A Newton Method for Faster Navigation in Cluttered Environments

Santiago Paternain†, Aryan Mokhtari*, and Alejandro Ribeiro†

***Abstract*— Navigation functions are a common alternative to navigate cluttered environments. The main idea is to combine repulsive potentials from the obstacles and an attractive potential with minimum at the desired destination. By following the negative gradient of the navigation function convergence to the destination while avoiding the obstacles is guaranteed. Rimon-Koditschek artificial potentials are a particular class of potentials that can be tuned to be navigation functions in the case of focally admissible obstacles. While this provides a large class of problems in which they can be used, they suffer from the drawback that by design unstable manifolds of the saddle points have associated Hessian eigenvalues that are smaller than those associated to the stable manifold. This makes the escape from the saddle point to take a large time. To tackle this issue, we propose a second-order method that pre-multiplies the gradient by a modified Hessian to account for the curvature of the function. The method is shown to escape saddles exponentially with base $3/2$ independently of the condition number of the Hessian.**

I. Introduction

Navigating towards a target goal configuration in cluttered environments is a problem that has been extensively studied in the robotics literature. A common approach is the use of artificial potentials which mix an attractive potential to the goal configuration with a repulsive artificial field that pushes the robot away from the obstacles; see e.g. [1]–[13]. In principle, one would expect this idea to have limited applicability because the combination of an attractive potential designed to have a minimum at the target location with a repulsive potential designed to have a maximum at the border of the obstacles is bound to yield a combined potential with multiple critical points. While this does turn out to be the case, artificial potentials work well in practice and, in fact, we have a strong theoretical understanding of why this is the case. When both, the attractive potential and the obstacles, are convex, it is possible to combine them in a way that yields multiple critical points, only one of which is a minimum and can be designed to be arbitrarily close to the target configuration [1], [5], [13], [14]. The remaining critical points are either local maxima or saddles, both of which are not attractors of gradient controllers except for a set that is not dense. Thus, navigating the artificial potential with a gradient controller moves the agent to the target destination while avoiding collisions with the obstacles for almost all initial conditions.

The design of artificial potentials with no local minima is based on Rimon-Koditschek potentials [1]. These have long been known to work for spherical quadratic potentials and spherical holes [1] and have more recently being generalized to focally admissible obstacles [5]. More recent generalizations derive conditions to guarantee absence of local minima in the case of arbitrary convex potential with arbitrary convex obstacles [13] and generalize to situations in which gradients and obstacles are measured with noise. [14]. While these papers guarantee navigability to the target configuration, they make no mention of convergence times. This is important because Rimon-Koditschek potentials eliminate local minima by generating saddle points. While it is true that gradient descent is guaranteed to converge to local minima if the saddle points are non-degenerate [15]–[17], it is well known that it can take long time to escape from saddles [18], [19].

The challenge in escaping saddles is twofold. For once, the presence of a saddle entails existence of a stable manifold for which the saddle is an attractor of the gradient field. This is not a problem in theory because the attractor is not a dense set, but it can be a problem if we initialize close to the stable manifold. This is a relatively tame problem because any noise in the system drives the agent away from the stable manifold with high probability [20], [21] A more challenging problem has to do with the condition number of the saddles defined as the ratio between the largest positive eigenvalue and the smallest negative eigenvalue. It is ready to see that when this condition number is large, escaping the saddle takes an inordinately large amount of time [22]. Essentially, there is an unstable direction that pushes the agent away from the saddle, but the force in that direction – given by the smallest negative eigenvalue – is weak. This is a fundamental problem for Rimon-Koditschek potentials because the saddles have large condition numbers by design.

In this paper, we propose to use the nonconvex Newton (NCN) method [22] to accelerate escape from saddles of the navigation function. NCN uses a descent direction analogous to the Newton step except that it uses the *Positive definite Truncated (PT)-inverse* of the Hessian in lieu of the regular inverse of the Hessian (Definition 3). The PT-inverse has the same eigenvector basis of the regular inverse but its eigenvalues differ in that: (i) All negative eigenvalues are replaced by their absolute values. (ii) Small eigenvalues are replaced by a constant. The idea of using the absolute values of the eigenvalues of the Hessian in nonconvex optimization was first proposed in [23, Chapters 4 and 7] and then in [24], [25]. These properties ensure that the value of the function is reduced at each iteration with an appropriate selection of the step size. In [22] it has been established that NCN can escape any saddle point with eigenvalues bounded away from zero at an exponential rate which can be further shown to

Supported by ARL DCIST CRA W911NF-17-2-0181. †Dept. of Electrical and Systems Engineering, University of Pennsylvania. Email: {spater, aribeiro}@seas.upenn.edu. * Laboratory for Information and Decision Systems, Massachusetts Institute of Technology. Email: aryanm@mit.edu.

have a base of 3/2 independently of the function's properties in a neighborhood of the saddle. These results are formally stated in this paper (Section III) and the proofs can be found in the longer version [22].

II. Navigation Functions

We are interested in navigating a punctured space while reaching a target point defined as the minimum of a convex potential function. Formally, let $\mathcal{X} \in \mathbb{R}^n$ be a non empty compact convex set and let $f : \mathcal{X} \rightarrow \mathbb{R}_+$ be a convex function whose minimum is the agent's goal. Observe that in the particular case where one desires to navigate to a specific destination $\mathbf{x}^*$ one can always use define the function $f(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^*\|^2$. Further consider a set of obstacles $\mathcal{O}_i \subset \mathcal{X}$ with $i = 1 \ldots m$ which are assumed to be open convex sets with nonempty interior and smooth boundary $\partial\mathcal{O}_i$. The free space, representing the set of points accessible to the agent, is then given by the set difference between the space $\mathcal{X}$ and the union of the obstacles $\mathcal{O}_i$,

$$\mathcal{F} \triangleq \mathcal{X} \setminus \bigcup_{i=1}^{m} \mathcal{O}_i. \tag{1}$$

The free space in (1) represents a convex set with convex holes and we assume here that the optimal point is in the interior $\text{int}(\mathcal{F})$ of free space. Further let $t \in [0, \infty)$ denote a time index and let $\mathbf{x}^*$ be the minimum of the objective function, i.e. $\mathbf{x}^* \triangleq \operatorname{argmin}_{x \in \mathbb{R}^n} f_0(x)$. Then, the navigation is to generate a trajectory $\mathbf{x}(t)$ that remains in the free space at all times and reaches $\mathbf{x}^*$ at least asymptotically,

$$\mathbf{x}(t) \in \mathcal{F},\ \forall t \in [0, \infty), \quad \text{and} \quad \lim_{t \to \infty} \mathbf{x}(t) = \mathbf{x}^*. \tag{2}$$

A common approach to problem (2) is to build a navigation function whose formal definition we introduce next [1].

