the fly. Thus, MPC controllers and formulations are the current state of the art for most practical applications. We, thus, aim to design an MPC controller for a social person following robot.

2.0.1 In-View Tracking

Vision-based tracking systems [1, 2, 5, 6] are quite mature today in terms of person detection, localization, and tracking. These [10,11] can even handle partial occlusions in the FOV of the target person, including person re-identification [4, 9] which is a challenge in itself. Because, after a partial occlusion of losing sight of the person momentarily, continuing tracking means, re-identifying the same person using deep features. When cameras [2, 5, 10] are unavailable, monocular or stereo, laser-based vision [1,3] is useful. However, none of these approaches discuss the controllers used to implement the same. Some works that utilize ROS (Robot Operating System) for real-world experiments [1, 3] use non-holonomic vehicles like Clearpath Husky and are completely autonomous. They rely on a very simple proportional controller or P-controller (of PID) that simply adjusts the vehicle velocity in proportion to the distance from the target person. The orientation and distance are available from the laser scanners and the optimal velocity commands can be easily passed to the Husky through ROS *geometry_msgs* message. However, the key argument in deriving a control law is to compute the optimal velocity commands in x and y directions, and angular velocity to orient the vehicle at each time step.

While a lot of work has been done in a human detection module that can give the 3D coordinates of the person in view, there is little work in designing a controller that can use that information and relay it faithfully to the vehicle. Since ROS is the worldwide standard for wheeled mobile robots, we also use the same as the baseline. Since the person tracking systems do not materialize it with a controller design that can carry out the task, handle the constraints, the literature is scarce for such complete packages. Another area of research similar to this is where an end-to-end system is available in the case of UAVs (unmanned aerial vehicles) like quadcopters. Person tracking in UAV is developed for holonomic motion vehicles and thus controllers designed for them cannot be applied to non-holonomic motion vehicles, e.g. cars.

A few of the recent works that have brought human-robot interaction to design person following robots include mostly assistive or service robots. These include autonomous shopping carts [1], helping the elderly [2, 4, 8] for walking. Such robots are different than complete autonomous robots whose main objective is not in assisting a human but completing the tasks on its own. For example, Amazon's Kiva bots in warehouses are simple line followers. But works such as PeTra[3], Piaggio's Gita,

Thus, all the above works assume tracking people as long as no ambiguities occur such as loss of sight. Further, they do not elaborate on the controller design relying on simple P or PID controllers in cases where real vehicle tests are carried out. This does not suffice the requirements for a person following a robot.

2.0.2 Off-View or Predictive Tracking

The tracking of a specific person is an indispensable ability for almost any robotic application, where they operate around humans and are mobile. The home robot pets like Sony's Aibo, service robots have to demonstrate that they have specific skills to allow them to interact with the environment as well as their human user. This means tracking when they might be momentarily out of sight and reuniting with them after some time at a probable meeting point. For example, traversing a crowded bend or intersection where the PFR can cross on its own and meet the person after the bend, even though it abandoned tracking it in every frame right before the bend appeared. We can reduce the search space for re-identification to just relevant hypotheses using spatial constraints, social constraints. The social constraints dictate how a human being would traverse a location in crowded dynamic environments.

The first of its kind of work is demonstrated in [28] and [29] where authors consider autonomous driving and controller design for maneuvering critical points on roads like intersections and crossroads. For example, knowing a critical turning point before a turn emerges helps the MPC to pre-calculate the acceleration and orientation "n" seconds/steps into the future. [28] uses an observable Markov process for its low-level planner while [29] uses a priority-based decision making to resolve conflicts. Thus, it uses position, velocity, and orientation as inputs to compute an optimal velocity reaching the intersections.

This forms our foundation to divide an environment into probable points of losing sight of the person. We further demark various types of intersections like X, Y, L and show their performance on them. We also derive the concept of a low-level planner and a high-level candidate early location ER generator at such critical points. This does not require expensive neural networks but map-based information. Further, we make it robust with predictive path generation in combination with location generation using LSTM (long short-term memory) to pass to a low-level MPC controller. Thus, we build our complete end-to-end person following system. And, this so far is the first of its kind to the best of our knowledge.

This is the novelty of our work, where we move beyond trivial in-sight tracking. Thus, our controller design also takes into account such scenarios in real-time. And, certainly, a simple P or PID controller isn't sufficient.

2.0.3 Summary

The thorough study of various practical implementations and simulation studies of the person following behaviour in mobile robots reveals a lack of interconnection. The literature can be divided broadly into two separate groups of extensive research. First, vision-based tracking, and identification studies. And, second, is the motion planning problem in robotics. But there is the hardware aspect where sen-

sors, vehicle kinematics, and dynamics make a huge difference in the approach. A very small section of studies deal with control laws that respect the above, and even smaller for real-time optimizations for agile behaviour. The applications we consider here have their own set of simplifications and challenges.

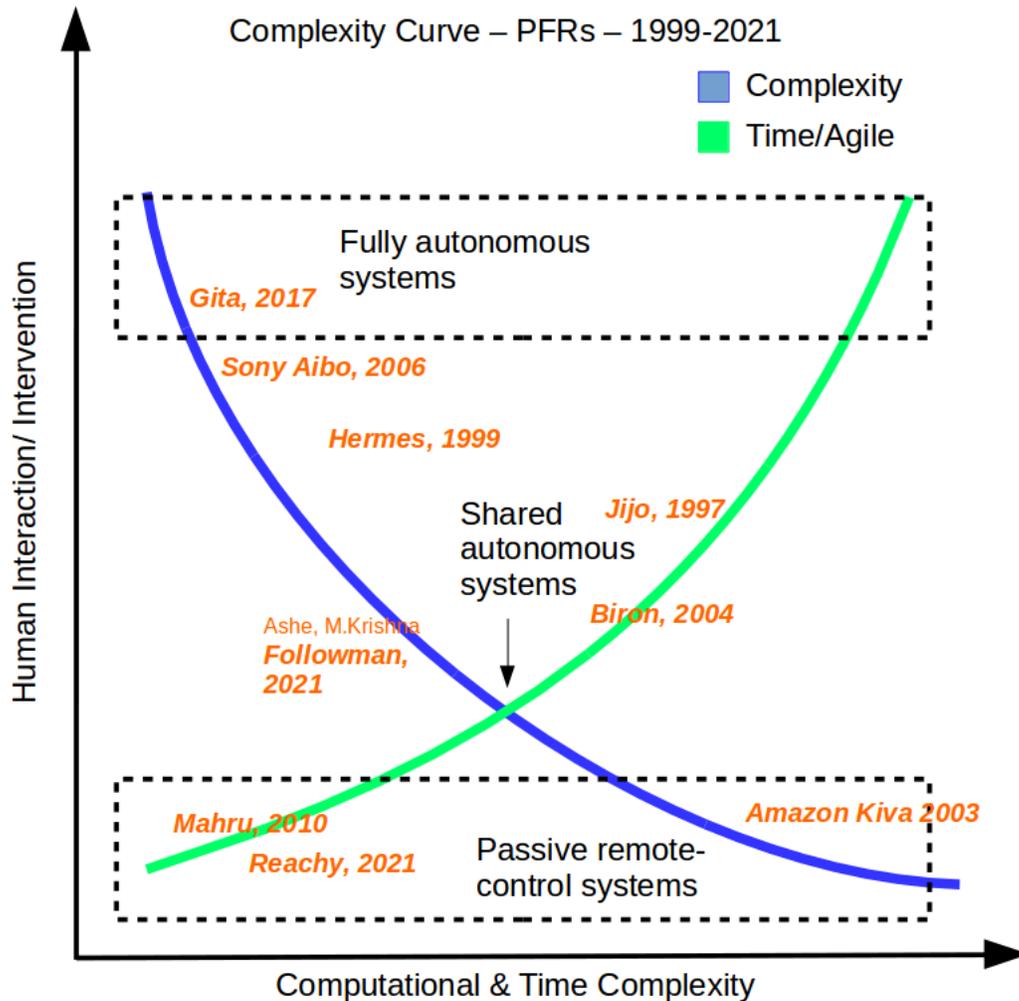


Figure 2.1: Studies from 1999-2021 of developing person-following robots. Evolution of human interaction vs system performance in agility and intelligent behaviour.

1. Complete autonomous systems are unnecessarily complex for such applications and not scalable or totally impractical in many cases.
2. A human-in-the-loop simplifies things and allows for natural service robots, to operate alongside humans.
3. Complete passive remote control systems are too mundane and incapable of any intelligent behaviour or assistance.

Thus, we try to connect a specific application to controllers for the same, and add to the research. Fig. 2.1 shows a graph of human interaction vs time and computational complexity of various works in the past two decades. It highlights the biggest milestones in person following robot study and implementation. The green curve shows real-time solutions using human-in-the-loop. The blue curve shows complexity of systems with human interactivity. For example, *Reachy 2021*, *Pollen Robotics* is a highly agile real-time but human-controlled robot but *Piaggio's Gita, 2017* is a highly complex person follower with least human interaction. While passive systems are under complete human control and need full interaction, they are not useful in many cases. Fully autonomous systems demand too much but require little to no intervention, are very specific, and do not generalize well. Our work, *Followman 2021*, achieves a balance, with feasible real-world controller design. It goes beyond trivial person following and brings forth off-view tracking that has not been attempted so far.

The balance is ideally in semi or shared autonomy, and we solve this using an MPC controller using basic hardware and vehicle dynamics that make it scalable and implementable. This is where our work stands among the available literature, adding an important bit to the void.

Chapter 3

Controller Design

Tracking a dynamic target, a specific person in this case, and generating collision-free trajectories in a coupled manner is the prime objective of developing a robust person-following behavior. Without focusing on the system or vehicle dynamics, we can still proceed with developing the controller framework because this can directly be translated into a desirable vehicle model at a later stage. The MPC controller focuses on two things in this case: **a.** target-tracking and **b.** collision-avoidance. The challenging task is to include dynamic actors in the scenario, and gradually increasing the complexity and testing the optimization to converge gracefully by obeying all the necessary constraints.

Because MPC has the ability to incorporate future predictions, vehicle kinematics, and non-linear constraints, using fast hardware and interior-point algorithm we can show that the optimization converges in real time too, albeit in a simulated environment. MPC sends velocity commands to the PFR with an adequate lower margin of 20Hz, and even adapts to various types of human movement patterns.

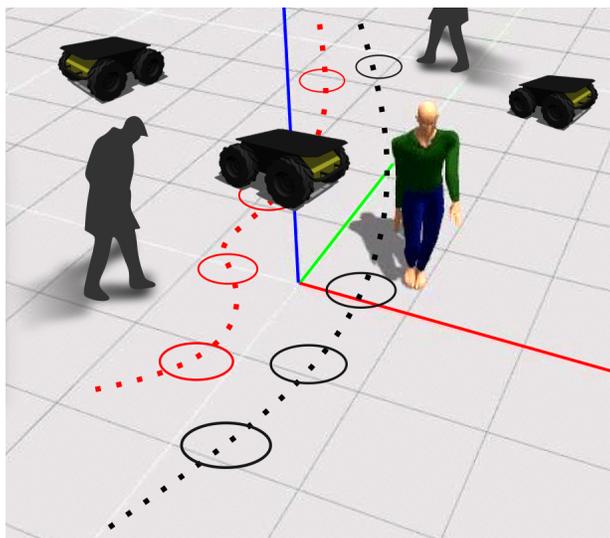


Figure 3.1: A Gazebo world snapshot. A person moves in a indoor scene with humans and traces a trajectory (black) with circular nodes as way-points. The Husky bot follows it safely (red).

In this work, we first review the literature in terms of tracking and obstacle avoidance for non-holonomic wheeled robots that can perform as a PFR, and review the MPC frameworks suitable for driving such a vehicle's controller.

3.1 Related Work

The ability to track a specific target and identify a person, localize and move around in a map in 3D is a computer vision problem where sensors like camera, lasers come into the picture. But, from the perspective of a PFR, a simple mobile robot, Flaco et al. [7], this can be divided into a three-part problem: 1. Environment perception, 2. Motion planning, and 3. Robot control. We focus here on the 2nd and 3rd part, and assume ideal mapping in the 1st. Further, controller design also can be divided into 3 parts: 1. Sensor, 2. Actuator(System), 3. Controller or computer.

MPC is also known as receding horizon control [12] and it has been implemented for mobile robots using time-delay. While most of the work[5,6] is purely based on vision, they use RGBD camera systems and even laser scanners for depth information. Collision avoidance has seen little improvement for non-holonomic dynamics because state-of-the-art methods like velocity obstacle [13] and potential fields have been extensively studied. In [14] Leman et al. use MPC for combined path tracking and obstacle avoidance using non-linear vehicle dynamics but limit to just static obstacles. MPC is a powerful method because it can deal with uncertainty and incomplete information inherently so that one does not have to explicitly model probability and uncertainty in dynamic environments like Fulgenzi et al. in [15] as it unnecessarily makes it computationally expensive. Thus, recent methods make use of current state-of-the-art predictive control and its variants like non-linear model predictive control or NMPC [16,17,18,19]. The work done by Essen et al [18] and Lim et al. [19] actually brings the obstacle avoidance for mobile robots with non-linear constraints and attempts to solve them mathematically. NLMPC [17] stresses that linearising is not feasible for and controllable, we avoid the same in this work, and directly solve them iteratively with improved convergence rates. The application to the person following scenario is first of its kind, and hence the MPC framework we develop is a first in its category.

3.1.1 MPC for Collision Avoidance & Path Tracking

The use of predictive or dynamic control was first implemented for incorporating constraints for non-holonomic motion models by formulating a single-step framework and using collision cone as a separate module for dealing with obstacles. Next, it was integrated with neural networks in [20] to improve trajectory tracking by solving the quadratic cost function using primal-dual neural network. Elsamir et al. [21] was the first to combine target tracking and obstacle-avoidance into a single MPC framework, but it required the complete map information which is unavailable in this particular application setting. They carried out 2D simulations with MATLAB and reported a convergence time of 7secs for linear velocities for the complete path and 2secs for angular velocity, but because in a PFR application, the

trajectory is not known a priori, such implementation is not applicable. Also, increasing the number of static obstacles further made it worse. We are able to show in our simulations that it performs much better than the previous similar attempts. In fact, relaxing the map information benefits the optimization. Leman et. al. [14] made a good contribution using a single NMPC model for avoiding obstacles on a highway using a 3DOF vehicle. But, once again they do not consider dynamic obstacles, which is where we add to the novelty.

