Chapter 5

Application in Robot Navigation with Collision Avoidance

5.1 Introduction

In this dedicated chapter, we consider the problem of navigation in sphere worlds, which involves designing a vector field over a compact, convex subset of Euclidean space punctured by Euclidean disks. The goal is to guide a Euclidean disk-shaped robot from almost all initial positions (excluding a set of measure zero) to a moving target, ensuring collision-free motion throughout the trajectory. We demonstrate the validity of our proposed predefined-time convergent dynamics to solve TVO problems with inequality constraints in robot navigation using simulation results.

The structure of this chapter is outlined as follows. In section 5.2, we formulate the problem. In section 5.3, we define the safe neighborhood of the robot by encoding collision. In section 5.4, we formulate the robot navigation to a moving target as a TVO problem. Finally, in section 5.5, the effectiveness of the proposed approach to solve the defined problem is validated through simulation results.

5.2 Problem Formulation

In this chapter, we consider a disk-shaped robot centered at $z \in \mathcal{W}$ with radius $r \in \mathbb{R}_{\geq 0}$ working in a closed compact convex environment $\mathcal{W} \subset \mathbb{R}^n$. Let workspace $\mathcal{W}$ is populated with $m \in \mathbb{Z}_{\geq 0}$ disk shaped obstacles centered at $q_i \in \mathcal{W}$ with radius $\varrho_i > 0$ for $i \in$

$\{1, 2, \ldots, m\}$. Hence, the free space $\mathcal{F}$ of the robot is defined as the set of configurations in the workspace in which the robot does not collide with any of the obstacles:

$$\mathcal{F} := \{z \in \mathcal{W} | \bar{\mathcal{B}}(z, r) \subseteq \mathcal{W} \backslash \bigcup_{i=1}^{m} \mathcal{B}(q_i, \varrho_i)\},$$

where, $\mathcal{B}(z, r)$ represents a n dimensional open ball centered at z with radius r, and $\bar{\mathcal{B}}(z, r)$ represents its closure. To ensure the connectivity of the free space $\mathcal{F}$, we assume that our disk-shaped robot is capable of navigating around any obstacle in any direction within the workspace $\mathcal{W}$.

Assumption 5.1 *Obstacles are separated from each other by clearance of at least*

$$\|q_i - q_j\| > \varrho_i + \varrho_j + 2r, \quad \forall i \neq j$$

and from the boundary $\partial\mathcal{W}$ of the workspace $\mathcal{W}$ as

$$\min_{p \in \partial\mathcal{W}} \|p - q_i\| > \varrho_i + 2r, \quad \forall i.$$

5.3 Encoding Collisions

In this section, we outline several key properties of power diagrams and explain how they are utilized to define a safe neighborhood for a robot.

5.3.1 Power Diagrams

The power diagram $P(q, \varrho) = \{P_1, P_2, ..., P_m\}$ of a convex environment $\mathcal{W} \in \mathbb{R}^n$, based on the set of generator disks centered at $q = (q_1, q_2, ..., q_m) \in \mathcal{W}^m$ with a vector of radii $\varrho = (\varrho_1, \varrho_2, ..., \varrho_m)$ is a partition of $\mathcal{W}$ such that every point $p \in \mathcal{W}$, is assigned to the closest generator based on the power distance $\|p - q_i\|^2 - \varrho_i^2$ as

$$P_i := \{p \in \mathcal{W} | \|p - q_i\|^2 - \varrho_i^2 \leq \|p - q_j\|^2 - \varrho_j^2, \ \ \forall j \neq i\},$$