Definition 1 (Navigation Function). *Let $\mathcal{F} \subset \mathbb{R}^n$ be a compact connected analytic manifold with boundary. A map $\varphi : \mathcal{F} \rightarrow [0, 1]$, is a navigation function in $\mathcal{F}$ if:*

Differentiable. *It is twice continuously differentiable in $\mathcal{F}$.*

Polar at $\mathbf{x}^*$. *It has a unique minimum at $\mathbf{x}^*$ which belongs to the interior of the free space, i.e., $\mathbf{x}^* \in int(\mathcal{F})$.*

Morse. *Its critical points on $\mathcal{F}$ are non-degenerate.*

Admissible. *All boundary components have the same maximal value, namely $\partial\mathcal{F} = \varphi^{-1}(1)$.*

The previous properties ensure that the solutions of the controller $\dot{\mathbf{x}} = -\nabla\varphi(\mathbf{x})$ satisfy (2) for almost all initial conditions. We formally state this result in the next Theorem.

Theorem 1 ([26])**.** *Let φ be a navigation function on $\mathcal{F}$. Then, the flow given by the gradient control law*

$$\dot{\mathbf{x}} = -\nabla\varphi(\mathbf{x}), \tag{3}$$

has the following properties:

(i) *$\mathcal{F}$ is a positive invariant set of the flow.*

(ii) *The positive limit set of $\mathcal{F}$ consists of the critical points of φ.*

(iii) *There is a set of measure one, $\tilde{\mathcal{F}} \subset \mathcal{F}$, whose limit set consists of x^*.*

Theorem 1 implies that if $\varphi(\mathbf{x})$ is a navigation function as defined in 1, the trajectories defined by (3) are such that $\mathbf{x}(t) \in \mathcal{F}$ for all $t \in [0, \infty)$ and that the limit of $\mathbf{x}(t)$ is the minimum $\mathbf{x}^*$ for almost every initial condition. This means that (2) is satisfied for almost all initial conditions. We can therefore recast the original problem (2) as the problem of finding a suitable navigation function $\varphi(\mathbf{x})$. To do so, it is convenient to provide a different characterization of free space. To that end, let $\beta_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ be a twice continuously differentiable concave function such that

$$\mathcal{X} = \left\{\mathbf{x} \in \mathbb{R}^n \,\middle|\, \beta_0(\mathbf{x}) \geq 0\right\}. \tag{4}$$

Since the function β_0 is assumed concave its super level sets are convex, thus a function satisfying (4) can always be found because the set $\mathcal{X}$ is also convex. The boundary $\partial\mathcal{X}$, which is given by the set of points for which $\beta_0(\mathbf{x}) = 0$, is called the external boundary of free space. Further consider the m obstacles $\mathcal{O}_i$ and define m twice continuously differentiable convex functions $\beta_i : \mathbb{R}^n \rightarrow \mathbb{R}$ for $i = 1 \ldots m$. The function β_i is associated with obstacle $\mathcal{O}_i$ and satisfies

$$\mathcal{O}_i = \left\{\mathbf{x} \in \mathbb{R}^n \,\middle|\, \beta_i(\mathbf{x}) < 0\right\}. \tag{5}$$

Functions β_i exist because the sets $\mathcal{O}_i$ are convex and the sublevel sets of convex functions are convex.

Given the definitions of the β_i functions in (4) and (5), the free space $\mathcal{F}$ can be written as the set of points at which all of these functions are nonnegative. For a more succinct characterization, define the function $\beta : \mathbb{R}^n \rightarrow \mathbb{R}$ as the product of the $m + 1$ functions β_i,

$$\beta(\mathbf{x}) \triangleq \prod_{i=0}^{m} \beta_i(\mathbf{x}). \tag{6}$$

If the obstacles do not intersect, the function $\beta(\mathbf{x})$ is nonnegative if and only if all of the functions $\beta_i(\mathbf{x})$ are nonnegative. This means that $\mathbf{x} \in \mathcal{F}$ is equivalent to $\beta(\mathbf{x}) \geq 0$ and that we can then define the free space as the set of points for which $\beta(\mathbf{x})$ is nonnegative – when obstacles are nonintersecting. We state next this assumption and definition formally.

Assumption 1 (Obstacles do not intersect). Let $\mathbf{x} \in \mathbb{R}^n$. If for some $i = 1 \ldots m$ we have that $\beta_i(\mathbf{x}) \leq 0$, then $\beta_j(\mathbf{x}) > 0$ for all $j = 0 \ldots m$ with $j \neq i$.

Definition 2 (Free space). *The free space is the set of points $\mathbf{x} \in \mathbb{R}^n$ where the function β in* (6) *is nonnegative,*

$$\mathcal{F} = \left\{\mathbf{x} \in \mathbb{R}^n : \beta(\mathbf{x}) \geq 0\right\}. \tag{7}$$

Observe that we have assumed that the optimal point $\mathbf{x}^*$ is in the interior of free space. We also assume that the objective function f_0 is strongly convex and twice continuously differentiable and that the same is true of the obstacle functions β_i. We state these assumptions formally for later reference.

Assumption 2. The objective function f_0, the obstacle functions β_i and the free space $\mathcal{F}$ are such that:

Optimal point. $\mathbf{x}^* \triangleq \operatorname{argmin}_{\mathbf{x}} f_0(\mathbf{x})$ is such that $f_0(\mathbf{x}^*) \geq 0$ and it is in the interior of the free space.