3.1.2 MPC for Person Following Behaviour

A person following behaviour model is less stringent and more flexible in terms of real-time operation and convergence speeds. It is also easier in terms of motion-planning because of the human in the loop which makes it inexpensive than completely autonomous operation. The NMPC adds other constraints in terms of social norms like safe space around the target person, dynamic actors, and other obstacles in its periphery. Because of fast hardware, the non-linearity can be solved iteratively in online fashion without need of linearisation that also preserves the dynamics. Linearization around a fixed point can introduce infeasibility and make the PFR difficult to control, so in a person following scenario we can forego this as a valuable tradeoff. None of the MPC formulations address the shared autonomy concept to this extent or only deal with the vision aspect. In this work, the NMPC models the multi-objective goals in a single holistic optimization and solves it in real-time.

3.2 Novelty

The novelty of this first phase of our work can be summarized as follows.

1. We develop the MPC control law for the novel application of a robust person following behaviour, a first of its kind.
2. We are also first in integrating target tracking and collision avoidance for dynamic obstacles, into a single MPC framework and jointly optimizing the linear and angular velocity as a multi-objective control
3. Our proposed non-linear MPC framework and solve it without explicit linearisation to avoid infeasibility and untractability issues.

3.3 Optimization

The optimization [22] from Ashe et. al. shows the basic guidelines for a single-step MPC that incorporates the above in a step by step fashion. The objective function minimizes the velocity of the PFR at current time instant and the preferred velocity values as per physical constraints of the bot. For

example, a Pioneer P3DX can move well with 1.2m/s while stressing to 1.6m/s for short bursts. The various constraints are divided into 1. Kinematics, 2. FOV and 3. Collision Avoidance. The kinematics and FOV take care of path tracking while the collision avoidance takes care of safe operation at all times. To understand it better, we introduce the model of the PFR.

3.3.1 Kinematic Model

An extremely common differential drive robot comprises of two powered wheels and a castor wheel. These two wheels (left and right) can be rotated with different velocities to move the robot around. This is called the Unicycle Model, and we base our experiments and simulations using this. Firstly, because it has 2 parameters to control and compute, **a.** forward translational velocity, and **b.** angular velocity. That way we can control both the left and right wheels of a differential drive PFR, and move it around.

$$\begin{aligned}\dot{x} &= R/2(v_r + v_l)\cos(\theta) \\ \dot{y} &= R/2(v_r + v_l)\sin(\theta) \\ \dot{\theta} &= R/L(v_r - v_l)\end{aligned}\tag{3.1}$$

where, v_r, v_l are right and left wheel forward velocities in m/s, and θ is the orientation with the global coordinates. Similarly, \dot{x}, \dot{y} are the velocities in each coordinate axis. However, as we shall soon see, non-holonomicity constraints movement perpendicular to the wheel base and the final translation is produced only by the forward component in direction of θ . But, for now, we keep it simple.

This brings us to the following way of rewriting the Unicycle Model, which uses just two parameters to be controlled, and is also a more natural way of driving the robot.

$$\left. \begin{aligned}\dot{x} &= \mathbf{u}\cos(\theta) \\ \dot{y} &= \mathbf{u}\sin(\theta) \\ \dot{\theta} &= \omega\end{aligned}\right\}\tag{3.2}$$

where, \mathbf{u}, ω is the control-space of the PFR, that controls its state-space $[x, y, \theta]$ at any point in time t .

$$(v_r, v_l) = \left[\frac{2\mathbf{u} + \omega L}{2R}, \frac{2\mathbf{u} - \omega L}{2R} \right]\tag{3.3}$$

where the right side are the control inputs that MPC optimizes for generating a collision-free trajectory. That is, it yields a set of v_x, v_y for each time step, t , as it follows the target person.

3.3.2 Cost Function

We begin by minimization of the difference between the current linear velocity of the PFR and the desired velocity. At all points in time, we drive the robot to follow the person's reference trajectory but consider its physical limitations. The reference trajectory is the human motion model or movement

pattern following a constant velocity model. This results in a lead-follow configuration where the control law guarantees that the PFR always stays behind the person.

$$\begin{aligned}
& \underset{v_x, v_y}{\text{minimize}} && (v_x - v_x^{pref})^2 + (v_y - v_y^{pref})^2 \\
& \text{subject to} && 3(a) : \text{Kinematic Constraints} \\
& && 3(b) : \text{FOV Constraints} \\
& && 3(c) : \text{Collision Avoidance Constraints}
\end{aligned} \tag{3.4}$$

where, v_x^{pref} and v_y^{pref} are the preferred forward velocities along (x,y) permissible by the physical constraints of the bot.

The different type of constraints are divided into the following sections to make more sense of it and justify their formulation.

3.3.3 Constraints

The constrains are the basis of subjecting multi-objective model predictive control where the main objective function is the mandatory requirement and the other goals are treated as non-linear constraints. The first of its kind comes from the motion model of the PFR itself, and is called the Kinematic Constraints. The PFR must use to to derive the next state information.

$$\left. \begin{aligned}
x_{t+dt} &= x_t + v_x \cos(\theta_t) dt \\
y_{t+dt} &= y_t + v_y \sin(\theta_t) dt \\
\theta_{t+dt} &= \theta_t + \frac{d\theta}{dt} dt
\end{aligned} \right\} \tag{3.5}$$

which may be relaxed to the following holonomic form for ease, such that $\Delta t = [t_{i+1} - t_i]$

3.3.4 Kinematic Constraints

$$\begin{aligned}
x_{t+1} &= x_t + v_x \Delta t \\
y_{t+1} &= y_t + v_y \Delta t
\end{aligned} \tag{3.6}$$

where, $\frac{d\theta}{dt} = \omega$, the angular velocity. The limits on these control variables are $-1 \leq v_x \leq 1, -1 \leq v_y \leq 1$, and $-2\pi \text{ rad/s} \leq \omega \leq 2\pi \text{ rad/s}$.

This allows us to move the robot around from one time step to another, in discrete mode.

3.3.5 Field of View Constraint

This is essential to maintain the target person inside the frame, which is possible by maintaining a direct line of sight from the PFR to the person's last recorded position. Then, assuming some error

margin about the line of sight, we allow a tolerance angle. This is modeled in the following equation as a non-linear constraint.

For this, we can assume a 360° FOV as a generic case and some arbitrary $\pm\beta^\circ$ a more specific case.

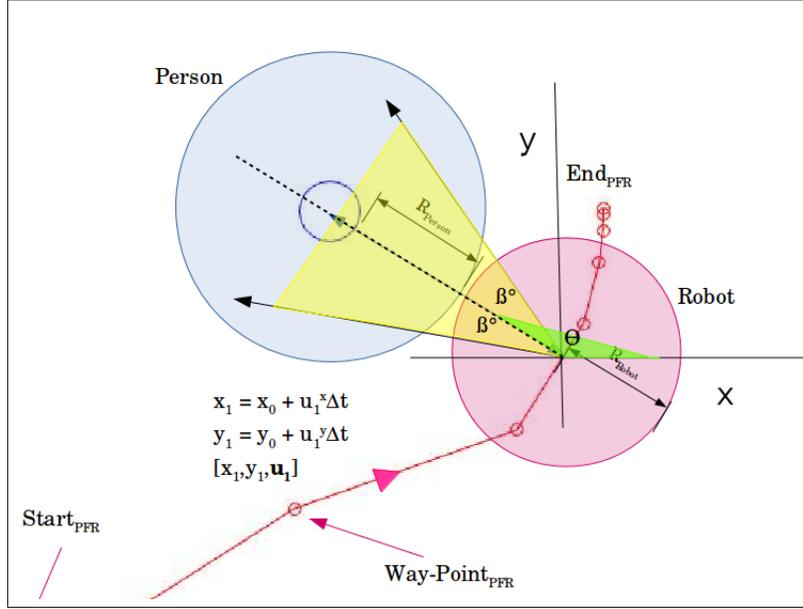


Figure 3.2: FOV constraint for an arbitrary angle β

$$b^2 \leq (x_{t+dt} - x_{t+dt}^{person})^2 + (y_{t+dt} - y_{t+dt}^{person})^2 \leq a^2 \quad (3.7)$$

for 360° FOV, where a^2 and b^2 are the lower and upper distance bounds on the robot. In other words, it must stay within this distance limits to not lose sight of the person. Now, if we assume the PFR has an onboard sensor that scans α° on either side of the line joining the current position of the robot and the person, it can be re-written as so,

$$(\theta_{bot} + \beta)^2 \leq \tan^{-1} \left(\frac{y_t^{person} - y_t^{bot}}{x_t^{person} - x_t^{bot}} \right) \leq (\theta_{bot} - \beta)^2$$

where, the global θ from (4) must be added to convert the above angles to world frame.

3.3.6 Collision Avoidance Constraint

Because this is not a single constraint but a set of equations depending on the number of obstacles or nodes considered inside the sensor radius, one must pay caution when implementing it in code. These sets of constraints conform to avoiding static and dynamic obstacles in the perceived periphery or sensor radius of the PFR. In an ideal case, we assume perfect mapping and availability of their global position information.

$$x_{t+dt} - \begin{pmatrix} x_t^{obs1} \\ x_t^{obs2} \\ \vdots x_t^{obsN} \end{pmatrix} + y_{t+dt} - \begin{pmatrix} y_t^{obs1} \\ y_t^{obs2} \\ \vdots y_t^{obsN} \end{pmatrix} \leq (r_{bot} + r_{clear})^2 \quad (3.8)$$

where a generic form of circular clearance radius of r_{bot} and r_{clear} is assumed for the PFR and obstacles, respectively. So, any obstacle on the impending course is a sample of points on its periphery, if static, or a sample of points on its trajectory, if dynamic. In case of a PFR, its sensor radius R_{Sensor} is the perceived boundary (if 360° FOV is considered). With a circular boundary, we impose convexity in the solution space.

3.4 Results

We setup the problem in a 2D workspace, and perform simulations using Matlab R2017a. The `fmincon` solver uses interior-point method which is suitable for small but dense problems like this, and solvable in polynomial time. Because of fast hardware, we run it on an Intel Core i5, 7th Gen CPU in single-core, and check its runtime performance. The scenarios grow in complexity in terms of number of obstacles, type of obstacles, and human movement patterns from simple, short to lengthy and complex. By this simulations, we test agility of the controller, violation of constraints, and overall performance.

3.4.1 Simple Person Following Behaviour

Fig. 3.3 illustrates an obstacle-free scenario where the objective is to track the person without colliding with it. That is, maintaining the person in its FOV from a safe distance. We assume a 360° FOV in this figure, so do not show the bearing markers.

We use this first example to explain how the PFR tracks the person from one node to the next one. For this, we focus on the green rectangular region of interest.

It (Fig. 3.4) shows the transit from *Node0* to *Node4* as the person and PFR move in that order, downwards. We select three nodes, $t = k - 1$, $t = k$, , which is the current node, and $t = k + 1$. As the PFR tracks down, shown by a solid red line, it passes through the circle of *Node1* towards the periphery of current *Node2*. The MPC uses the (x, y) of this node as the goal and issues the computed (v_x, v_y) for the same. This means that only the boundary of *Node2*, highlighted by solid blue concentric circles, is active, and the other circles disappear clearing the space for free movement. Despite it seems that the red line passes through the circles violating the constraints, it does not, because the other circles (for *Node 0, 1, 3*) simply cease to exist at $t = k$.

3.4.2 Bending Around Corners

This figure (Fig 3.5) illustrates a scenario where the person bends around a corner. The multi-colored line segments are walls/obstacles depicting a typical urban indoor setting. While both start roughly at

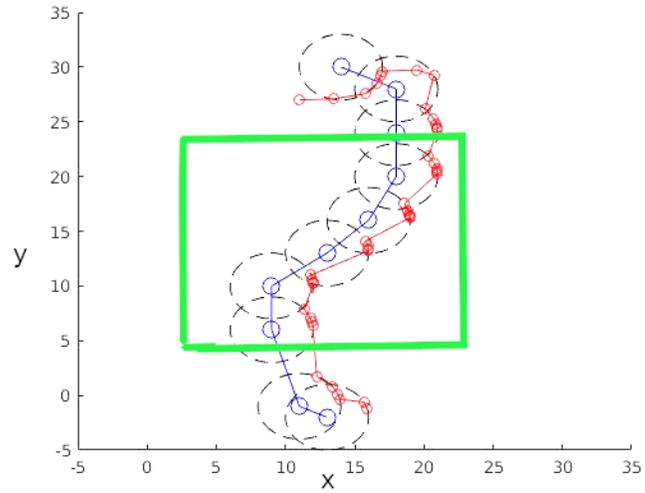


Figure 3.3: Simple piece-wise straight line/curvilinear path

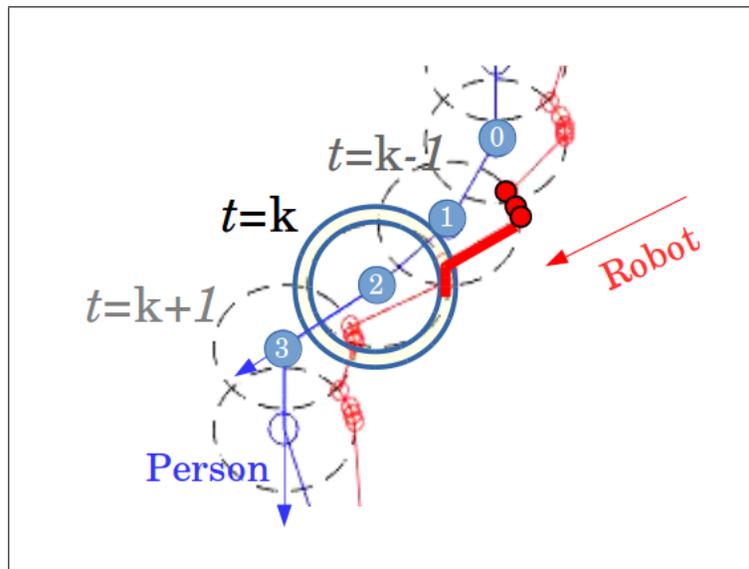


Figure 3.4: Node-to-node transition strategy

$[0, 0]$, they execute their first right bend together at $[5, 12]$ and the second (left turn) at the exit after crossing the corridor, terminating at $[15, 15]$.