A power diagram defines a convex cell decomposition of a convex environment, and the boundary ∂P_i of P_i is define by the boundary $\partial\mathcal{W}$ of the workspace $\mathcal{W}$ and the separating hyperplane H_{ij} between power cells P_i and P_j for some $j \neq i$ [51]. The separating hyperplane H_{ij} between any pair $i \neq j$ of power cells P_i and P_j is perpendicular to the line joining q_i and q_j and passes through the point $h_{ij} := \eta_{ij} p_i + (1 - \eta_{ij} p_j)$,

$$H_{ij} := \{p \in \mathbb{R}^n | (p - h_{ij})^\top (q_i - q_j) = 0\}$$

where, $\eta_{ij} := \frac{1}{2} - \frac{\varrho_i^2 - \varrho_j^2}{2\|q_i - q_j\|^2}$, and the perpendicular distance of qi to H_{ij} is given by

$$d(q_i, H_{ij}) := \min_{q \in H_{ij}} \|p - q_i\| = (1 - \eta_{ij})\|q_i - q_j\| = \varrho_i + \frac{(\|q_i - q_j\| - \varrho_i)^2 - \varrho_j^2}{2\|q_i - q_j\|}.$$

Objective: Let z_c be the center of mass of the robot and our objective is for $z_c(t)$ to track a moving target $z_d(t) \in \mathcal{F}$ after a certain time $t > 0$ ensuring collision avoidance.

5.3.2 Safe Neighborhood of a Robot

Using the robot and obstacles as generator disks of a power diagram of $\mathcal{W}$, we define the local workspace, $\mathcal{LW}(z_c)$, of the robot as

$$\mathcal{LW}(z_c) := \{p \in \mathcal{W} | \|p - z_c\|^2 - r^2 < \|p - q_i\|^2 - \varrho_i^2, \quad \forall i\}$$

To determine a collision-free neighborhood of the robot, we define the robot's local

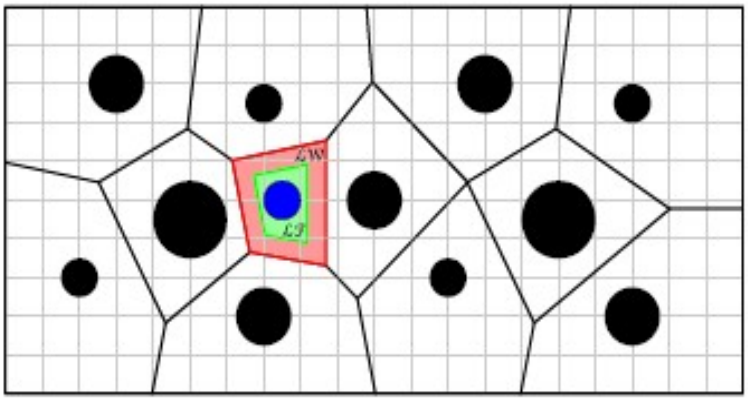


Figure 5.1: Local workspace (red) and local free space (green) of a robot (blue) [4].

free space, $\mathcal{LF}(z_c)$, by eroding $\mathcal{LW}(z_c)$, removing the volume swept along its boundary, $\partial\mathcal{LW}(z_c)$, by the robot body radius, illustrated in Figure 5.1.

Next, collision-free space around z_c is defined as [4]: $\mathcal{LF}(z_c) := \{p \in \mathcal{W} | c_i(z_c)^\top z - d_i(z_c) \leq 0, i = 1, ..., m\}$, where $c_i(z_c) = q_i - z_c$, $d_i(z_c) = (q_i - z_c)^\top \left(\eta_i q_i + (1 - \eta_i) z_c + r\frac{z_c - q_i}{\|z_c - q_i\|}\right)$, where, $\eta_i = \frac{1}{2} - \frac{\varrho_i^2 - r^2}{\|q_i - z_c\|^2}$.

Arslan and Koditschek (2016) [4], proposed the idea of a projected goal which constitutes a continuous computation of projection of $z_d(t)$ onto a safe neighborhood around z_c. Assuming the robot follows single integrator dynamics $\dot{z}_c = u(z_c)$, the controller proposed by Arslan and Koditschek (2016) [4] is $\dot{z}_c = -k(z_c - \hat{z})$, where $k > 0$ is gain and $\hat{z}$ is the orthogonal projection of z_d onto $\mathcal{LF}(z_c)$. The proposed controller works with the assumption that $\|z_i - z_j\| > r_i + r_j + 2r$.