Twice differential strongly convex objective The function f_0 is twice continuously differentiable and strongly convex in $\mathcal{X}$. The eigenvalues of the Hessian $\nabla^2 f_0(\mathbf{x})$ are thus contained in the interval $[\lambda_{\min}, \lambda_{\max}]$ with $0 < \lambda_{\min}$.

Twice differential strongly convex obstacles The functions β_i are twice continuously differentiable and strongly convex in $\mathcal{X}$. The eigenvalues of the Hessian $\nabla^2 \beta_i(\mathbf{x})$ are thus contained in the interval $[\mu^i_{\min}, \mu^i_{\max}]$ with $0 < \mu^i_{\min}$.

Following the development in [1] we introduce an order parameter $k > 0$ and define the function φ_k as

$$\varphi_k(\mathbf{x}) \triangleq \frac{f_0(\mathbf{x})}{\left(f_0(\mathbf{x})^k + \beta(\mathbf{x})\right)^{1/k}}. \tag{8}$$

In this section we state sufficient conditions such that for large enough order parameter k, the artificial potential (8) is a navigation function in the sense of Definition 1. These conditions relate the bounds on the eigenvalues of the Hessian of the objective function $\lambda_{\min}$ and $\lambda_{\max}$ as well as the bounds on the eigenvalues of the Hessian of the obstacle functions $\mu^i_{\min}$ and $\mu^i_{\max}$ with the size of the obstacles and their distance to the minimum of the objective function x^*. The first result concerns the general case where obstacles are defined through general convex functions.

Theorem 2 ([13])**.** *Let $\mathcal{F}$ be the free space defined in* (7) *satisfying Assumption 1 and $\varphi_k : \mathcal{F} \rightarrow [0,1]$ be the function defined in* (8). *Let $\lambda_{\max}$, $\lambda_{\min}$ and $\mu^i_{\min}$ be the bounds in Assumption 2. Further let the following condition hold for all $i = 1, \ldots, m$ and for all $\mathbf{x}_s$ in the boundary of $\mathcal{O}_i$*

$$\frac{\lambda_{\max}}{\lambda_{\min}} \frac{\nabla \beta_i(\mathbf{x}_s)^T(\mathbf{x}_s - \mathbf{x}^*)}{\|\mathbf{x}_s - \mathbf{x}^*\|^2} < \mu^i_{\min}. \tag{9}$$

Then, for any $\varepsilon > 0$ there exists a constant $K(\varepsilon)$ such that if $k > K(\varepsilon)$, the function φ_k in (8) *is a navigation function with minimum at $\bar{\mathbf{x}}$, where $\|\bar{\mathbf{x}} - \mathbf{x}^*\| < \varepsilon$. Furthermore if $f_0(\mathbf{x}^*) = 0$ or $\nabla \beta(\mathbf{x}^*) = 0$, then $\bar{\mathbf{x}} = \mathbf{x}^*$.*

Theorem 2 establishes sufficient conditions on the obstacles and objective function for which φ_k defined in (8) is guaranteed to be a navigation function for sufficiently large order k. This implies that an agent that follows the flow (3) will succeed in navigating towards $\mathbf{x}^*$ when $f_0(\mathbf{x}^*) = 0$. In cases where this is not the case the agent converges to a neighborhood of $\mathbf{x}^*$. This neighborhood can be made arbitrarily small by increasing k. Notice that the problem of navigating a spherical world to reach a desired destination $\mathbf{x}^*$ [1] can be understood as particular case where the objective function takes the form $\|\mathbf{x} - \mathbf{x}^*\|^2$ and the obstacles are spheres. In this case φ_k is a navigation function for large enough k for every valid world (satisfying Assumption 1), irrespectively of the size and placement of the obstacles. This result can be derived as a corollary of Theorem 2 by showing that condition (9) is always satisfied in the setting of [1].

In this work we are interested in situations in which the agent does not apply control actions continuously but it does in a discrete manner. In that sense, instead of following the gradient flow in (3) its dynamics are given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \nabla \varphi_k(\mathbf{x}_k), \tag{10}$$

where $\eta > 0$ is a step size. While the previous dynamics converge to the global minimizer [16], [17] it performance is degraded in the neighborhood of the saddle points (see e.g., [25]). To overcome this issue we introduce a variation of Newton Method's for non-convex optimization introduced in [22]. The main property of this method is that it ensures exponential escape from the saddle points with a base that is independent of the condition number of the saddle point, i.e., the ratio between the maximum and minimum absolute value of the eigenvalues. We introduce this method and formalize its theoretical properties in the next section.

III. Nonconvex Newton Method (NCN)

To better understand the intuition behind the nonconvex Newton algorithm, let us start discussing the classic Newton iteration for convex optimization. The latter algorithm ameliorates slow convergence of gradient descent by premultiplying gradients with the Hessian inverse. Since the Hessian is positive definite for strongly convex functions, Newton's method provides a descent direction and converges to the minimizer at a quadratic rate. The reason for the improvement in the convergence of Newton's method as compared with gradient descent is that by premultiplying the gradient by the inverse of the Hessian we are performing a local change of coordinates by which the level sets of the function become circular. The algorithm proposed here relies in performing an analogous transformation that turns saddles with "slow" unstable manifolds – this is smaller absolute value of the negative eigenvalues of the Hessian than its positive eigenvalues– into saddles for which the absolute values of the eigenvalues are all the same. This is the case for potentials of the form (8), where the positive eigenvalue of the Hessian at the saddles is of order $O(1)$ and the negative eigenvalues are of order $O(1/k)$. For nonconvex functions the Hessian is not necessarily positive definite and convergence to a minimum is not guaranteed by Newton's method. In fact, all critical points are stable relative to Newton dynamics. This shortcoming can be overcome by adopting a modified inverse using the absolute values of the Hessian eigenvalues [23].