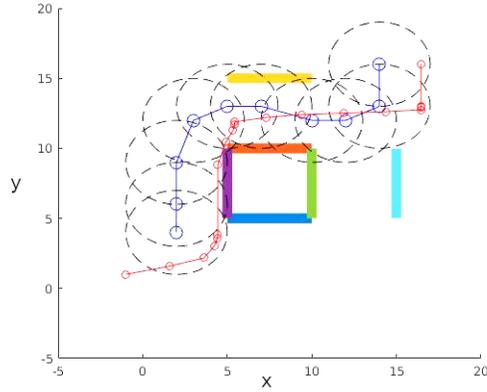


Figure 3.5: Indoor setting with walls, a square room, executing two bending around corners

3.4.3 Multiple Consecutive Bendings & Corridors

This figure illustrates the ability of MPC to react quickly with a real-time response rate. This agile behaviour demonstrates and loosely mimics a human-like natural person-following characteristics. Here (Fig. 3.6), we show three walls making up two long corridors. These serve as static obstacles that are unknown prior to detecting it. While the path begins coarsely at $[15, 22]$ (top-center) with the bot(red) right behind the person(blue). The PFR moves through the corridor, takes a right turn around the green wall, falls back soon after, and takes another right turn around the purple wall, and continues. As shown, the MPC reacts with agility, waits for some time to let the person complete the turn, and keeps following.

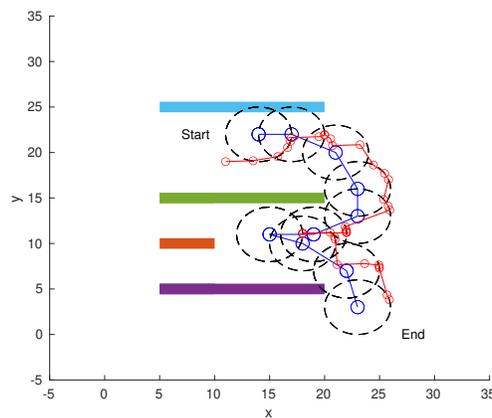


Figure 3.6: Multiple long corridors, hallways, executing consecutive turns.

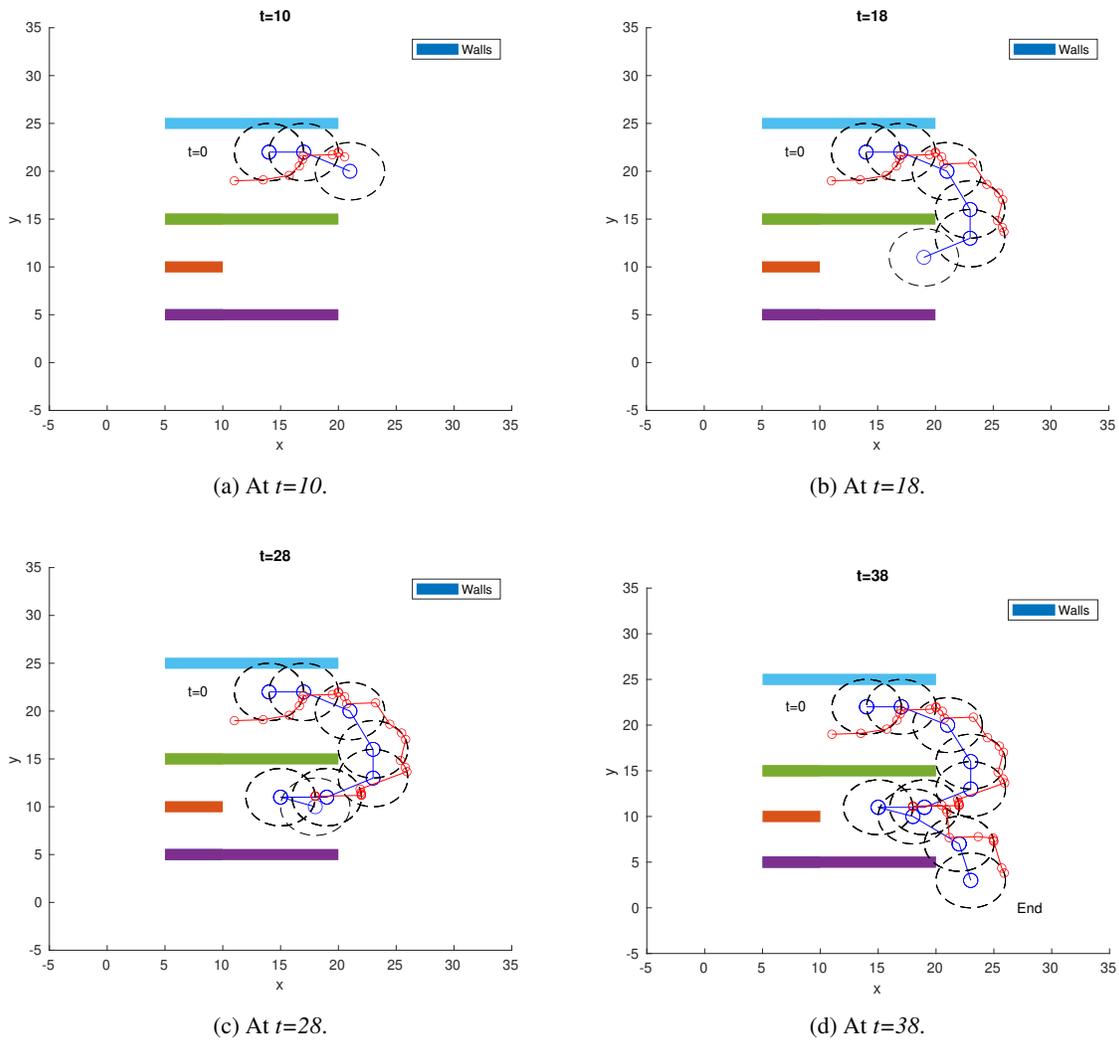


Figure 3.7: Space-time plot of Fig 3.6 with positions of Person and PFR at $t=10, 18, 28$, and 38 .

To make the visualization more elaborate, we show the below space-time graph of the same. The above optimization proceeds through several internal and external iterations. Each external iteration corresponds to a time instant t , and 3.7 shows four such time instants.

3.4.4 A Dynamic Obstacle & Moving Person

In this case we consider that the position of the obstacle also varies with time, instead of being fixed as in previous case. This modifies the MPC formulation with an additional set of constraints, that vary depending on the proximity to the obstacle at time t .

The dynamic obstacle, bold green, follows a trajectory that crosses the person around $(12, 7)$. The plot captures a moment when the person is just about to cross after the obstacle, another person or bot,

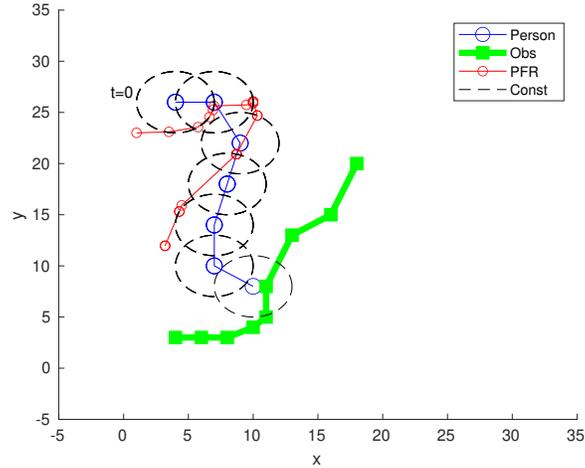


Figure 3.8: One dynamic obstacle crosses the person, shown at $t=26$

has passed. While, the PFR waits for it to follow positioned around $(4, 11)$. Here, the PFR obeys the collision constraints with both the person and the moving obstacle. We can see how this evolves over time in the following space-time plot for the same scenario.

3.4.5 Static & Moving Obstacles Cross Person's Trajectory

In the next scenario, we consider two such situations during a collision course.

The above simulation (Fig. 3.10) illustrates a dynamic obstacle in the form of another person or another PFR in the vicinity. As the obstacle, shown in green trajectory, crosses paths with the target person, the bot waits it out from a safe distance and continues to track the trajectory thereafter. A PFR waits if an immediate solution is unavailable for up to the maximum distance from the person passes or if the person is still. As we can see at $(15, 10)$, when a solution is available, it curves and continues following the target person.

3.4.6 FOV Analysis

We show here an example of "with and without the FOV constraint", that is, with 360° FOV vs. a smaller arbitrary FOV. We shall see in Section V-G (Fig. 3.14, Page 26) that despite adding to runtime, it's still agile. In this example, we consider an "S" pattern human movement where all parameters like trajectory length, nodes, number of hopping vertices are kept constant.

With a 360° FOV, bearing markers are unnecessary. Here, we compare the bearings of three PFR nodes as it tracks the trajectory of the person from $Node1$ to 2 from $t = k - 1$ to $t = k$. As $t = k$ is the currently active node (goal (x, y) for the MPC), and the three red dots marked 1, 2 and 3 with bearing markers in dotted-red lines are the PFR's tracks, we focus on them. We see something noticeably different between these positions in the top and the bottom figures.

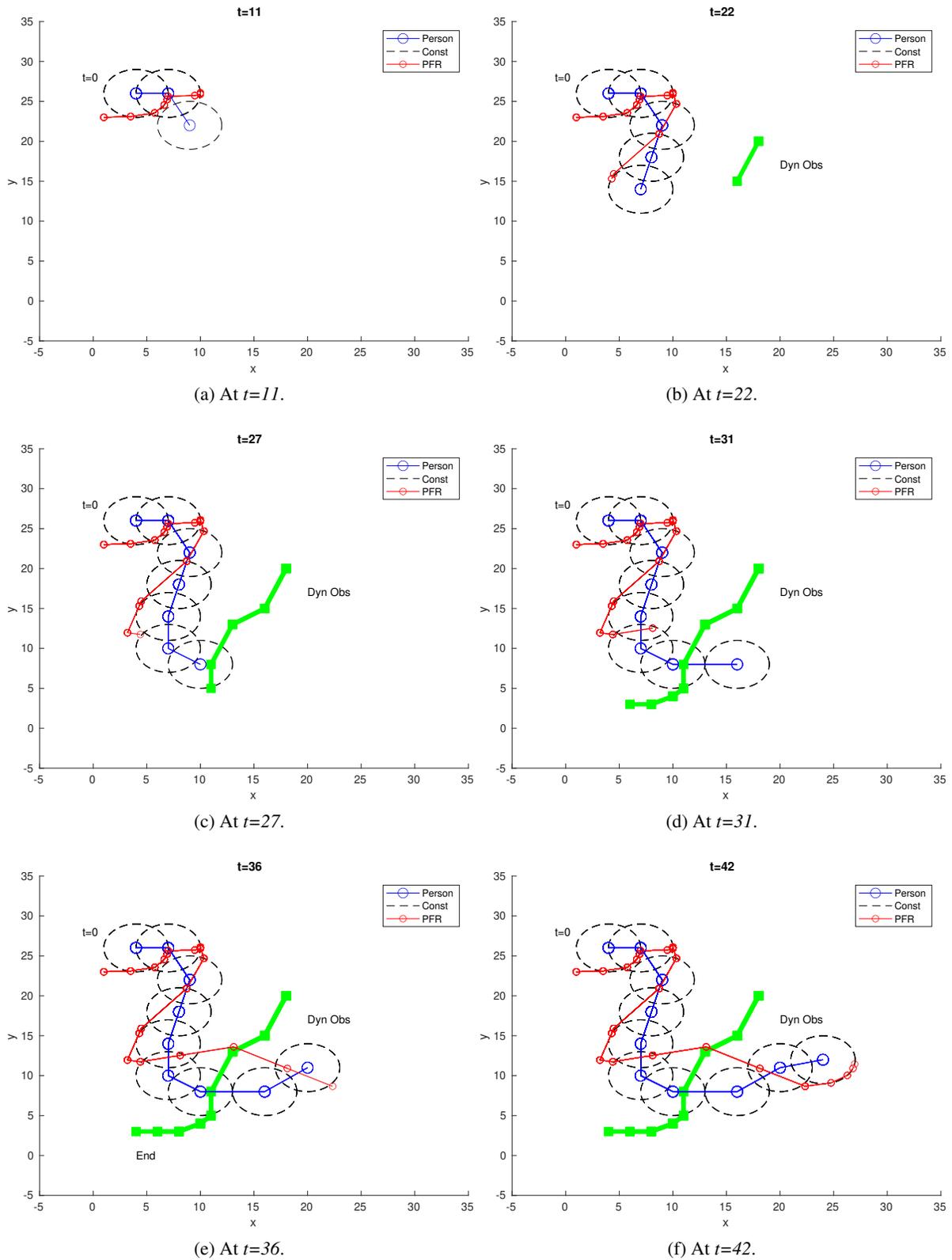


Figure 3.9: Space-time plot of Fig 3.8 with positions of Person and PFR at $t=11, 22, 27, 31, 36, \text{ and } 42$.

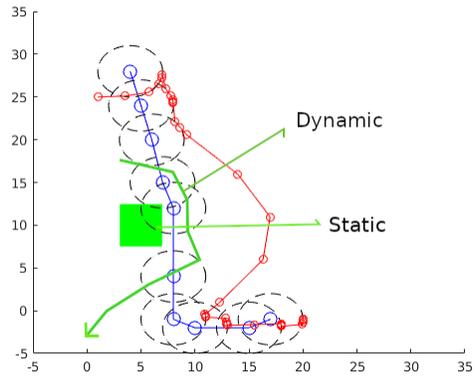


Figure 3.10: Scenario with both static and dynamic obstacle

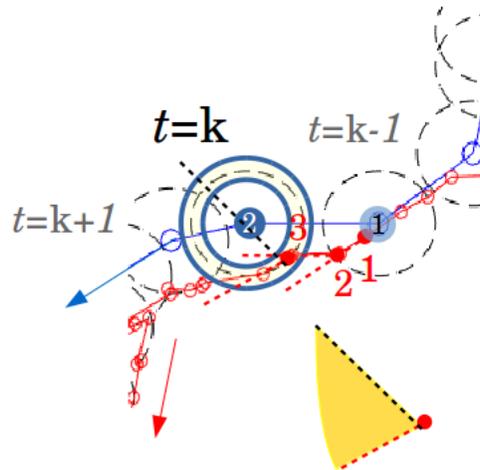


Figure 3.11: Trajectory with 360° FOV

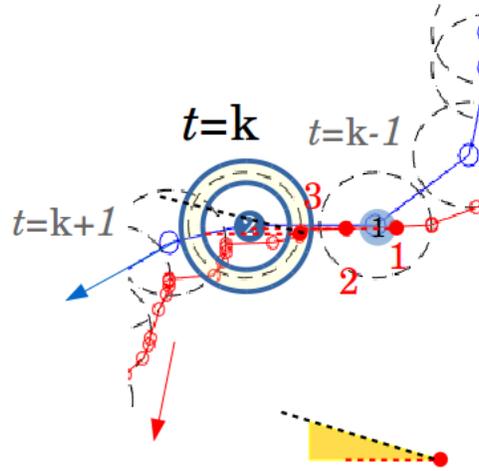


Figure 3.12: Trajectory with 30° FOV

The difference between the angle of the red bearing marker in *Node3* of PFR and the black bearing marker (desired angle for direct line of sight) is greater than $\pm 15^\circ$ (Fig. 3.11). So, we can verify that the FOV constraint was not applied here. Hence, it stays in a safe distance but with arbitrary bearing. However, in Fig. 3.12 we can see that the bearing marker of the *Node3* of PFR makes a very small angle with the line of sight of *Node2* of the person, confirming that the FOV constraint was applied here. That is, from a set of permissible (v_x, v_y) the MPC chooses a pair that not only tracks the person but also adjusts its bearing to force the line of sight within tolerance ($\pm 15^\circ$ in this case).