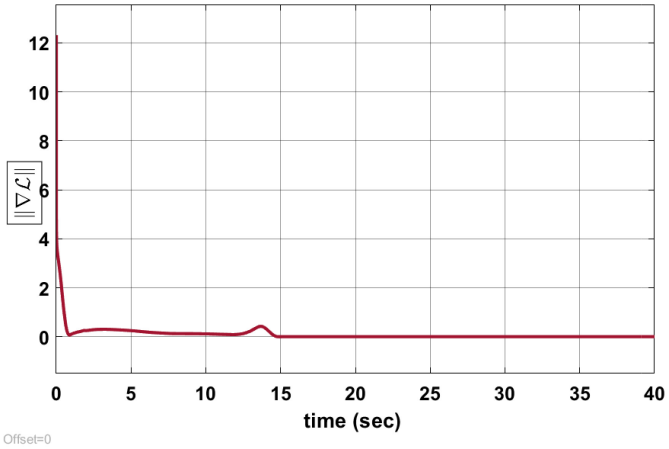


Figure 5.2: Evolution of $||\nabla\mathcal{L}||$ with PUBST based convergence within $t_f = 15$ sec using proposed continuous-time dynamics (4.31). It is shown that $\|\mathcal{L}\| \to 0$ as $t \to t_f$

5.4 Formulation of Robot Navigation as a Time-Varying Optimization Problem

In this section, robot navigation problem discussed in previous section is reformulated as an optimization problem as follows:

$$\hat{z} := \underset{z\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|z - z_d(t)\|^2,$$
$$\text{subject to} \quad c_i(z_c)^\top z - d_i(z_c) \leq 0, \quad i = 1, ..., m$$

Lagrangian for the above problem is written as (discussed in previous chapter)

$$\tilde{L}(t, z, z_c) = \frac{1}{2}\|z - z_d(t)\|^2 - \frac{1}{\sigma(t)}\sum_{i=1}^{m} \log(d_i(z_c) - c_i(z_c)^\top z),$$

We solve the above problem using the proposed predefined-time convergent dynamics

$$\dot{z}(t) = \begin{cases} -(\nabla^2\tilde{\mathcal{L}})^{-1}\left(\vartheta\psi(t, z, \sigma, \delta) + \frac{\partial\nabla\tilde{\mathcal{L}}}{\partial\sigma}\dot{\sigma} + \frac{\partial\nabla\tilde{\mathcal{L}}}{\partial\delta}\dot{\delta} + \frac{\partial\nabla\tilde{\mathcal{L}}}{\partial t}\right) & \text{for } \ t_0 \leq t < t_f, \\ -(\nabla^2\tilde{\mathcal{L}})^{-1}\left(\vartheta\nabla\tilde{\mathcal{L}} + \frac{\partial\nabla\tilde{\mathcal{L}}}{\partial\sigma}\dot{\sigma} + \frac{\partial\nabla\tilde{\mathcal{L}}}{\partial\delta}\dot{\delta} + \frac{\partial\nabla\tilde{\mathcal{L}}}{\partial t}\right), & \text{for } \ t \geq t_f, \end{cases} \tag{5.1}$$

where $\vartheta \in \mathbb{R}$ and satisfies $\vartheta > 1$, $\psi(t, z, \sigma, \delta) = (\psi_1, \psi_2, ..., \psi_n) \in \mathbb{R}^{n\times 1}$, $\psi_i = \frac{(e^{\nabla\tilde{\mathcal{L}}_{z_i}(t,z,\sigma,\delta)} - 1)}{e^{\nabla\tilde{\mathcal{L}}_{z_i}(t,z,\sigma,\delta)}(t_f - t)}$, for $1 \leq i \leq n$.