Definition 3 (PT-inverse). *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a symmetric matrix, $\mathbf{Q} \in \mathbb{R}^{n\times n}$ a basis of orthonormal eigenvectors of $\mathbf{A}$, and $\mathbf{\Lambda} \in \mathbb{R}^{n\times n}$ a diagonal matrix of corresponding eigenvalues. We say that $|\mathbf{\Lambda}|_m \in \mathbb{R}^{n\times n}$ is the Positive definite Truncated (PT)-eigenvalue matrix of $\mathbf{A}$ with parameter m if*

$$(|\mathbf{\Lambda}|_m)_{ii} = \begin{cases} |\mathbf{\Lambda}_{ii}| & \textit{if} \quad |\mathbf{\Lambda}_{ii}| \geq m \\ m & \textit{otherwise.} \end{cases} \tag{11}$$

The PT-inverse of $\mathbf{A}$ with parameter m is $|\mathbf{A}|_m^{-1} = \mathbf{Q}\, |\mathbf{\Lambda}|_m^{-1}\, \mathbf{Q}^\top$.

Algorithm 1 Non-Convex Newton Step

1: **function** $[\mathbf{x}_{k+1}] = \text{NCNstep}(\mathbf{x}_k, \alpha, \beta)$
2: Compute $\mathbf{H}_m^{-1} := \mathbf{H}_m(\mathbf{x}_k)^{-1} = \left|\nabla^2\varphi_k(\mathbf{x}_k)\right|_m^{-1}$.
3: Set $\eta_k = 1$. Compute $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \mathbf{H}_m^{-1}\nabla\varphi_k(\mathbf{x}_k)$.
4: **while** $\varphi_k(\mathbf{x}_{k+1}) > \varphi_k(\mathbf{x}_k) - \alpha\eta_k \nabla\varphi_k(\mathbf{x}_k)^\top \mathbf{H}_m^{-1}\nabla\varphi_k(\mathbf{x}_k)$ **do**
5: Reduce step size to $\eta_k = \beta\eta_k$.
6: Compute $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \mathbf{H}_m(\mathbf{x}_k)^{-1}\nabla\varphi_k(\mathbf{x}_k)$.
7: **end while** $\{\text{return } \mathbf{x}_{k+1}\}$

The PT-inverse flips the signs of the negative eigenvalues and truncates small eigenvalues by replacing m for any eigenvalue with absolute value smaller than m. Both of these properties are necessary to obtain a convergent Newton method for nonconvex functions. We use the PT-inverse of the Hessian to define the NCN method. To do so, consider iterates $\mathbf{x}_k$, a step size $\eta_k > 0$, and use the shorthand $\mathbf{H}_m(\mathbf{x}_k)^{-1}$to represent the PT-inverse of the Hessian evaluated at the $\mathbf{x}_k$ iterate. The NCN method is defined by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \mathbf{H}_m(\mathbf{x}_k)^{-1}\nabla\varphi_k(\mathbf{x}_k) \tag{12}$$

The step size η_k is chosen with a backtracking line search as is customary in regular Newton's method; see, e.g., [23]. This yields a step routine that is summarized in Algorithm 1. In Step 3 we update the iterate $\mathbf{x}_k$ using the PT-inverse Hessian $\mathbf{H}_m(\mathbf{x}_k)^{-1}$ computed in Step 2 and initial stepsize $\eta_k = 1$. The updated variable $\mathbf{x}_{k+1}$ is checked against the decrement condition with parameter $\alpha \in (0, 0.5)$ in Step 4. If the condition is not met, we decrease the stepsize η_k by backtracking it with the constant $\beta < 1$ as in Step 5. We update the iterate $\mathbf{x}_k$ with the new stepsize as in Step 6 and repeat the process until the decrement condition is satisfied. Since the PT-inverse is defined to guarantee that $\mathbf{H}_m(x_k)^{-1}\nabla\varphi_k(\mathbf{x}_k)$ is a proper descent direction, it is unsurprising that NCN converges to a local minimum. What it is of interest, is that the convergence is at a faster rate because of the Newton-like correction in (12). Intuitively, the Hessian inverse in convex functions implements a change of coordinates that renders level sets approximately spherical around the current iterate $\mathbf{x}_k$. The Hessian PT-inverse in nonconvex functions implements an analogous change of coordinates that renders level sets in the neighborhood of a saddle point close to a symmetric hyperboloid. We formalize this result next and the hypotheses needed for it to hold.

Assumption 3. Let $\varphi : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable with Lipchitz continuous gradient and Hessian, i.e., there exits constants $M, L > 0$ such that $\forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

$$\begin{aligned} \|\nabla\varphi(\mathbf{x}) - \nabla\varphi(\mathbf{y})\| &\le M\|\mathbf{x}-\mathbf{y}\| \\ \left\|\nabla^2\varphi(\mathbf{x}) - \nabla^2\varphi(\mathbf{y})\right\| &\le L\|\mathbf{x}-\mathbf{y}\|. \end{aligned} \tag{13}$$

Assumption 4. There exists a constant B such that the norm of the local minima $\mathbf{x}^\dagger$ satisfies $\|\mathbf{x}^\dagger\| \le B$ for all $\mathbf{x}^\dagger$.

Assumption 5. Local minima and saddles are non-degenerate, i.e., there exists a constant $m > 0$ such that $\min_{i=1\ldots n}\left|\lambda_i\left(\nabla^2\varphi(\mathbf{x}^\dagger)\right)\right| > 2m$ for all critical points $\mathbf{x}^\dagger$. The notation $\lambda_i(\mathbf{A})$ refers to the i-th eigenvalue of $\mathbf{A}$.