3.4.7 Quantitative Analysis

When designing a controller, the runtime is an important criterion. A faster runtime means quicker response and agile person-following behaviour. Each time MPC computes the control variables, it does so in a finite number of iterations and time. During a path-tracking trajectory, there are several such computes, also known as planning and replanning. Below (Fig. 3.13) on Page 26 we show the time consumed per MPC computation during a curvilinear (S-pattern) path-tracking scenario. There can be several other types of human movement patterns as well, however, which we shall soon see.

We repeat the simulations 10 times for each obstacle scenario. The minimum time per plan (compute) for different number of obstacles average out at around 0.025s. While, the trend is noticeable by looking at the median values, which is a statistical measure dormant to outliers, the average runtime is skewed in this figure. Because 4 and 7 obstacle-scenarios seems to be outliers, that pushes the mean out of the trend, while the rest of the values show a steady-state rise. The median shows that as the number of obstacles increases, there is an increase in runtime as well. It increases from 0.02s to 0.2s, a 10x increase.

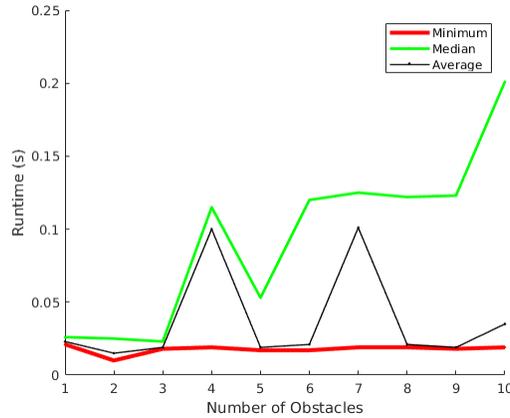


Figure 3.13: Number of Obstacles vs Runtime per MPC Computation

In another case, a set of different human movement patterns were considered. These were taken from pedestrian movement pattern datasets where a straight line (type 1) is most preferred and convenient (Fig. 3.14) while type 6 is most unusual. They were divided into 6 types and each scene was repeated (non-identically) with a fixed number of hopping vertices and obstacles 10 different times. So, a total of 60 simulations were performed. Then, the average minimum (& maximum) runtime per compute for each trajectory type was recorded.

Human Movement Pattern (10 Hops)	Avg. Min Runtime (s) (10 Rounds) 360° FOV	Avg. Max Runtime (s) (10 Rounds) 360° FOV	Avg. Median Runtime (s) 10 Rounds ±15° FOV	Avg. Median Runtime (s) 10 Rounds ±45°FOV
1	0.010	0.013	0.017	0.015
2	0.014	0.018	0.037	0.027
3	0.019	0.021	0.030	0.024
4	0.011	0.015	0.018	0.013
5	0.017	0.120	0.032	0.019
6	0.021	0.048	0.037	0.027

Figure 3.14: Human Movement Patterns vs Trajectory Tracking Runtime

The chart (Fig. 3.14) confirms that with a significant increase in complexity there is a noticeable increase in the per compute run-time. For example, 0.018s to 0.021s for the "S" pattern. This translates to a replanning frequency of 50Hz. Secondly, the runtime for a specific type of trajectory fluctuates

between a range too. For example, the "V" pattern takes between 0.011s and 0.015s per MPC plan during the entire trajectory. This translates to a replanning frequency $\simeq 100$ Hz.

For a specific trajectory type, the runtime for 360° is usually lower while the more constrained 30° or $\pm 15^\circ$ FOV is the highest, though for easier trajectories the difference is not significant. For example, the "S" pattern has a median value of 0.019s for 360° FOV, surpassed by 0.024s for 90° FOV, even surpassed by 0.030s for 30° .

To summarize, there are several other factors that can affect the results, such as number of hopping vertices, distance between them, and so on. For the above metric, they were kept pretty much constant. Overall, the MPC performed with an upper margin of 0.05s per plan, resulting in a replanning frequency of 20Hz, across trajectories.

3.4.8 Ablations

To identify the violation of constraints with changes in the complexity of scenarios, we consider the following chart (Fig. 5.7). The trajectory, as in number of nodes, length, and the type of trajectory, is kept constant. To ensure that the obstacles interact with the same, a small area is selected and a random generator is used to generate an increasing number of static (and dynamic) obstacles in the scene. Then, the number of collision avoidance (CA) & FOV violations are recorded along with total number of constraints, in each case.

No Violated FOV ($\pm 15^\circ$) Constraints	0/0	0/0	0/1	1/3	1/3	3/4
No Violated Collision Constraints	0/0	3/5	5/7	9/10	11/13	13/16
Interacting Static/Dynamic Obst	0/0	1/1	2/2	3/3	4/4	5/5
Total Number Of Constraints	42x3	84x8	126x9	168x7	210x12	252x18

Figure 3.15: Violation of CA and FOV constraints vs Scene Complexity

In Fig. 5.7, the first row shows the number of violated FOV constraints, the second row shows the number of violated collision avoidance constraints, the third row shows the variation in number of static/dynamic obstacles, and the fourth row shows the total number of constraints. The last value is computed as the *number of collision constraints in the entire trajectory times the avg. number of iterations per MPC compute*. In Fig. 5.8 we show the violation of FOV and collision constraints with the change in trajectory type, for a fixed distribution of obstacles.

To ensure that the baseline is clean, we consider a scenario without zero constraint violations. We then introduce obstacles in the scenario such that it interacts with the target person's path and records

No Violated FOV ($\pm 15^\circ$) Constraints	0	0	1	1	2	4
No Violated Collision Constraints	0	0	3	7	6	10
Human Movement Pattern						

Figure 3.16: Violation of CA and FOV constraints vs Movement Patterns

the changes. As the number of obstacles increase, so does the number of constraints. From these (Fig. 5.7, 5.8) we can infer a linearly rising trend in both FOV and CA constraints. However, the FOV constraint appears to be minutely affected by the variation. For example, in Fig. 5.7 Col3-Row3, we have 5 CA violations for 2 static obstacles. For 2 dynamic obstacles, we get 7 CA violations. In contrast, we only get 0 and 1 FOV violations for static/dynamic respectively.

3.5 Conclusions

In brief, we have approached the problem of person-following mobile robot in an urban setting using a 2D simulated environment. In this paper, we derive the proposed MPC formulation, look at specific constraints that impose walking behaviour traits of a person, such as different trajectories, and distribution of obstacles. To have a fully functional social robot, we have to assist the human-centric goal-reaching behaviour with collision avoidance with obstacles in the environment as well as with the person itself. The final achievements can be summarized as below.

- The formulation solves a single MPC to perform tracking as well as collision avoidance.
- The MPC works reasonably fast with an upper margin of 20Hz across trajectories, which is a lot more than a practical person walking speed.

Future Work

The design of control law for a mobile robot depends on its motion or mechanical behaviour. And, understanding this behaviour starts with understanding the wheel constraints placed on its mobility. A person following robot at a mall will have a different locomotion type (mobility constraints) from that at an airport, a warehouse, a restroom, unpaved roads, or a crosswalk. So, the focus can also be on developing MPCs for each of these specific applications of social robots.

In practical scenarios such as implementing this MPC on a P3DX via ROS, we have to solve the inverse kinematics problem where given the pose $[x, y, \theta]$, we have to derive the steering angle (β) and left-right wheel speeds (Φ), as a function of time (t). At present, we limit ourselves to solving

for the forward velocities $[V_x(t), V_y(t)]$ only, and a future goal of testing it on a real robot is already underway.

Eventually, the long-term goal is obviously to add predictability and out-of-sight tracking to make the person following behaviour as natural as possible.

Chapter 4

Maneuvering Intersections

When talking about human-robot interaction in case of wheeled mobile robots those are supposed to assist their human counterparts, while simultaneously traversing dynamically changing environments, dynamic control is key. The dynamic control is also called model predictive control because it attempts to use uncertain information to make predictions, and change the output continuously to rectify and match the desired output of the system. In here, we apply MPC for integrated motion-planning and obstacle avoidance, in scenarios like intersections that are pretty common in indoor environments. For example, warehouses, offices, homes, and even in the neighborhood, we have T,L-junctions, and even cross-roads. In this phase, we introduce and develop the concept of early relocation or ER that helps to make use of available local map information to give the PFR a better insight to move where and when.



Figure 4.1: A snapshot of real-world intersections. In yellow we show L and T-junctions in a structured outdoor environment.

Our approach ensures that the target person is in the FOV of the PFR as much as possible, during such occlusions and maneuver it smartly. By constantly updating the MPC's reference path, and prioritizing ER locations rather than person-following, the new trajectories are generated in an incremental fashion for the local map. While, the global map corresponds to the final goal location. We build the social

representation of the PFR directly into the MPC as constraints, which we have mentioned before in Chapter 2, but now additionally introduce intersection-specific PFR dynamics, and framework. Thus, a non-linear MPC is developed with ER and tested in several complex scenarios. We report a margin of over 20Hz during even the most crowded scenarios, in our 2D simulations.

4.1 Related Work

The role of perception has been studied well from developing a person-specific follower [5] to real-time target tracking of a person [6]. A categorical overview of state-of-the-art methods for motion planning, perception, and control has been elaborately discussed here [12]. Visual-based planning and control often ignore the kinodynamic constraints of a robot. For example, the Pioneer 3-DX has a preferred linear velocity of 1.2 m/s because of its physical limitations, so the controller should consider this too. In autonomous robotics synthesizing time-optimal kinodynamic solutions, is a long-standing problem. So differential-drive robots with bounds on velocity, acceleration, and turning capabilities that constitute non-holonomic constraints of the system, must be met in minimal or real-time. Incremental path-planning using A* and obstacle avoidance with velocity obstacle and collision cones though help when accurate and complete maps are available, dynamics of the environments require a local controller. The computation costs for recomputation on every step prove infeasible for real-time operation. So, despite its success with holonomic WMRs, it could not be directly translated to non-holonomic motion models.

Motion planning for autonomous driving using sampling-based planners [13] and RRT (Rapidly-exploring Random Tree) [14] are efficient for real-time traversal of non-convex spaces, but its inclusion with a controller design is scarce. For instance, many do not explicitly include obstacle avoidance in the formulation or treat them independently. Therefore, despite the simplicity of the kinematic model of a non-holonomic WMR [15], works on devising a coupled optimal control scheme directly included in the optimization framework are pretty thin. Our work directly addresses this gap.

4.1.1 Motion Planning for Autonomous PFR At Intersections

In urban indoor settings [17], scope of this work, the most frequent cause of occlusion arises at **T** or **L** junctions (Fig. 4.2),B,D,E). However, it can be directly translated to outdoor environments as well. [17] focuses on localization but neither considers such junctions nor human factors. The latest state-of-the-art works employ MDP (Markov Decision Processes)[8, 9] that uses a probabilistic motion model for various possible future measurements and uncertainties of states to develop a policy for optimal acceleration along preplanned paths. [10] focuses on T-junctions using MDP for path predictions, but without ER, it still fails to regain tracking. As mentioned earlier, we study this from a control point of view. And, model the 4-part motion-planning criteria (kinodynamics) of a PFR as constraints of the optimization, to holistically solve for control signals to steer the robot around. To complement this,

we compare a generic one-step MPC without early-relocation (ER) strategy known for high-frequency agility to an N-step MPC controller with ER.

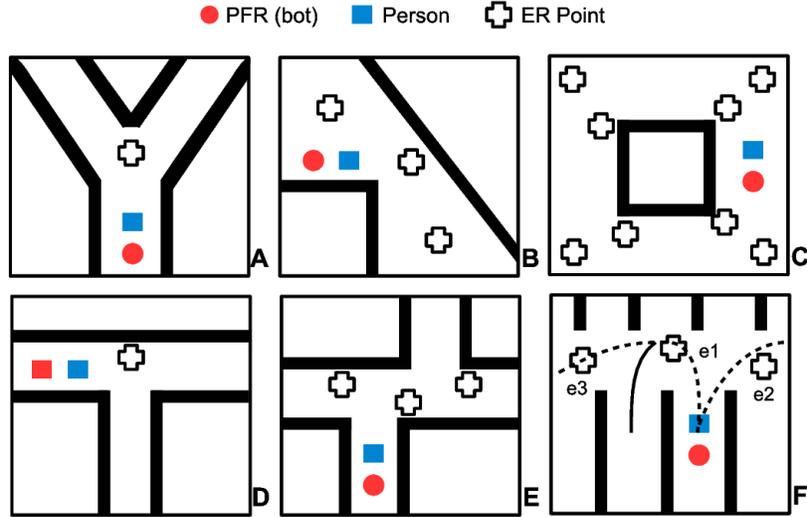


Figure 4.2: Common Y, T, L-junctions in structured urban indoor floor plans, and probable ER locations in various scenarios

Therefore, the major novelty of our work and contributions of the paper are two fold: 1. Develop the concept of Early-Relocation for maneuvering cusps and intersections naturally as a human being. 2. Design an MPC controller that performs motion planning around static and dynamic obstacles with safe person-following constraints.

4.2 Problem Formulation

Consider a single differential-drive wheeled mobile robot and a target person set in a 2-D workspace \mathbb{R}^2 . The set O denotes static obstacles in 2D given by $obs_i = [x^{obs_i}, y^{obs_i}]$. But, the dynamic obstacles are unknown apriori and detected only when they land within the perceived boundary of the PFR (r^{bot}). There can be any number of obstacles which is dynamically chosen and positioned as a user-defined parameter per simulation. The trajectory of a dynamic obstacle, which is also randomly generated as a set of 2D positions is O . The robot's i th pose is given by $X = [x, y, \theta]$ and the entire state-space trajectory is denoted by $X(t)$, where $t = 1 \dots N$. At $t = 0$ both the PFR and the specific target person are at rest at their respective user-defined 2D start positions. They are denoted by $[x^{bot}(0), y^{bot}(0)]$ and $[x^{person}(0), y^{person}(0)]$ respectively. The target person's movement pattern generates the reference trajectory as the simulation proceeds and unavailable apriori. The control actions $U(X, t)$ are sent at a lower frequency than the person's stride.