We propose a controller based on the Lemma 2.9 to navigate $z_c(t)$ to $\hat{z}(t)$ in a priori

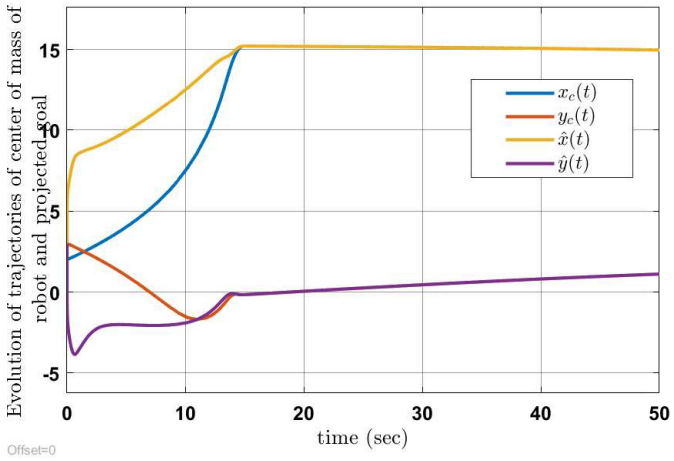


Figure 5.3: Evolution of xy position of center of mass of robot $(x_c(t), y_c(t))$ and projected goal $(\hat{x}, \hat{y})$ evaluated solving TVO problem. It is shown that $z_c(t)$ starts tracking $\hat{z}(t)$ within $t_f = 15sec$.

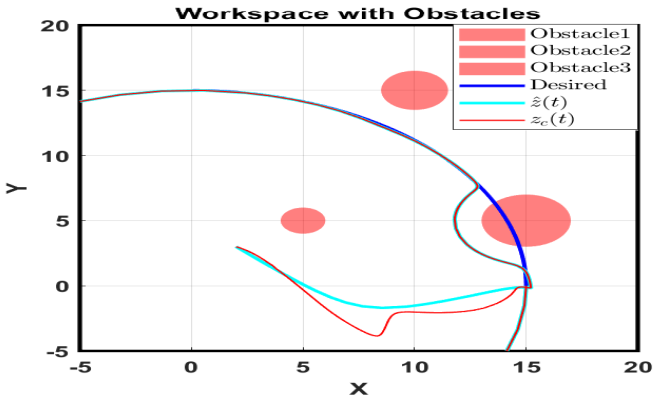


Figure 5.4: Plot of workspace with three obstacles showing the evolution of desired trajectory $z_d(t)$, estimated projected goal $\hat{z}(t)$, and center of mass of robot $z_c(t)$.

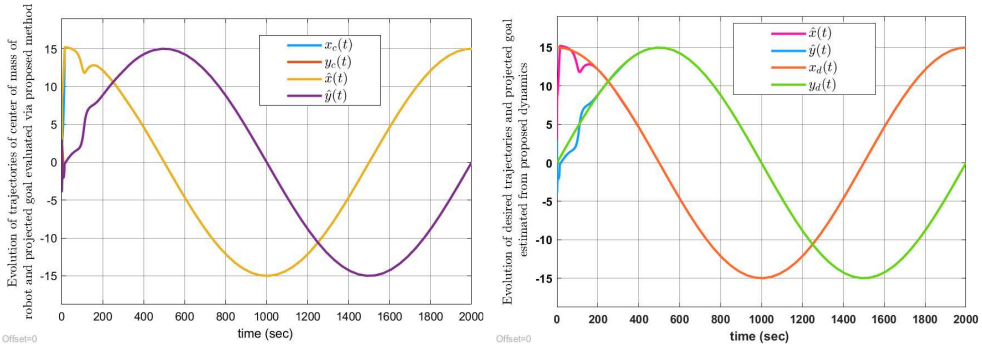
chosen time. Controller dynamics is:

$$u = \begin{cases} -k\frac{e^{(z_c(t)-\hat{z}(t))}-1}{e^{(z_c(t)-\hat{z}(t))}(t_f-t)}, & \text{for } \ t_0 \leq t < t_f, \\ -k(z_c(t)-\hat{z}(t)), & \text{for } \ t \geq t_f, \end{cases} \tag{5.2}$$

where $k > 1$ is gain.

5.5 Simulation Results

In our simulation, we consider a moving target $z_d(t) = [x_d(t), y_d(t)]^\top \in \mathbb{R}^2$, following a circular trajectory with circumference of radius $(r_d = 15)$, centered at origin, and moving



(a) Evolution of xy position of center of mass of robot $(x_c(t), y_c(t))$ and projected goal $(\hat{x}, \hat{y})$ evaluated solving TVO problem for complete time $T = 2000sec$.