The main feature of the update in (12) is that it exploits curvature information to accelerate the rate for escaping from saddle points relative to gradient descent. In particular, the iterates of NCN escape from a local neighborhood of saddle points exponentially fast at a rate which is independent of the problem's condition number. To state this result formally, let $\mathbf{x}^\ddagger$ be a saddle of interest and denote $\mathbf{Q}_-$ and $\mathbf{Q}_+$ as the orthogonal subspaces associated with the negative and positive eigenvalues of $\nabla^2\varphi_k(\mathbf{x}^\ddagger)$. For a point $\mathbf{x} \neq \mathbf{x}^\ddagger$ we define the gradient projections on these subspaces as

$$\nabla\varphi_k^-(\mathbf{x}) := \mathbf{Q}_-^\top\nabla\varphi_k(\mathbf{x}) \quad \text{and} \quad \nabla\varphi_k^+(\mathbf{x}) := \mathbf{Q}_+^\top\nabla\varphi_k(\mathbf{x}). \tag{14}$$

These projections have different behaviors in the neighborhood of a saddle point. The projection on the positive subspace $\nabla\varphi_k^+(\mathbf{x})$ enjoys an approximately quadratic convergent phase as in Newton's method (cf., [22, Theorem 3.2]). This is because the positive portion of the Hessian is not affected by the PT-inverse. The negative portion $\nabla\varphi_k^-(\mathbf{x})$ can be shown to present an exponential divergence from the saddle point with a rate independent of the problem conditioning. These results provide a bound in the number of steps required to escape the neighborhood of the saddle point as we state next.

Theorem 3 ([22]). *Consider φ as a function satisfying Assumptions 3 and 5. Let $\varepsilon > 0$ be the desired accuracy of the solution provided by Algorithm 2 and $\alpha \in (0,1)$ be one of its inputs. Define $\delta = \min\{m^2(1-2\alpha)/L, m^2/5L\}$. Let the following conditions hold $\|\nabla\varphi(\mathbf{x}_0)\| \le \delta/2$ and*

$$\|\nabla\varphi_-(\mathbf{x}_0)\| \ge \max\{(5L/2m^2)\,\|\nabla\varphi(\mathbf{x}_0)\|^2, \varepsilon\}. \tag{15}$$

Then, we have that $\|\nabla\varphi(\mathbf{x}_{K_1})\| \ge \delta/2$, with

$$K \le 1 + \log_{3/2}\left(\frac{\delta}{2\varepsilon}\right). \tag{16}$$

The result in Theorem 3 establishes an upper bound for the number of iterations needed to escape the saddle point which is of the order $O(\log(1/\varepsilon))$ as long as the iterate $\mathbf{x}_0$ satisfies $\|\nabla\varphi_-(\mathbf{x}_0)\| \ge ((5L)/(2m^2))\,\|\nabla\varphi(\mathbf{x}_0)\|^2$ and $\|\nabla\varphi_-(\mathbf{x}_0)\| \ge \varepsilon$. However, the fundamental result is that the rate at which the iterates escape the neighborhood of the saddle point is a constant $3/2$ independent of the constants of the specific problem. For the previous result to hold condition (15) needs to hold. This condition requires the portion of the energy corresponding to the unstable part of the gradient to be large enough, intuitively this means that the iterate cannot be too close to the stable manifold of the saddle. To make sure that (15) holds with high probability we modify the algorithm nearby in the vicinity of the saddle points, which leads to Algorithm 2. Its main core is the NCN step described in (12) (Step 3) that is performed as long as the iterates are not in a neighborhood of a local minimum satisfying $\|\nabla\varphi(\mathbf{x}_k)\| < \varepsilon$. Steps 4–12 are introduced to add Gaussian

Algorithm 2 Non-Convex Newton Method

```
1: Input: x_k = x_0, accuracy ε > 0 and parameters α ∈ (0, 1/2), β ∈ (0, 1)
2: while ‖∇φ(x_k)‖ > ε or λ_min(∇²φ(x_k)) < 0 do
3:     Update the variable x_{k+1} = NCNstep(x_k, α, β)
4:     if ‖∇φ(x_{k+1})‖ ≤ ε and λ_min(∇²φ(x_{k+1})) < 0 then
5:         x̃ = x_{k+1} + X with X_i ∼ N(0, 2ε/m)
6:         while ‖∇φ(x̃)‖ > (2√n M/m + 1)ε do
7:             x̃ = x_{k+1} + X with X_i ∼ N(0, 2ε/m)
8:         end while
9:         Set x_{k+1} = x̃
10:        if ‖∇φ(x_{k+1})‖ ≤ ε then
11:            Repeat x_{k+1} = NCNstep(x_k, α, β) twice
12:        end if
13:    end if
14: end while
```

noise to satisfy the hypothesis of Theorem 3. If the iterate is in a neighborhood of a saddle point such that $\|\nabla\varphi(\mathbf{x}_k)\| < \varepsilon$ (Step 4), noise from a Gaussian distribution is added (Step 5). The draw is repeated as long as $\|\nabla\varphi(\mathbf{x}_k)\| > (2\sqrt{n}M/m + 1)\varepsilon$ (Steps 6–8). This is done to ensure that the iterates remain close to the saddle point. Once this condition is satisfied we perform the NCN step twice if the iterate is still in the neighborhood $\|\nabla\varphi(\mathbf{x}_k)\| < \varepsilon$ (steps 10–12). In cases where $\|\nabla\varphi(\mathbf{x}_k)\| \geq (5L/2m^2)\|\nabla\varphi(\mathbf{x}_k)\|^2$, two steps of NCN are enough to escape the ε neighborhood of the saddle point and therefore to satisfy the hypothesis of Theorem 3.

Combining the result of Theorem 3 with classic results from Newton's method analysis it is possible to establish a total complexity of order $O(\log(1/p) + \log(1/\varepsilon))$ for NCN to converge to an ε neighborhood of the minimum of $\varphi(\mathbf{x})$ with probability $1 - p$. We formalize this result next.

Theorem 4 (Theorem 2.3 [22])**.** *Consider φ as a function satisfying Assumptions 3–5. Let $\varepsilon > 0$ be the accuracy of the solution and $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$ the remaining inputs of Algorithm 2. Then, with probability $1 - p$ and for any small enough ε Algorithm 2 outputs $\mathbf{x}_{sol}$ satisfying $\|\nabla\varphi(\mathbf{x}_{sol})\| < \varepsilon$ and $\nabla^2\varphi(\mathbf{x}_{sol}) \succ 0$ in at most $O(1/\varepsilon, 1/p)$ iterations.*

The result in Theorem 4 states that the NCN method outputs a solution with a desired accuracy, with a given probability in a number of steps that is of the order of the inverse of the accuracy and the probability. To apply the previous theorem to the navigation functions, we need to show that the potential in (8) satisfies assumptions 3– 5. Likewise, we need to show that the algorithm is such that $\mathbf{x}_k \in \mathcal{F}$ for all $k \geq 0$. We do so in the next section.