4.3 Dynamics of Person Following Robot

The unicycle model (Ch3,Eq:1) for a differential drive WMR relies on simply two parameters, the desired forward velocity of magnitude V and direction θ . It directly gives the translational velocity of the PFR along both the axis in the global frame, and ω the angular velocity to turn the vehicle, as control inputs. This is more intuitive from a control point of view instead of the rate of change angle of the wheels and used for the design phase.

4.3.1 Reference Trajectory For Person Following Robot

At time $t = 0$, both the PFR (red filled circle) and the reference person (blue filled square) are at rest (Fig. 4.3). The distance between them at rest is $d2$. The projected distance $d1$ is the allowed head start for the person by k time steps so that a visible reference trajectory is available. At $t = k$, the person completes covering the projected distance till which the tracking is at a halt. The PFR begins tracking at $t = k$ based on the visible reference trajectory Rf as recorded by its sensor radius r^{sensor} .

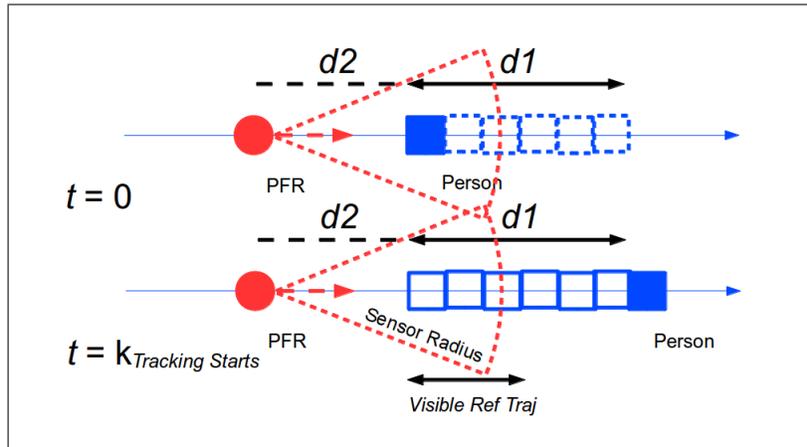


Figure 4.3: Setting Starting Distance & Prediction Horizon

At any given time $t = k + 1$, the PFR moves by $d3$ while the person moves by $d4$, one step forward into time. At all times, the Rf is the visible reference trajectory, recorded by the robot. In Fig. 4.4 the blue filled square is the current position of the person while the unfilled squares towards its left are the past states.

The prediction horizon of the MPC or N in Eq.4.1 holds the relation $N \leq Rf$. The longer this horizon, the lesser the tracking error, but slower is the MPC. The tradeoff depends on several factors such as physical limitations, computing resources, application-specific constraints, and so on. We find that covering 60-70% ($N = 0.6Rf$) gives desirable results for our simulation parameters and hardware. A larger N consumes more time and memory. The rule of thumb is to have a horizon of N such that $Ndt = T$, where T is closed-loop response time, and dt is sampling time.

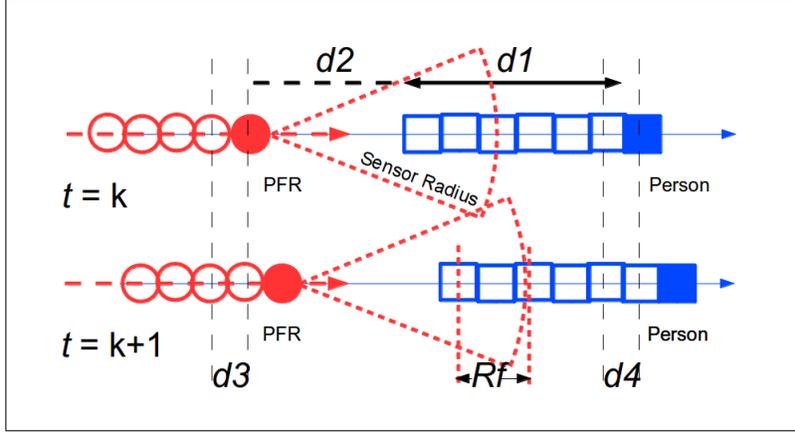


Figure 4.4: State-to-State Transition of PFR & Reference Person

Human Motion Prediction: The reference trajectory for the PFR is generated by a human agent, which uses a constant velocity model for now. But, in Chapter 5, we replace it with real world data to introduce the fuzzyness and uncertainties in real world movement patterns. As of the experiments in this Chapter, the above Fig. 4.4 state-to-state transition at constant velocity is applied.

4.4 OPTIMIZATION

The following optimization problem pertains to the 4-part motion planning strategy. The $U(X, t)_i = [v_x(t_i), v_y(t_i)]$ denote the optimal velocity commands from the MPC. They have constant acceleration throughout $[t_i, t_{i+1}]$, such that $i = 0..k, k + 1, ..T$. This gives us the trajectory of the PFR. The optimization is solved at each time step of the chosen prediction horizon $j = 1..N$ for the reference trajectory $X^{person}(t)_j$ given by the person.

$$\begin{aligned} & \underset{v_x, v_y}{\text{minimize}} && \sum_{j=1}^N (v_x(j) - v_x^{pref}(j))^2 + (v_y(j) - v_y^{pref}(j))^2 \\ & \text{subject to} && \text{Constraints: } A, B, C, D, \text{ and } E \end{aligned} \quad (4.1)$$

where, v_x^{pref} and v_y^{pref} are the preferred forward velocities along (x,y) allowed by the physical constraints of the bot. e.g. 1.2m/s for a P3DX, although, physically it is possible to stress it to the maximum linear speed limit (1.6m/s). The cost function conforms to minimize the forward velocities penalizing deviations from a fixed set of values, standard protocol in a PFR scenario. Note that ω or rate of change in angle is not in this because it is constant and slowly reaches the desired θ at every time step dt , and handled in Eq.4.2. This works because of the non-holonomicity here.

4.4.1 Goal Reaching Constraints

While Eq.4.2 performs go-to-angle behaviour, moving towards the person (or tracking) is accomplished by the following non-linear constraints. Let $X_{t+dt}^{bot_i} = [x_{t+dt}^i]_1^N$ and $Y_{t+dt}^{bot_i} = [y_{t+dt}^i]_1^N$ be the PFR's pose in next iteration, while $X_t^{g_i} = [x_t^{g_i}]_1^N$ and $Y_t^{g_i} = [y_t^{g_i}]_1^N$ be the corresponding goal. This gives us,

$$\Lambda = (X_{t+dt}^{bot_i} - X_t^{g_i})^2 + (Y_{t+dt}^{bot_i} - Y_t^{g_i})^2, \forall i \in [1, N] \quad (4.2)$$

such that $\Lambda \leq (r_{clear})^2$ with N 2D goal/reference positions captured by R_f .

4.4.2 Early Relocation Constraints

These allow the PFR to momentarily abandon tracking the person frame-by-frame, and relocate to a given desired location ($er_i \in E, \forall i$ s.t. $er_i = (x_i^{er}, y_i^{er})$) based on the local map. Ideally, it should also have θ_i^{er} , but we omit for the sake of simplicity. When a map is available, if desired, the ER-points may not be estimated locally from r^{sensor} , and passed manually in the static global map. We use the same principle because it is fast. For a single ER point, $x_{low}^{er} \leq x^{bot}(t^{er}) \leq x_{upr}^{er}$, and $y_{low}^{er} \leq y^{bot}(t^{er}) \leq y_{upr}^{er}$ where, $t = t^{er}, 1 \leq t^{er} \leq N$ is the time in future prediction horizon where the PFR must obey the upper and lower bounds.

As the PFR reaches a location in the map and an ER point falls within its sensor radius, the associated additional constraints are added to the current optimization for the set time interval (iterations) and removed once it passes out of the area. This makes it the first invoke of the Reactive Layer (Fig.1.3) that gives dynamic input to the MPC. This additional constraint forces the PFR to stay close to the ER point during the given interval, assigning a higher priority than the current reference trajectory.

4.5 Intersections & Early Relocation

Obstacles like walls (Figure 4.6), fail the sensor to record the R_f well to maintain the FOV, resulting in loss of line of sight. Notice that the red tracks of the PFR (Figure 4.7) are aligned with the black cross instead of the blue reference track. So, instead of tracking the person in every frame, it (MPC controller) moves to a new goal, waits, and returns to normal tracking once the person arrives.

In Figure 4.5 we show two successive instants in time as the PFR tracks a Person about a right turn without any intersection or divisor such as a wall. We can see how the FOV manages to record the visible reference trajectory inside the sensor radius.

4.5.1 Choice of Early-Relocation Way-Points

Consider that the PFR must leap ahead to K early-relocation points in the set E denoted by $[x_i^{er}, y_i^{er}]$ at $t = t_i^{er} \forall i = 1 \dots K$, (and) $K \leq N$ that are associated with K such time instants along the trajectory.

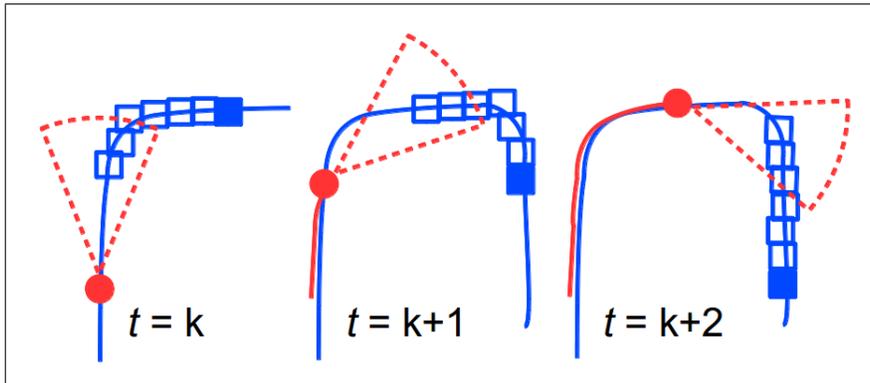


Figure 4.5: Setting Starting Distance & Prediction Horizon

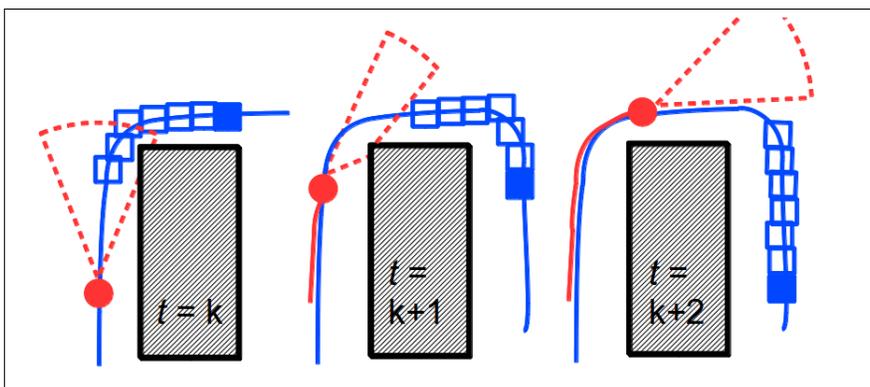


Figure 4.6: Normal Maneuver Around L-Junction Using Rf Without ER

Then, the below set of constraints $x_{low_i} \leq x_i \leq x_{upr_i}, y_{low_i} \leq y_i \leq y_{upr_i}, \forall i$ are added to Eq. 4.1 so that the solver tries to fit as close as possible with some tolerance on the upper and lower limits that the pose can take at $[x^{bot}(t_i^{er}), y^{bot}(t_i^{er})]$. With a long enough prediction horizon and Rf , it can achieve a smoother path.

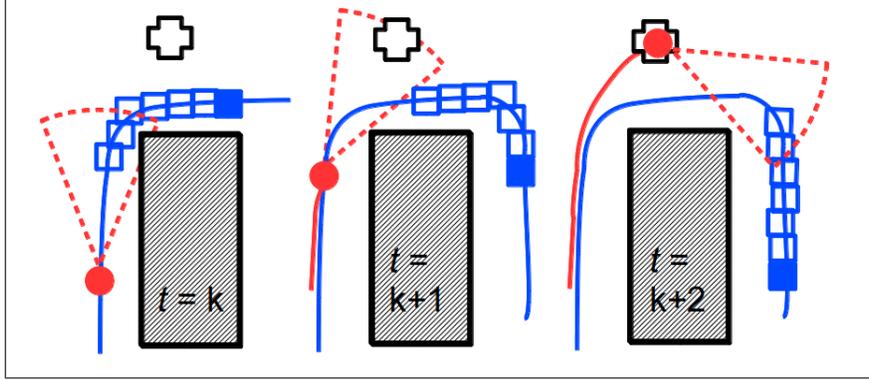


Figure 4.7: Maneuver Around L-Junction Using Early Relocation & Rf

The chosen $N = 0.7Rf$, a simulation parameter, helps deal with unexpected changes such as incoming dynamic obstacles (like pedestrians). The normal one-step MPC is too agile (over 100fps) to deal with uncertainties while with a longer horizon (dl in Fig. ??), it needs to wait longer leading to slow response and loss of FOV. To solve this we employ ER and, at present, pass location manually to the simulator, but it can also be automated. Fig. 4.2: A-F gives 6 scenarios and possible ER positions for the same. We manually select a 2D (x_{e1}, y_{e1}) location that is around the center of an intersection at the blind spots, equidistant from all branching routes such that all routes are visible. The $\theta_{current} - \theta_{desired}$ gives the heading guidance.

4.6 RESULTS

The proposed formulation is implemented using the *fmincon* solver in Matlab R2020b with the interior-point method as they are best suited for small dense problems and solve in polynomial-time. We first show two scenarios with varying complexity and then tabulate the runtime analysis of the MPC, in simulation mode, based on a Corei5, 1.6Ghz, single-core machine.

4.6.1 Case 1: Simple T & L Junction

The trajectory of the person (Figure 4.8) starts from lower-left section to the upper-right section following a long corridor. In this case, there is a single early-relocation region. As the PFR moves past (4, 28), the strategy triggers, and the MPC switches from normal behaviour to MPC1, aiming for the target region. When, the Person reaches and makes the turn, only then the normal operation resumes. The PFR waits for the Person to turn till that time instant. It is assumed that this information is available

based on local map and t_e when the event triggers. While the Normal process loses the person at around (10, 40), the proposed MPC is able to traverse till the goal point.