(b) Evolution of xy position of the desired trajectory $(x_d(t), y_d(t))$ and projected goal $(\hat{x}, \hat{y})$ evaluated solving TVO problem for complete time $T = 2000sec$.

Figure 5.5: Results using PTC-TVO dynamics (4.31) in solving TVO problem considered for computing projected goal required for safe robot navigation and plot of the desired trajectory which is moving periodically with $T = 2000sec$.

periodically with $T = 2000$ second. We consider $m = 3$ obstacles in workspace $\mathcal{W}$, with centres $z_1 = [5, 5], z_2 = [15, 5], z_3 = [10, 15]$ with radii $r_1 = 1, r_2 = 2, r_3 = 1.5$ respectively. We consider a disk-shaped robot with a radius $r = 2$ with the initial point of its center of mass $z_c(0) = [2, 3]$. Next, we solve TVO problem to estimate the projected goal $(\hat{z}(t)) = [\hat{x}(t), \hat{y}(t)]$ using proposed continuous-time dynamics (5.1), where, $\tilde{\mathcal{L}}(t, z, z_c) = \frac{1}{2}\|z - z_d(t)\|^2 - \frac{1}{\sigma(t)} \sum_{i=1}^{m} \log(d_i(z_c) - c_i(z_c)^\top z)$, here we took $\sigma(t) = e^{0.01t}, \vartheta = 50, t_f = 15sec$.

It is shown in Figure 5.2 that $\|\nabla \mathcal{L}(t, z, z_c)\| \to 0$ as $t \to t_f$, which means we reach estimated optimal trajectory $\hat{z}(t)$ within $t_f = 15sec$.

Optimal trajectory $\hat{z}(t)$ provides safe navigation, avoiding obstacles in following the desired trajectory which is allowed to intersect obstacles as shown in Figure 5.4. To track this moving projected target $\hat{z}(t)$ by the robot in a priori chosen time (t_f), we use controller (5.2), where for simulation, we chose gain $k = 5$. It is shown in Figure 5.3 that the trajectory of the center of mass of robot $(z_c(t))$ starts tracking $\hat{z}(t)$ within a predefined-time $t_f = 15sec$.

Figure 5.5 shows evolution of trajectories $z_c(t), \hat{z}(t)$, and $z_d(t)$ over time. Figure 5.6 shows a phase portrait representing navigation of the robot. As we can see in Figure 5.4,

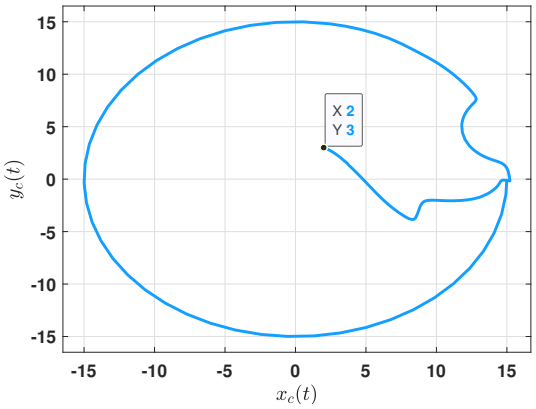


Figure 5.6: Phase-portrait: $x_c(t)$ opposite $y_c(t)$ showing that center of mass of the robot follows a circular trajectory as desired over time.

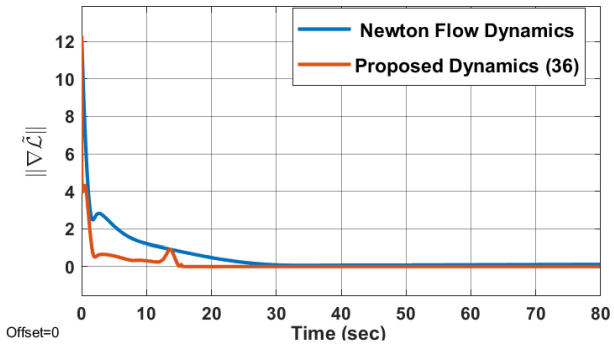


Figure 5.7: Comparison of evolution of $\|\nabla\tilde{\mathcal{L}}\|$ with predefined-time convergence within $t_f = 15$ sec using proposed dynamics (4.31) and Newton flow dynamics proposed in Fazlyab et al.(2017) [3]

.

the robot successfully tracks the moving target while avoiding the circular obstacles.