IV. NCN for Navigation Functions

In this section we present the main contribution of this work, which is to show that the function defined in (8) satisfies Assumptions 3–5. To do so, we need the following assumption regarding $\nabla^2 f_0(\mathbf{x})$ and $\nabla^2\beta_i$ for all $i = 0 \ldots m$.

Assumption 6. The goal function $f_0(\mathbf{x})$ and each obstacle function $\beta_i(\mathbf{x})$ are such that their Hessian are Lipschitz continuous, i.e. there exist constants $L_f > 0$ and $L_\beta > 0$ such that for all $\mathbf{x}, \mathbf{y} \in \mathcal{F}$ we have that

$$\left\|\nabla^2 f_0(\mathbf{x}) - \nabla^2 f_0(\mathbf{y})\right\| \leq L_f \|\mathbf{x} - \mathbf{y}\|, \qquad (17)$$

and for all $i = 0 \ldots m$ it also holds that

$$\left\|\nabla^2 \beta_i(\mathbf{x}) - \nabla^2 \beta_i(\mathbf{y})\right\| \leq L_\beta \|\mathbf{x} - \mathbf{y}\|, \qquad (18)$$

Observe that the previous assumption is trivially satisfied in the classic setting of [1] where obstacles are spherical and the objective function is the square norm. In this case the Hessian of $f_0(\mathbf{x})$ and each β_i with $i = 1 \ldots m$ is the identity and in the case of $i = 0$ is the negative identity. Hence, the condition is satisfied with $L_f = L_\beta = 0$. Under Assumption 6 we claim next that the navigation function in (8) satisfies the hypothesis of Theorem 4.

Proposition 1. *Let $\mathcal{F}$ be the free space defined in* (7) *satisfying Assumption 1 and let $\varphi_k : \mathcal{F} \to [0, 1]$ be the function defined in* (8)*. Let $\lambda_{\max}$, $\lambda_{\min}$ and $\mu^i_{\min}$ be the bounds in Assumption 2. Further let* (9) *hold for all $i = 1, \ldots, m$ and for all $\mathbf{x}_s$ in the boundary of $\mathcal{O}_i$. Then if Assumption 6 holds, there exists $K > 0$ such that for any $k > K$, $\varphi_k(\mathbf{x})$ satisfies the hypothesis of Theorem 4.*

Proof. By virtue of Theorem 2 it follows that $\varphi_k(\mathbf{x})$ is a navigation function. Hence, by definition (cf., Definition 1) it follows that the function is polar on $\mathcal{F}$ and Morse. Since the set $\mathcal{F}$ is compact it follows that the norm of the minimum of $\varphi_k(\mathbf{x})$ is bounded. Hence Assumption 4 is satisfied. Likewise, the fact that $\varphi_k(\mathbf{x})$ is Morse is equivalent to Assumption 5. We are left to show that the navigation function has Lipschitz gradient and Hessian. Differentiating (8) one can obtain the following gradient

$$\nabla\varphi_k(\mathbf{x}) = \frac{k\beta(\mathbf{x})\nabla f_0(\mathbf{x}) - f_0(\mathbf{x})\nabla\beta(x)}{k\left(f_0(\mathbf{x})^k + \beta(\mathbf{x})\right)^{1+1/k}}. \qquad (19)$$

Observe that the denominator of the previous expression is strictly positive and continuously differentiable for all $\mathbf{x} \in \mathcal{F}$. The latter follows from the fact that $\mathbf{x}^* \in int(\mathcal{F})$ and that $f_0(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathcal{F}$ (cf., Assumption 2. Likewise, the numerator is a continuous differentiable function since $\beta(\mathbf{x})$, $\nabla f_0(\mathbf{x})$, $f_0(\mathbf{x})$ and $\nabla\beta(\mathbf{x})$ are continuously differentiable. It follows, that the gradient is a continuous differentiable function in $\mathcal{F}$. Because the free space is a compact set (cf., (1)) the Hessian of $\varphi_k(\mathbf{x})$ is bounded. Thus the gradient of the navigation function is Lipschitz. We are left to show that the Hessian of the navigation function is Lipschitz as well.

Differentiating (19) we can compute $\nabla^2\varphi_k(\mathbf{x})$

$$\nabla^2\varphi_k(\mathbf{x}) = \frac{k\beta(\mathbf{x})\nabla^2 f_0(\mathbf{x}) + k\nabla\beta(\mathbf{x})\nabla f_0(\mathbf{x})^\top}{k\left(f_0(\mathbf{x})^k+\beta(\mathbf{x})\right)^{1+1/k}} - \frac{f_0(\mathbf{x})\nabla^2\beta(\mathbf{x}) + \nabla f_0(\mathbf{x})\nabla\beta(\mathbf{x})^\top}{k\left(f_0(\mathbf{x})^k+\beta(\mathbf{x})\right)^{1+1/k}} - \left(1+\frac{1}{k}\right)\frac{kf_0(\mathbf{x})^{k-1}\nabla f_0(\mathbf{x}) + \nabla\beta(\mathbf{x})}{f_0(\mathbf{x})^k+\beta(\mathbf{x})}\nabla\varphi_k(\mathbf{x})^\top. \tag{20}$$

As it was previously argued, the term $f_0(\mathbf{x})^k+\beta(\mathbf{x})$ is strictly positive for every $\mathbf{x}\in\mathcal{F}$. Hence, to show that $\nabla^2\varphi_k(\mathbf{x})$ is Lipschitz it suffices to show that the numerators in (20) are. Observe that because the functions $\beta(\mathbf{x})$ and $f_0(\mathbf{x})$ are twice continuously differentiable (cf., Assumption 2) and the free space is a compact set (cf., (1)) and that $f_0(\mathbf{x})$ all the terms in the numerator expect those related to the Hessian of $f_0(\mathbf{x})$ and $\beta(\mathbf{x})$ have bounded derivatives. And thus, they are Lipschitz. The fact that the terms involving $\nabla^2 f_0(\mathbf{x})$ and $\nabla^2\beta(\mathbf{x})$ are Lipschitz as well follow from Assumption 6. Which concludes the proof of the proposition. ■

The previous result establishes that the navigation function in (8) satisfies the hypotheses of Theorem (4) and hence if we run NCN (Algorithm 2 the fast escape from the saddle points will be observed. For the latter to be true, we require that all the iterates of NCN remain in the free space. This is the subject of the following proposition.