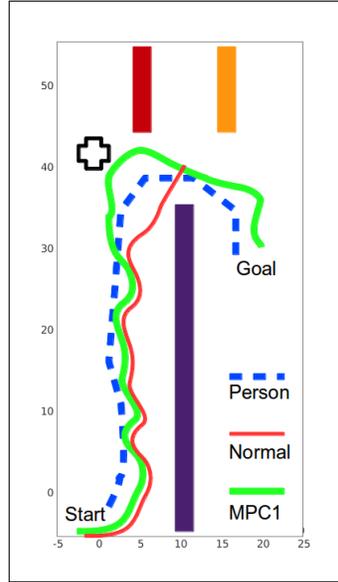


Figure 4.8: Simple T,L-Junction With/Without Early-Relocation

In another case, we see how a larger turning radius C Figure 4.9, means the Normal & MPC1, both pretty much complete the task, but with an obstacle course it suffers from loss of sight and breakdown (Figure 10) next. And, the resulting trajectory with the ER constraints in MPC1 is considerably different when the dynamic obstacle appears. Such large radius and space might not be available always, and this is why ER is necessary. These are cases where Normal fails, and we present some more scenarios below.

4.6.2 Case 1: Simple T & L Junction + Dynamic Obstacle

When a dynamic obstacle appears at an intersection, it makes the situation even more complex. Because now the PFR can be blocked from following the Person resulting in a delay, and eventual loss of sight when the person disappears into the corridor after the right turn at (15, 28). This therefore has 2 early-relocation regions for ease. While the first one at around (3, 24) helps with the obstacle course collision, the 2nd at around (22, 32) helps with the 2nd turn into the corridor. Again, the Normal process loses the person at around (7, 22) due to both the T-junction and the obstacle, the proposed MPC is able to traverse till the goal point. Look at Appendix for a step-by-step transition result.

4.6.3 Case 3: 3-Way T-Junction At Crossroads

Occasionally in indoor environments PFRs may encounter typical cases (Figure 4.10) where a single early-relocation region suffices the smooth transition. It shows how the one-step MPC keeps following

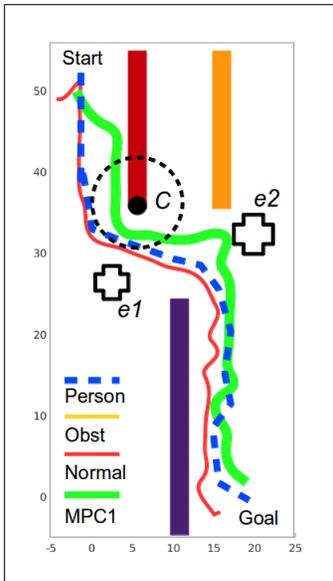


Figure 4.9: Large Turning Radius at T,L-Junction With/Without Early-Relocation

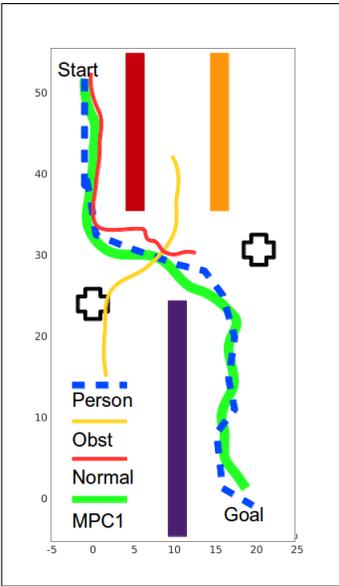


Figure 4.10: Dynamic Obstacle Course in T,L-Junction With/Without Early-Relocation

the forward lane eventually losing the track, while the MPC1 with early-relocation notices the bend, U-turn the person takes, and is able to follow the same gracefully.

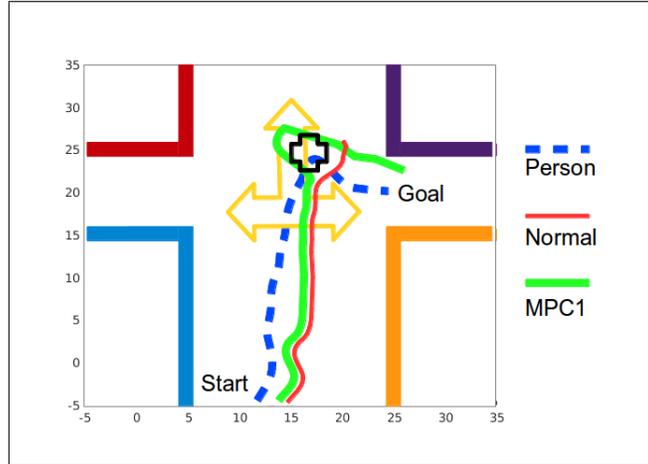


Figure 4.11: Simple T-Junction With/Without Early-Relocation

4.6.4 Case 1: FOV Analysis in Crowd & Dynamic Obstacles

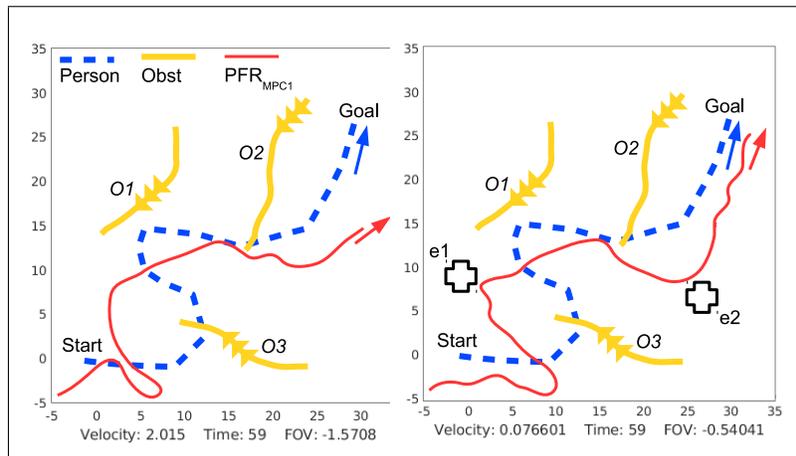


Figure 4.12: 3 Dynamic Obstacles Left: Without ER, Right: With ER

Even without any junction or barrier, dynamic obstacles like a pedestrian can significantly block the PFR causing delay, disorientation, and losing sight of the target person. The ER strategy allows avoiding this issue in many cases. For example, while the PFR (Figure 4.12) can avoid O1 and O3, and resume tracking, it soon gets disoriented without receiving further tracks or goal nodes. The last recorded node and the FOV suffers from a big delay due to the two dynamic obstacle course. The PFR is moving at high velocity over 2 units/iteration and makes a relative angle (red angle) more than 57.50^{deg} with the bearing of the person (blue arrow). The same is avoided in Figure 4.12 where the PFR reaches the

destination almost at rest at 0.01 units/iteration. The FOV is -0.57 translating to 28.07^{deg} , well within the β tolerance of the FOV constraint.

4.6.5 Runtime Analysis

Several 2D goal positions denoted as nodes or vertices comprise a typical state-space trajectory $X(t)$. Each requires multiple outer-iterations (call of the objective function or CoBF) to track, and each computed control input $[v_x, v_y, \theta]$ is the result of multiple inner iterations (of solver) till convergence of the online optimization. The following tables show the runtime obtained from 5 simulations each of different scenarios, with 60 nodes each for a specific user-defined reference trajectory and fixed 2D workspace. But, it varies in the number of obstacles, including dynamic paths, the type of junctions, and their overall distribution in the map.

No. Static Obstacles	Dynamic Obs Length	Avg of 5 Minimum (s)	Avg of 5 Maximum (s)	Avg of 5 Median (s)	No. Constraints Violations
0	0	0.0066	1.2638	0.0097	3
4	1	0.0087	1.2150	0.0112	5
8	4	0.0092	1.3603	0.0129	6
16	7	0.0163	1.3775	0.0166	11
32	11	0.0262	1.4597	0.0311	17

Figure 4.13: Runtime vs Number of Obstacles for Fixed Trajectory

Fig 4.13 shows per-MPC plan runtime for varying number of obstacles, length of dynamic obstacle trajectory, and several constraint violations. A large number of obstacles increase the maximum per-plan runtime, because of higher density around the trajectory and reduced free space. While 32 obstacles result in the violation of 17 constraints, the MPC still converges to local minima. There is a 3.96x increase in minimum average runtime from 0.0066s, for baseline without any obstacle, to 0.0262s, with 32 obstacles. The same on the higher side results in a 1.15x increase. MPC re-planning for early-relocation points add to the runtime but benefit from better FOV tracking and reduced instances of infeasible convergence of the MPC. Among all simulations, the upper margin of 20 Hz is the maximum runtime in any scenario costing up to 0.05 secs/plan. Field studies on pedestrian walking speed have found the average speed to be around 1.25m/s, considering which the interior-point-based solver is almost real-time for our application, by solving up to 20 nodes per second.

Fig. 4.14 shows the per-plan runtime of a particular trajectory type, the number of L, T, and X-junctions along the way, the number of times the MPC optimization results in an infeasible convergence, and the number of nodes lost during tracking without ER. The total number of MPC plan/computes along each trajectory is - outer iterations x iterations per MPC-plan. For example, having obstacles along with intersections results in a higher per-plan runtime of 1.1890s for 5 junctions, while resulting in 4 cases of infeasible convergence. The same trajectory parsed without ER misses 35 out of 60 nodes, which

Trajectory Types ER Strategy	No. L	No. T	No. X	Avg of 5 Minimum Runtime (s)	Avg of 5 Maximum Runtime (s)	Infeasible CoBF Converg	FOV Loss w/o ER (node/60)
Turn Around Long Corridor	2	1	0	0.0083	0.0105	2	20
Pass Multiple Junctions	2	2	0	0.0070	0.0132	3	25
Curve Around Cross Road	0	0	1	0.0077	0.0124	2	15
S-Shaped Pass	2	3	0	0.0115	0.0155	3	13
S-Shaped & Passing Obstacle	2	3	0	1.1890	1.2538	4	35

Figure 4.14: Trajectory Types vs per-MPC Plan Runtime, Effect of ER on Loss of Target Tracking

doesn't happen otherwise. This shows the success of ER. The runtime is also affected by the number and type of junction because it increases or decreases the number of ER shifts performed by the PFR.

Table 4.1: Example of ER on Trajectory Completion

Environment Type	Path Length	Area	Final FOV
Simple T-Junction	60 nodes	750 sq units	Yes
Three Way Junction	60 nodes	1100 sq units	Yes

4.7 CONCLUSIONS

The proposed controller allows the PFR to continue tracking at almost all times. It fulfills the primary objective and contribution of this work. In comparison with the normal mode, without an early-relocation block, the presence of intersections and dynamic obstacles results in a breakdown sooner or later. Though a basic per-node MPC is good enough for close-by tracking, it does not yield relief from impending occlusions in near future. Thus, the MPC-based prioritized tracking momentarily ignores tracking the person in every frame but prioritizes another location in its vicinity, then relocates to the new location early in time, offers added benefits. It shows a natural person following behaviour. Any differential-drive system can be converted into a socially adept person-following robot with this controller. The future direction of research would be to implement this on a real physical robot like P3DX.

Chapter 5

Out-of-Sight Prediction

The ability to predict the movements of the target person allows a person following robot (PFR) to coexist with the person while still complying with the social norms. In human-robot collaboration, this is an essential requisite for long-term time-dependent navigation and not losing sight of the person during momentary occlusions that may arise from a crowd due to static or dynamic obstacles, other human beings, or intersections in the local surrounding. The PFR must not only traverse to the previously unknown goal position but also relocate the target person after the miss, and resume following. In this paper, we try to solve this as a coupled motion-planning and control problem by formulating a model predictive control (MPC) controller with non-linear constraints for a wheeled differential-drive robot. And, using a human motion prediction strategy based on the recorded pose and trajectory information of both the moving target person and the PFR, add additional constraints to the same MPC, to recompute the optimal controls to the wheels. We make comparisons with RNNs like LSTM and Early Relocation for learning the best-predicted reference path.

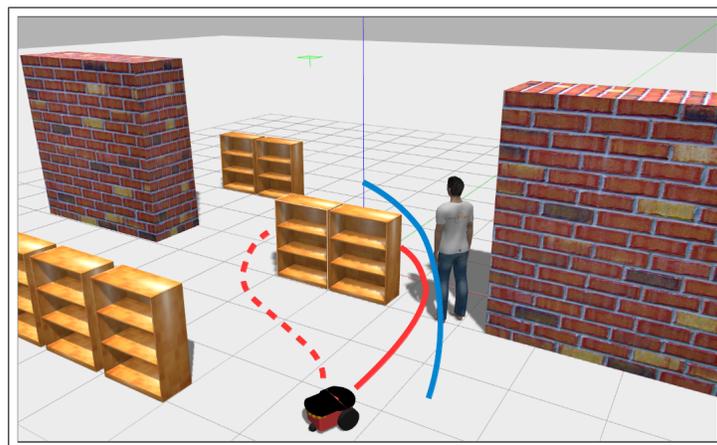


Figure 5.1: A target person traces a reference path in a structured environment (blue) between the narrow wall and bookshelf. The P3DX bot can follow the person (solid red) normally or it can use long-term prediction to find the unknown goal position behind the bookshelf, at a T-junction and move to it autonomously (dotted red).

MPC is best suited for complex constrained problems because it allows the PFR to periodically update the tracking information, as well as to adapt to the moving person’s stride. We show the results using a simulated indoor environment and lay the foundation for its implementation on a real robot. Our proposed method offers a robust person following behaviour without the explicit need for policy learning or offline computation, allowing us to design a generalized framework.

5.1 Related Work

In this paper, we approach the problem as a coupled motion-planning and generating corresponding optimal control inputs to steer the robot to minimize the tracking error and a list of other special constraints. First, we propose an MPC controller that can follow a specific person abiding by all the criteria of a PFR. Second, based on the robot’s observations the MPC can use predictions from a model such as LSTMs [23,24,25] to learn the possible reference path of the person (as its new goal positions) for some instances into the future. This allows us to still maintain tracking when the person goes out of sight momentarily. We implement this using the kinodynamic constraints of a differential-drive PFR, which is often ignored in most of the works [2,26,27]. This allows us to design a more generalized framework that can be directly applied to real-world commercially available wheeled mobile robots.

When the target person is in view, following it is a simple task, but missing the line of sight, and then trying to relocate the person even in the immediate neighborhood can be extremely difficult. Such scenarios can arise in uncertain dynamic environments such as blind curves, T and L-intersections [28], cross-roads [29], and partial or complete occlusion due to crowd. As we know they are more frequent in structured indoor settings, and hence simulate using a few of such empirical cases. From [30] we find that the average walking speed of younger pedestrians between age 14 and 64 is 1.25 m/s, while for older than 65 it drops to 0.67 m/s. This additional information favors certain constraints allowing us to implement a real-time online control as well as to adapt to the speed of movement of the person. While real-time path tracking using MPC for a differential drive mobile robot [31,32] when the reference path is known apriori is a trivial problem, a person-following scenario introduces significant challenges, both from a social and control perspective. MPC is still the controller of choice for complex multi-variate systems.