In Figure 5.7, we demonstrate that $\|\nabla\tilde{\mathcal{L}}\| \to 0$ within a predefined upper bound of settling time $t_f = 15$ seconds whereas dynamics proposed in Fazylab et al.(2017) [3] for solving TVO problems with inequality constraints takes more than $t = 40$ seconds to reach the estimated optimal trajectory. Additionally, we show the results obtained by plotting $\|\nabla\tilde{\mathcal{L}}\|$ on a semilog scale. Figure 5.8, illustrates that the steady-state error ($\|\nabla\tilde{\mathcal{L}}\| < 10^{-3}$ after $t_f > 15$ seconds) in the case of proposed dynamics is comparatively much smaller than the Newton flow dynamics. Figure 5.9 shows the plot of control inputs using dynamics (5.2) and the controller proposed by Arslan and Koditschek (2016) [4]: $\dot{z}_c = -k(z_c - \hat{z})$, where $k > 0$ is the gain.

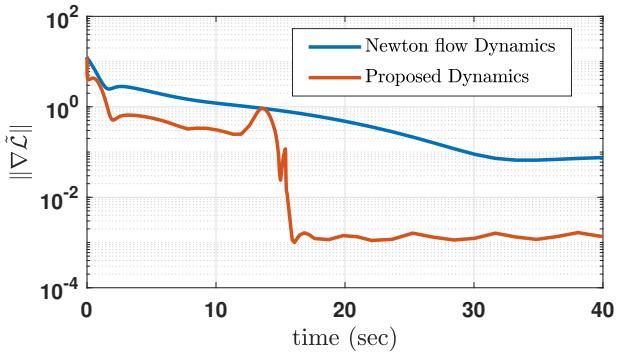


Figure 5.8: Comparison of plot of $\|\nabla\tilde{\mathcal{L}}\|$ on semilog scale between our proposed dynamics to Newton flow dynamics discussed in Fazylab et al. (2017) [3] to show the accuracy of proposed dynamics.

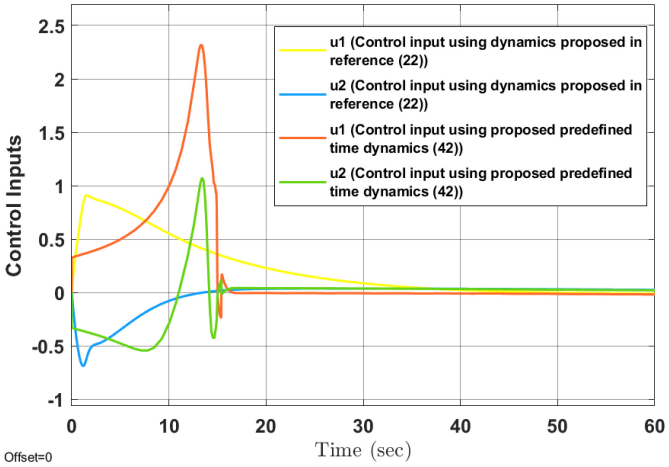


Figure 5.9: Plot of control inputs using controller dynamics (5.2) and dynamics proposed in Arslan and Kod (2016) [4].

5.6 Conclusion

In conclusion, this chapter shows that the robot navigation to a moving target in the presence of obstacles can be formulated as a time-varying optimization problem. The formulated optimization problem is solved using the proposed predefined-time convergent dynamics, and a projected goal or optimal trajectory avoiding obstacles is found in the a priori chosen time. We apply a predefined-time controller to a robot so that it can also start tracking the found optimal projected goal in a user defined time.