Proposition 2. *Let $\mathcal{F}$ be the free space defined in* (7) *satisfying Assumption 1 and let $\varphi_k:\mathcal{F}\to[0,1]$ be the function defined in* (8)*. Then, the iterates of Algorithm 2 are such that $\mathbf{x}_k\in\mathcal{F}$ for all $k\geq 0$.*

Proof. We will show this result by induction. Assume that for some $k\geq 0$ we have that $\mathbf{x}_k\in\mathcal{F}$ we will show that the same holds for $\mathbf{x}_{k+1}$. Notice that by definition of the artificial potential (8) we have that $\mathbf{x}_k\in\mathcal{F}$ if and only if $\varphi_k(\mathbf{x}_k)\leq 1$ and that Step 4 in Algorithm 1 ensures that

$$\varphi_k(\mathbf{x}_{k+1}) \leq \varphi_k(\mathbf{x}_k) - \alpha\eta_k\nabla\varphi_k(\mathbf{x}_k)^\top\mathbf{H}_m(\mathbf{x}_k)^{-1}\nabla\varphi_k(\mathbf{x}). \tag{21}$$

By definition of the PT-inverse (cf., Definition 3), $\mathbf{H}_m(\mathbf{x}_k)$ is a positive definite matrix, and therefore we have that

$$\alpha\eta_k\nabla\varphi_k(\mathbf{x}_k)^\top\mathbf{H}_m(\mathbf{x}_k)^{-1}\nabla\varphi_k(\mathbf{x}) \geq 0. \tag{22}$$

The last two bounds imply that $\varphi_k(\mathbf{x}_{k+1})\leq\varphi_k(\mathbf{x}_k)\leq 1$, which in turn allows us to claim that $\mathbf{x}_{k+1}\in\mathcal{F}$. The proof if then completed because $\mathbf{x}_0\in\mathcal{F}$. ■

The previous proposition allows us to guarantee that the iterates of the algorithm remain in the free space, hence avoiding collisions. We next claim that NCN applied to the problem of navigation functions is such that it ensures safe navigation and that it converges to the desired destination in a number of iterations that is logarithmic with the inverse of the accuracy and the inverse of the probability of success.

Theorem 5. *Let $\mathcal{F}$ be the free space defined in* (7) *satisfying Assumption 1 and let $\varphi_k:\mathcal{F}\to[0,1]$ be the function defined in* (8)*. Let $\lambda_{\max}$, $\lambda_{\min}$ and $\mu^i_{\min}$ be the bounds in Assumption 2. Further let* (9) *hold for all $i=1,\ldots,m$ and for all $\mathbf{x}_s$ in the boundary of $\mathcal{O}_i$. If Assumption 6 holds, there exists $K>0$ such that for any $k>K$ the iterations of Algorithm 2 remain in the free space for all $k\geq 0$ and with probability $1-p$ it outputs a solution $\mathbf{x}_{sol}$ such that $\|\nabla\varphi(\mathbf{x}_{sol})\|<\varepsilon$ and $\nabla^2\varphi(\mathbf{x}_{sol})\succ 0$ in at most $O(1/\varepsilon,1/p)$ iterations.*

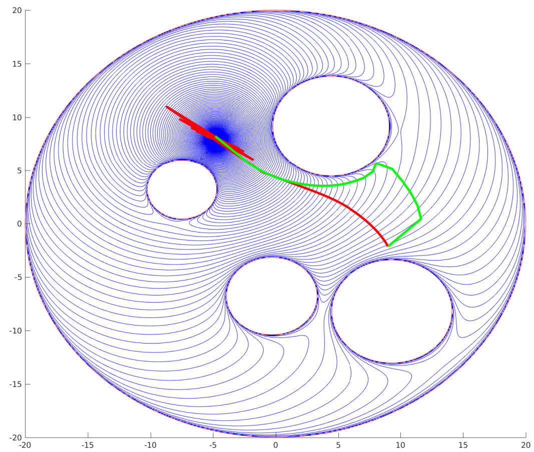
Proof. Follows from Propositions 1 and 2 and Theorem 4. ■