But, prediction-based MPC can be used to its true potential when the person goes out of sight. While we capture the pose of the target person $[x, y, \theta]$ till the missing time frame when alone, replicating the same for all the dynamic actors in its perceived boundary allows for making predictions into the future. As shown in Fig 4.2 The curve fitting is no more useful than the map information. Foka et al [33] classify prediction into two types, short and long. In our work, we specifically aim to model long-term prediction because this allows us to re-establish the navigation goal points. This allows us to use the map information and develop Early Relocation (ER) strategy where we simply relocate the robot by abandoning the person tracking to increase the chances of not missing it at intersections and occlusions. Or, to employ RNNs like Social LSTM [23] to predict trajectories few seconds into the future. Now,

considering the average speed of 1.25 m/s, this translates to several meters of local motion planning. In terms of MPC, this means a longer prediction horizon based on the predicted reference path instead of the true reference path that the person would take in the near future. Because of this MPC can generate control inputs at each time step without causing discomfort and natural following behavior.

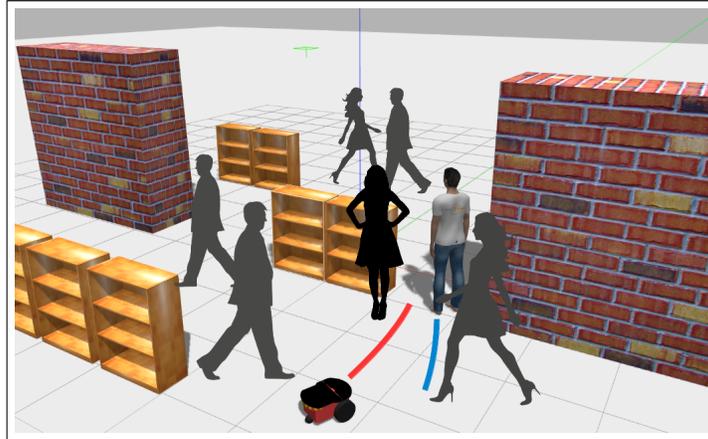


Figure 5.2: With dynamic humans in the local map, the above trajectories get shorter. Based on the visible reference path of the target person (blue) and current robot pose (red) it predicts a path leading to human discomfort and collision. Tracking breaks as the person moves beyond the shelf, requiring LSTMs for path prediction after missing the person from FOV.

5.2 Problem Formulation

Consider the 2D workspace (discussed in Chapters 2 and 3) with human beings too, resulting in crowd and occlusions. The dynamic obstacles are unknown until they are detected by the robot's sensors, denoted by r^{bot} . The number of actors and their trajectory is user-defined for every simulation. It can be considered as a set of 2D positions that belong to the above broader set. The $Obsi = [x_i, y_i]$ set is updated at every iteration. The predicted path from the LSTM model is the updated $X^g(t)$. The path is composed of nodes or vertices, and the length of the path is denoted by the count. A typical motivating scene is presented in Fig 5.2.

5.3 Long-Term Prediction Model

Consider a trivial case where the local map is provided with certain known locations where the intersections or narrow junctions appear. To continue tracking and not cause discomfort to the person, the PFR must relocate to an unknown goal location. There are 2 ways to do so.

1. Using Early Relocation: When the known map locations are appended to the $X^g(t)$ and additional bounds are added for specific time slots.

- Using RNNs like Social LSTM: Based on different input vectors consisting of pose only, pose and velocity, and acceleration, etc. to predict the complete trajectory, rather than just a goal location. Then, update the $X^g(t)$

In 1, that is, ER, given a set $E = [x_i^{er}, y_i^{er}]$ at $t = t_i^{er}, \forall i = 1 \dots K$, and $K \leq N$ for K such time instants or most probable early-relocation points along the trajectory. By adding the below set of constraints to the MPC, $x_{low_i} \leq x_i \leq x_{upr_i}, y_{low_i} \leq y_i \leq y_{upr_i}, \forall i$ with reasonable upper and lower limits on the 2D positions feasible for the bot. We can see that increasing the N (Prediction Horizon) leads to smooth transitions as well. But, for 2, we use the Alahi et. al. [7] Social LSTM as the base model, trained on Zara 1 and 2, and Hotel sequences. Because LSTM gives possible future trajectories adopted by a human in a social scene, a PFR can use it as a reference path for tracking ahead of time. Fig.5.3 gives the complete pipeline.

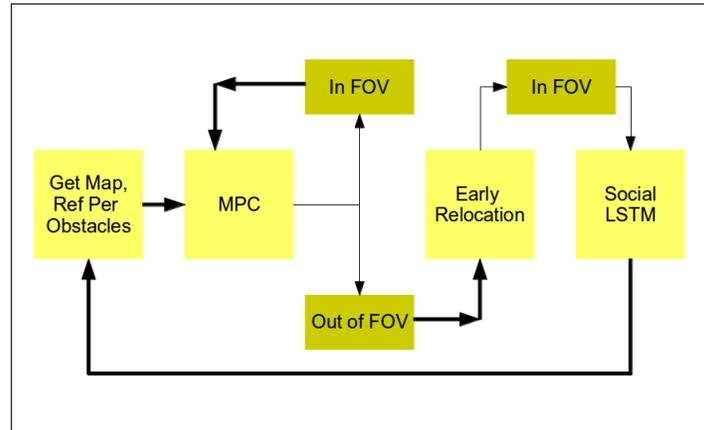


Figure 5.3: Closed-Loop Framework for In-FOV vs Out-of-FOV States of MPC Controller for Long-Term Prediction by LSTM and ER

5.4 Results

We use *fmincon* solver for the optimization in Matlab R2020b. The choice of Interior Point Algorithm makes it fast enough to process up to 100 nodes in a second and can find optimal solutions in polynomial-time. We discuss some empirical cases and then discuss the runtime efficiency of the MPC, in simulation mode, using a 1.6Ghz, single-core i5 CPU.

5.4.1 Case 1: Simple Junctions, No Crowd

Figs. 5.4,5.5 show per-node-indexed trajectories of the target person (in red) and PFR (in blue). The PFR relocates to the available ER point $[8, 25]$, node = 28. In another case with the same scene, the person takes a possible diversion at around the 26th node. Note that both the diversions towards up or left, are in FOV from the above ER point.

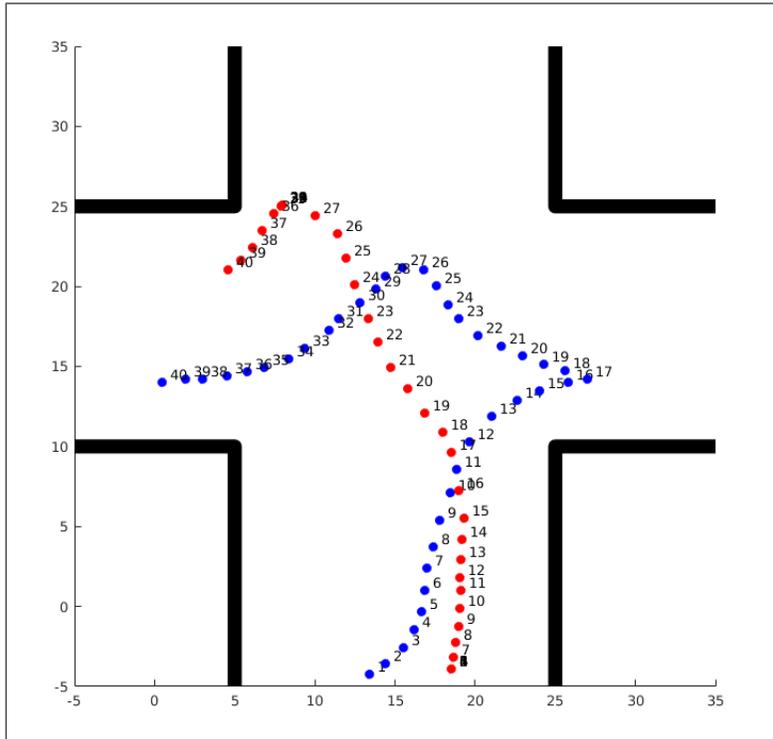


Figure 5.4: Static Obstacle Course in T,L-Junction

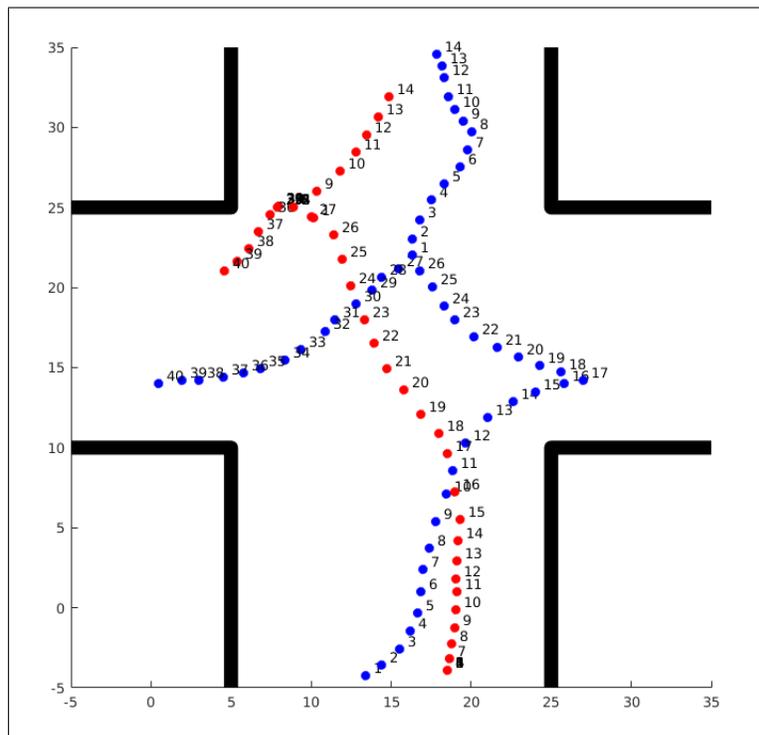


Figure 5.5: Static Obstacle Course in T,L-Junction Diversion at 26th Node, in FOV from ER Point

In the same environment, we show another modified trajectory path that the user might take in anticipation of the available options. We show in Figure 5.6 that the same node 26 can result in another diversion. The point is that without a prediction model, and simply following the target person can result in a lot of confusion and infeasible path planning only to discard a little later. In a trivial case, there are no provisions for discarding, and hence in a crowded setting, this will always lead to freezing and complete breakdown of the path-following behaviour. This is where prediction models can give a better choice of ER locations, that we may not be able to manually assign during fully autonomous operation.

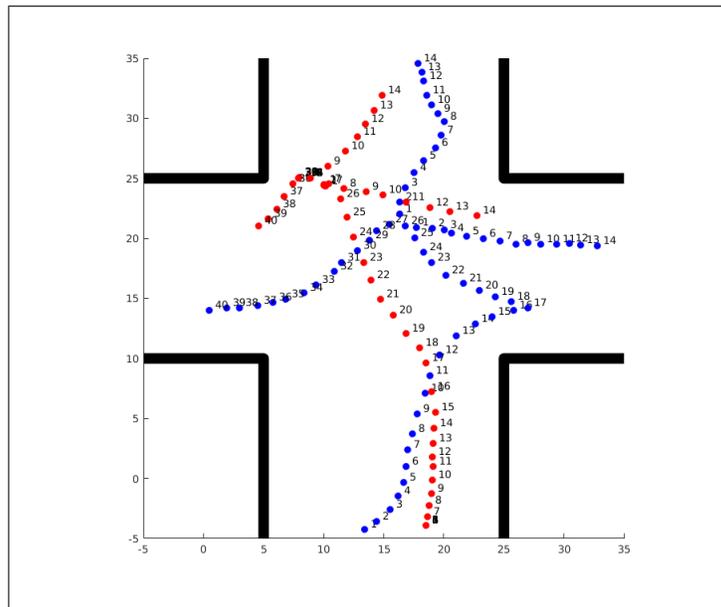


Figure 5.6: Static Obstacle Course in T,L-Junction Another Possible Diversion at 26th Node, in FOV from ER Point

5.4.2 Case 2: Crowd by Moving Humans and LSTM

When there is a crowd due to moving obstacles such as human beings, ER is no longer useful. It fails because it cannot guarantee a collision-free path. Then, the LSTM block Fig.5.3 is used to predict the reference path in a coupled manner for the MPC.

As we can see, the predicted and true trajectories, for ped24 in Fig.5.7 and ped111 in Fig.5.8, from different walking persons in the frames, are used to learn the path into the future. This is then fed into the initial block of the MPC, at the next iteration. In Fig.5.7 the PFR follows the predicted trajectory of ped 24 successfully while complying with social norms.

In Fig.5.8 we emphasize how an even more complex scenario with 4 moving obstacles (ped 111,112,113,109) and one static (ped 107), the chosen Social LSTM can predict reliable trajectory following social norms, to update the goal locations of the MPC.