V. Numerical Experiments

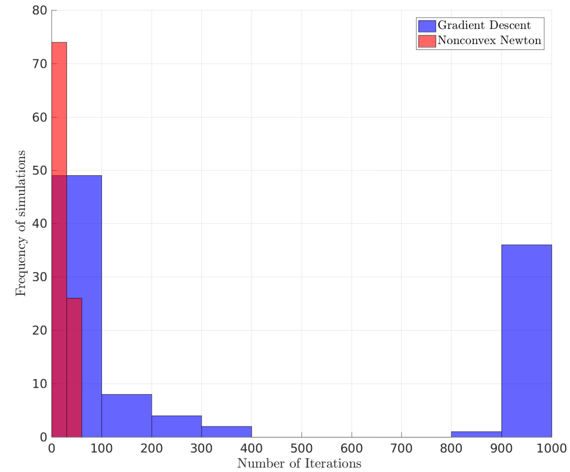
In this section we compare the performances of NCN and gradient descent for a navigation function of the form in (8). The experimental set up is the following. For all simulations we define the workspace to be an n-dimenstional sphere centered at the origin and radius $r_0>0$. For each simulation we draw at random the centers of the obstacles $\mathbf{c}_i\in\mathbb{R}^n$ from a uniform distribution over $[-r_0/2, r_0/2]^n$ and their radius a r_i also from a uniform distribution over $[1, 1+r_0/5]$. We verify that obstacles do not intersect with each other and they are contained in the workspace. If this is not the case we draw the parameters of the obstacles again. Likewise, we draw the desired destination $\mathbf{x}^*$ uniformly from $[r_0/2, r_0/2]^n$ and we verify that it is in the free space. If not we draw it again. Finally, we draw the initial condition uniformly from $[-r_0/2, r_0/2]^n$ and we make sure that it is in the free space. The parameters selected are $n=2$, $m=4$, $r_0=20$ and $k=10$. In Figure 1a we can observe the result of a simulation where we compare the trajectory that arises from following the gradient system (10) and the one that arises from NCN (Algorithm (2)). In both cases we require a line search algorithm as described in Steps 4–7 in Algorithm 1. The parameters selected are $\alpha=0.01$ and $\beta=0.9$. One of the advantages of NCN is that it is a scale free algorithm – as it is classic Newton – meaning that if we scale the function the sequence of iterates remains invariant. This is not the case for gradient descent. In particular, without any scaling after one thousand iterations it fails to converge for the example in Figure 1a. To obtain that trajectory we had to scale the gradient by 1.37×10^7, the latter produces an oscillatory behavior nearby the minimum. The previous scaling provided a reasonable trade off between convergence speed and oscillations. We run 100 simulations for both gradient descent and NCN. The initial conditions are drawn at each instance of the simulation and it is the same for both methods. The stopping conditions are $\|\mathbf{x}-\mathbf{x}^*\|<0.01$ or a maximum of 1000 iterations. As it can be observed from the histogram representing the frequency of iterations that each algorithm takes for convergence (cf., Figure 1b) we can conclude that NCN converges faster than gradient descent and that the latter gets stuck in 35% of the cases while NCN never does.

VI. Conclusion

In this work we considered the problem of navigating towards a desired configuration in a cluttered environment.



(a) Paths for gradient descent (red) and Nonconvex Newton (green).



(b) Number of iterations for convergence .

Fig. 1: Comparison of gradient descent and NCN for a potential of the form (8) with parameter $k = 10$. The stopping criteria for both algorithms is $\|\mathbf{x} - \mathbf{x}^*\| < 0.01$. Nonconvex Newton converges in less iterations as expected and never gets stuck.

In particular, using navigation functions of the Rimon-Koditschek form is guarantee to solve the problem, the performance of gradient flows of these potentials suffer from ill conditioning of the Hessian at the saddle points. To overcome this limitation we used Nonconvex Newton, where the negative gradient is pre-multiply by the PT-inverse. Convergence of this method is faster both in theory and in the practice when compared to gradient descent.

References

[1] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Advances in Applied Mathematics*, vol. 11, no. 4, pp. 412–442, 1990.

[2] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 501–518, 1992.

[3] O. Khatib, *Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles*. PhD thesis, 1980.

[4] I. F. Filippidis and K. J. Kyriakopoulos, "Navigation functions for everywhere partially sufficiently curved worlds," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 2115–2120, IEEE, 2012.

[5] I. Filippidis and K. J. Kyriakopoulos, "Navigation functions for focally admissible surfaces," in *American Control Conference (ACC), 2013*, pp. 994–999, IEEE, 2013.

[6] I. Filippidis, *Navigation functions for unknown sphere worlds, general geometries, their inverse problem and combination with formal methods*. PhD thesis, National Technical University of Athens, 2011.

[7] I. Filippidis and K. J. Kyriakopoulos, "Adjustable navigation functions for unknown sphere worlds," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pp. 4276–4281, IEEE, 2011.

[8] S. G. Loizou, "Navigation functions in topologically complex 3-d workspaces," in *American Control Conference (ACC), 2012*, pp. 4861–4866, IEEE, 2012.

[9] S. G. Loizou, "Closed form navigation functions based on harmonic potentials," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pp. 6361–6366, IEEE, 2011.

[10] G. Lionis, X. Papageorgiou, and K. J. Kyriakopoulos, "Locally computable navigation functions for sphere worlds," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 1998–2003, IEEE, 2007.

[11] G. Lionis, X. Papageorgiou, and K. J. Kyriakopoulos, "Towards locally computable polynomial navigation functions for convex obstacle workspaces," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 3725–3730, IEEE, 2008.

[12] G. Roussos and K. J. Kyriakopoulos, "Decentralized and prioritized navigation and collision avoidance for multiple mobile robots," in *Distributed Autonomous Robotic Systems*, pp. 189–202, Springer, 2013.

[13] S. Paternain, D. E. Koditschek, and A. Ribeiro, "Navigation functions for convex potentials in a space with convex obstacles," *IEEE Transactions on Automatic Control*, 2017.

[14] S. Paternain and A. Ribeiro, "Stochastic artificial potentials for online safe navigation," *arXiv preprint arXiv:1701.00033*, 2016.

[15] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential equations, dynamical systems, and an introduction to chaos*, vol. 60. Academic press, 2004.

[16] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent only converges to minimizers," in *Conference on Learning Theory*, pp. 1246–1257, 2016.

[17] I. Panageas and G. Piliouras, "Gradient descent converges to minimizers: The case of non-isolated critical points," *arXiv preprint arXiv:1605.00405*, 2016.

[18] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Advances in neural information processing systems*, pp. 2933–2941, 2014.

[19] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, "The loss surfaces of multilayer networks.," in *AISTATS*, 2015.

[20] R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points-online stochastic gradient for tensor decomposition.," in *COLT*, pp. 797–842, 2015.

[21] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, "How to escape saddle points efficiently," *arXiv preprint arXiv:1703.00887*, 2017.

[22] S. Paternain, A. Mokhtari, and A. Ribeiro, "A second order method for nonconvex optimization," *arXiv preprint arXiv:1707.08028*, 2017.

[23] J. Nocedal and S. Wright, "Numerical optimization: Springer science & business media," *New York*, 2006.

[24] W. Murray, "Newton-type methods," *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[25] R. Pascanu, Y. N. Dauphin, S. Ganguli, and Y. Bengio, "On the saddle point problem for non-convex optimization," *arXiv preprint arXiv:1405.4604*, 2014.

[26] D. E. Koditschek, "Strict global lyapunov functions for mechanical systems," 1988.