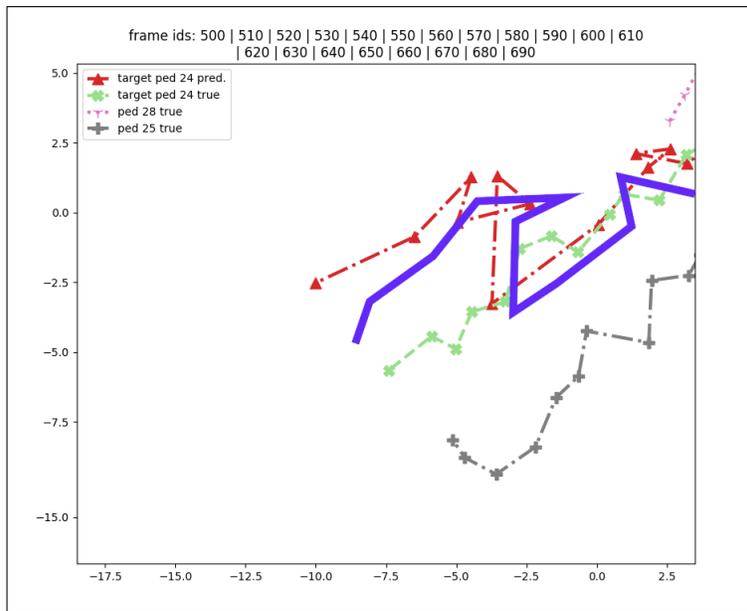


Figure 5.7: Dynamic Obstacle Trajectory With LSTM with 2 Moving People, and 1 Target Person. 1 PFR in Purple Track

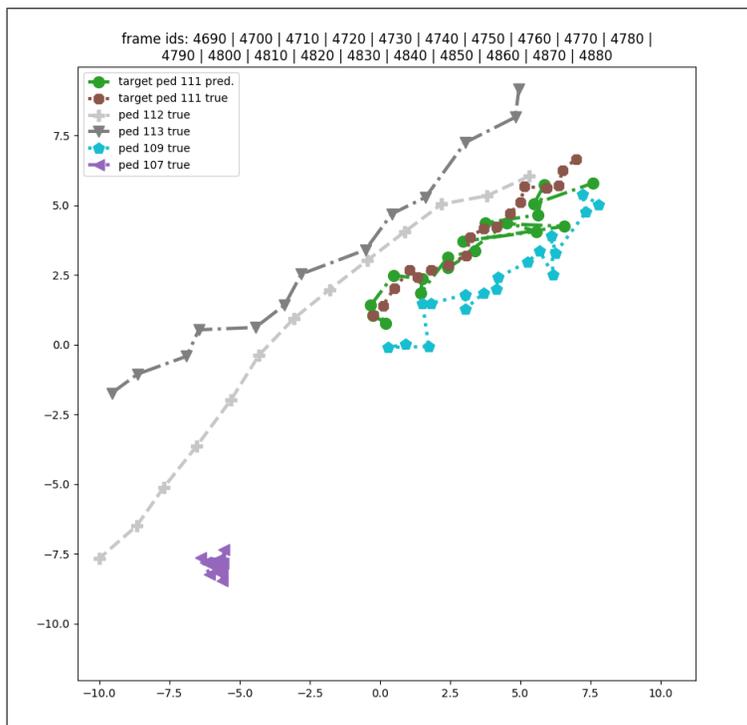


Figure 5.8: Dynamic Obstacle Trajectory With LSTM for 4 Moving People, and 1 Target Person

5.4.3 Runtime Analysis

A typical trajectory consists of several goal points shown as nodes/vertices on a trajectory as shown in Fig 5.7 and 5.8. Each requires multiple outer-iterations to track, and each computed control input $[v_x, v_y, \theta]$ is the result of multiple inner iterations (of solver) till convergence of the online optimization. We present here a tabular view of runtime obtained from running 50 simulations of each scenario, each with a fixed number of nodes (40) for a fixed reference trajectory across a fixed simulated area.

Table 5.1: Ablations: Path Completion vs LSTMs

LSTM	Avg Loss	Avg Late Arr	Avg CV
Vanilla	13	3	23
Social + Pose	9	2.1	18
Social + Pose + Velocity	6	1.3	11
Early Relocation + Crowd	Total	None	None
Early Relocation + Static	3	2.2	6
Social LSTM + 2 People	2	1.27	12
Social LSTM + 4 People	4	2.31	16

Here, CV (Constraint Violations) helps study the optimization and solver’s performance concerning the constraints for human comfort and safety. So, the lesser they are number, the more socially adept the results are.

5.5 CONCLUSIONS

To summarize, the main contributions of the paper are as follows.

1. A socially- acceptable path tracking controller for the nonholonomic differential-drive wheeled robot.
2. Based on long-term path prediction from RNNs like Social LSTM models, we generate paths for out-of-sight time duration.
3. A single holistic NMPC that complies with all the criteria for a PFR’s behavioral dynamics for motion planning and obstacle avoidance.
4. Evaluation of the framework in terms of the breakdown of tracking due to loss of FOV, agility for various human movement patterns, and ablation studies for comparison with various prediction models.

The average runtime achieves a rate of 20Hz offering enough fidelity for pedestrians between age 14 to 60’s walking speeds. The LSTMs offer the use of velocity and acceleration as a time-dependent

smooth reference path by utilizing the kinematics of the PFR as well in the modeling. The future direction of research would be to implement this on a real physical robot, which is currently on the way.

APPENDIX

The Pioneer P3DX (Fig. 5.9) is a 2-wheeled differential-drive robot used for testing our controller, where (v_r, v_l) are left-right wheel velocities, V is same as before, (L) is the distance between the 2 wheels, and (R) is the turning radius of the vehicle.

$$(v_r, v_l) = \left[\frac{2V + \omega L}{2R}, \frac{2V - \omega L}{2R} \right] \quad (5.1)$$



Figure 5.9: Our Configuration of P3DX 2-Wheeled Drive with Mounts

Chapter 6

Conclusions

The primary contributions of the work rest on the predictive controller design that takes into account the social aspect of shared autonomy and moving a robot safely around humans yet following a single person.

1. We adopt natural person-following behaviour into a differential drive system with non-holonomic constraints. We formulate a multi-layered model predictive controller-based framework for a novel application where it goes beyond trivial in-view tracking per frame to autonomous off-view predictive tracking. It shifts from reactive path following to active relocation to vantage points and resumes normal tracking. This is the first work of its kind to the best of our knowledge.
2. We emphasize the single holistic coupled motion-planning and obstacle avoidance routine of the MPC's optimization problem using constraints for satisfying various kinematics, and dynamics of PFR. The optimizations are solved without explicitly linearizing around a fixed point to avoid uncontrollability issues, using non-linear solvers. We also include results on violations of constraints to verify optimization performance.
3. We develop the concept of early relocation or ER which allows us to use local map information for avoiding target miss and maintaining FOV, especially around corners and intersections like T and L-junctions.
4. We show the application of Social LSTM-like recurrent neural networks in predicting pedestrian future trajectories, that are used for out of sight or off-view active target tracking and vantage point relocation.
5. The results include extensive experiments in various scenarios with single and multiple static and dynamic obstacles, single and multiple intersections, 6 different but common human movement patterns at various walking speeds, and real-world pedestrian movement data. We have tested the effects of FOV, single vs n-step MPC, delay, and short vs long-term navigation. It achieves agile run-time person following behaviour in all the cases.

Related Publications

6.1 Related Publications

Ashe, Avijit, and K. Madhava Krishna. "Dynamic Target Tracking & Collision Avoidance Behaviour of Person Following Robot Using Model Predictive Control." 2020 24th International Conference on System Theory, Control and Computing (ICSTCC). IEEE, 2020.

6.2 Other Publications

1. Ashe, Avijit Kumar, and K. Madhava Krishna. "Followman: Control of Social Person Following Robot." 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, 2021.
2. Ashe, Avijit Kumar, and K. Madhava Krishna. "Maneuvering Intersections & Occlusions Using MPC-Based Prioritized Tracking for Differential Drive Person Following Robot." 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE). IEEE, 2021.

6.2.1 Supplementary Publication

Ashe, Avijit Kumar. "Towards Out-of-Sight Predictive Tracking for Long-Term Indoor Navigation of Non-Holonomic Person Following Robot." 2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN). IEEE, 2021.

Bibliography

1. Nishimura, Soh, Keita Itou, Takashi Kikuchi, Hiroshi Takemura, and Hiroshi Mizoguchi. "A study of robotizing daily items for an autonomous carrying system-development of person following shopping cart robot." In 2006 9th International Conference on Control, Automation, Robotics and Vision, pp. 1-6. IEEE, 2006.
2. Piezzo, Chiara, and Kenji Suzuki. "Feasibility study of a socially assistive humanoid robot for guiding elderly individuals during walking." *Future Internet* 9, no. 3 (2017): 30.
3. Nakauchi, Yasushi, and Reid Simmons. "A social robot that stands in line." *Autonomous Robots* 12.3 (2002): 313-324.
4. Berns, Karsten, and Syed Atif Mehdi. "Use of an autonomous mobile robot for elderly care." 2010 *Advanced Technologies for Enhancing Quality of Life*. ieee, 2010.
5. Tsalatsanis, Athanasios, K. Valavanis, and Ali Yalcin. "Vision based target tracking and collision avoidance for mobile robots." *Journal of Intelligent and Robotic Systems* 48.2 (2007): 285-304.
6. Kim, Minkyu, et al. "An architecture for person-following using active target search." arXiv preprint arXiv:1809.08793 (2018).
7. Flacco, Fabrizio, et al. "A depth space approach to human-robot collision avoidance." 2012 *IEEE International Conference on Robotics and Automation*. IEEE, 2012.
8. Shiomi, Masahiro, et al. "Effectiveness of social behaviors for autonomous wheelchair robot to support elderly people in Japan." *PloS one* 10.5 (2015): e0128031.
9. Kuo, Cheng-Hao, and Ram Nevatia. "How does person identity recognition help multi-person tracking?." *CVPR 2011*. IEEE, 2011.
10. Shu, Guang, et al. "Part-based multiple-person tracking with partial occlusion handling." 2012 *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012.
11. Koide, Kenji, Jun Miura, and Emanuele Menegatti. "Monocular person tracking and identification with on-line deep feature selection for person following robots." *Robotics and Autonomous Systems* 124 (2020): 103348.

12. Y. Gao, C. G. Lee, K. T. Chong, "Receding horizon tracking control for wheeled mobile robots with time-delay", *Journal of mechanical science and technology*, vol. 22, pp. 2403-2416, 2008.
13. Fulgenzi, Chiara, Anne Spalanzani, and Christian Laugier. "Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid." *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007.
14. Leman, Zulkarnain Ali, et al. "Model Predictive Controller for Path Tracking and Obstacle Avoidance Manoeuvre on Autonomous Vehicle." *2019 12th Asian Control Conference (ASCC)*. IEEE, 2019.
15. Fulgenzi, Chiara, Anne Spalanzani, and Christian Laugier. "Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid." *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007.
16. Kehler, Johannes, Matthias A. Miller, and Frank Allgwer. "A nonlinear model predictive control framework using reference generic terminal ingredients." *IEEE Transactions on Automatic Control* (2019).
17. Hu, Yingbai, et al. "Nonlinear model predictive control for mobile robot using varying-parameter convergent differential neural network." *Robotics* 8.3 (2019): 64.
18. Van Essen, H. A., and Henk Nijmeijer. "Non-linear model predictive control for constrained mobile robots." *2001 European Control Conference (ECC)*. IEEE, 2001.
19. Lim, Heonyoung, et al. "Nonlinear model predictive controller design with obstacle avoidance for a mobile robot." *2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*. IEEE, 2008.
20. Z. Li, J. Deng, R. Lu, Y. Xu, J. Bai, and C.-Y. Su, "Trajectory-tracking control of mobile robot systems incorporating neural-dynamic optimized model predictive approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, pp. 740-749, 2016.
21. Eslami, Narges, and Roya Amiadifard. "Moving Target Tracking and Obstacle Avoidance for a Mobile Robot Using MPC." *2019 27th Iranian Conference on Electrical Engineering (ICEE)*. IEEE, 2019.
22. Ashe, Avijit, and K. Madhava Krishna. "Dynamic Target Tracking & Collision Avoidance Behaviour of Person Following Robot Using Model Predictive Control." *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, 2020.
23. Alahi, Alexandre, et al. "Social LSTM: Human trajectory prediction in crowded spaces." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

24. Vemula, Anirudh, Katharina Muelling, and Jean Oh. "Social attention: Modeling attention in human crowds." 2018 IEEE international Conference on Robotics and Automation (ICRA). IEEE, 2018.
25. Quan, Ruijie, et al. "Holistic LSTM for pedestrian trajectory prediction." IEEE transactions on image processing 30 (2021): 3229-3239.
26. Chan, Wesley P., Sina Radmard, Zhao Quan Hew, Jon Morris, Elizabeth Croft, Van der Loos, and H. F. Machiel. "Autonomous Person-Specific Following Robot." arXiv preprint arXiv:2010.08017 (2020).
27. Islam, Md Jahidul, Jungseok Hong, and Junaed Sattar. "Person-following by autonomous robots: A categorical overview." The International Journal of Robotics Research 38, no. 14 (2019): 1581-1618.
28. Shu, Keqi, Huilong Yu, Xingxin Chen, Long Chen, Qi Wang, Li Li, and Dongpu Cao. "Autonomous Driving at Intersections: A Critical-Turning-Point Approach for Left Turns." arXiv preprint arXiv:2003.02409 (2020).
29. Alonso, Javier, Vicente Milans, Joshu Prez, Enrique Onieva, Carlos Gonzlez, and Teresa De Pedro. "Autonomous vehicle control systems for safe crossroads." Transportation research part C: emerging technologies 19, no. 6 (2011): 1095-1110.
30. Knoblauch, Richard L., Martin T. Pietrucha, and Marsha Nitzburg. "Field studies of pedestrian walking speed and start-up time." Transportation research record 1538.1 (1996): 27-38.
31. Gu, Dongbing, and Huosheng Hu. "Receding horizon tracking control of wheeled mobile robots." IEEE Transactions on control systems technology 14, no. 4 (2006): 743-749.
32. Kuhne, Felipe, Walter Fetter Lages, and J. Gomes da Silva Jr. "Model predictive control of a mobile robot using linearization." In Proceedings of mechatronics and robotics, pp. 525-530. 2004.
33. Foka, Amalia F., and Panos E. Trahanias. "Predictive autonomous robot navigation." IEEE/RSJ international conference on intelligent robots and systems. Vol. 1. IEEE, 2002.
34. Leigh, Angus, et al. "Person tracking and following with 2d laser scanners." 2015 IEEE international conference on robotics and automation (ICRA). IEEE, 2015.
35. Germa, Thierry, et al. "Vision and RFID-based person tracking in crowds from a mobile robot." 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009.
36. Guerrero-Higueras, ngel Manuel, et al. "Tracking people in a mobile robot from 2d lidar scans using full convolutional neural networks for security in cluttered environments." Frontiers in neurobotics 12 (2019): 85.

37. Wengefeld, Tim, et al. "May i be your personal coach? bringing together person tracking and visual re-identification on a mobile robot." Proceedings of ISR 2016: 47st International Symposium on Robotics. VDE, 2016.
38. Singh, Arun Kumar, and K. Madhava Krishna. "Reactive collision avoidance for multiple robots by non linear time scaling." 52nd IEEE Conference on Decision and Control. IEEE, 2013.
39. Haasch, A., et al. "Bironthe bielefeld robot companion." Dialog 11111111 (2004): 11111111.
40. Sidenbladh, Hedvig, Danica Kragic, and Henrik I. Christensen. "A person following behaviour for a mobile robot." Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C). Vol. 1. IEEE, 1999.
41. Nickel, Kai, and Rainer Stiefelhagen. "Real-time person tracking and pointing gesture recognition for human-robot interaction." International Workshop on Computer Vision in Human-Computer Interaction. Springer, Berlin, Heidelberg, 2004.
42. Bischoff, Rainer. "Design concept and realization of the humanoid service robot hermes." Field and Service Robotics. Springer, London, 1998.
43. Asoh, Hideki, et al. "Socially embedded learning of the office-conversant mobile robot jijo-2." IJCAI (2). 1997.