J. Nonlinear Var. Anal. 5 (2021), No. 5, pp. 831-849
Available online at http://jnva.biemdas.com
https://doi.org/10.23952/jnva.5.2021.5.13

INERTIAL-TYPE PROJECTION METHODS FOR SOLVING CONVEX CONSTRAINED MONOTONE NONLINEAR EQUATIONS WITH APPLICATIONS TO ROBOTIC MOTION CONTROL

ABUBAKAR BAKOJI MUHAMMAD[1,3], CHRISTIANE TAMMER[1,*], ALIYU MUHAMMED AWWAL[2,3], ROSALIND ELSTER[4], ZHAOLI MA[5]

[1]*Faculty of Natural Sciences II, Institute of Mathematics, Martin Luther University Halle-Wittenberg, 06099 Halle (Saale), Germany*
[2]*Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), Pracha-Uthit Road, Bang Mod, Thung Khru, Bangkok 10140, Thailand*
[3]*Department of Mathematics, Faculty of Science, Gombe State University, Gombe 760214, Nigeria*
[4]*06116 Halle (Saale), Germany*
[5]*College of Statistics and Mathematics, Yunnan University of Finance and Economics, Kunming, China*

Abstract. In this paper, we introduce two derivative-free projection iterative algorithms for solving a system of nonlinear monotone operator equations. The two proposed algorithms can be viewed as two-step methods where the first step uses an inertial effect in every iteration. The global convergence of the proposed algorithms is established under some mild assumptions. We present numerical experiments to show the efficiency and advantage of the inertial projection steps of the proposed algorithms and compare it with some existing methods for solving nonlinear problems. Finally, we consider the problem of solving a motion control problem involving a two-joint planar robotic manipulator.

Keywords. Derivative-free method; Inertial method; Motion control problem; Nonlinear monotone equations; Projection method.

1. INTRODUCTION

Let $x_0 \in \mathbb{R}^n$ be a starting point. Consider the iterative algorithm

$$x_{k+1} := x_k + \eta_k d_k,\ k = 0,1,2,\ldots, \tag{1.1}$$

where $\eta_k > 0$ denotes a step length and $d_k \in \mathbb{R}^n$ denotes a search direction. Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a continuous function, the search direction in (1.1) is usually computed as follows

$$d_k := \begin{cases} -F(x_k), & \text{for } k = 0, \\ -\theta_k F(x_k) + \beta_k s_{k-1}, & \text{for } k = 1,2,3,\ldots, \end{cases} \tag{1.2}$$

*Corresponding author.
E-mail addresses: abubakar.muhammad@mathematik.uni-halle.de (A.B. Muhammad), christiane.tammer@mathematik.uni-halle.de (C. Tammer), aliyumagsu@gmail.com (A.M. Awwal), r.elster@t-online.de (R. Elster), kmszmzl@126.com (Z. Ma).

Received April 29, 2021; Accepted June 30, 2021.

where $s_{k-1} := x_k - x_{k-1}$, $\theta_k > 0$ and $\beta_k \in \mathbb{R}$ are referred to be spectral parameter and conjugate gradient parameter, respectively. Iterative method (1.1) with the search direction d_k defined in (1.2) is referred to a spectral-conjugate gradient method. If parameter $\theta_k = 1$, for all k, then (1.1) is called a conjugate gradient method while if $\beta_k = 0$, for all k, then (1.1) is called a spectral gradient method. Iterative method (1.1) has been investigated by researchers (see, e.g., [1, 2]) that it is very efficient for solving unconstrained optimization problems $\min\{f(x) : x \in \mathbb{R}^n\}$, where f is a continuous differentiable function that is assumed to be bounded from below.

Definition 1.1. A function $F : \mathbb{R}^n \to \mathbb{R}^n$ is said to be monotone if, for all $x, y \in \mathbb{R}^n$,

$$\langle F(x) - F(y),\ x - y\rangle \geq 0.$$

Definition 1.2. A function $F : \mathbb{R}^n \to \mathbb{R}^n$ is said to be Lipschitz continuous if there exists some positive constants, say L, such that, for all $x, y \in \mathbb{R}^n$,

$$\|F(x) - F(y)\| \leq L\|x - y\|.$$

Recently, with the aid of the hyperplane projection proposed by Solodov and Svaiter in [3], some researchers (see, e.g., [4, 5, 6]) extended iterative method (1.1) to solve the following nonlinear system of equations

$$F(x) = 0,\ \text{such that } x \in \Lambda \subseteq \mathbb{R}^n, \tag{1.3}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is assumed to be Lipschitz continuous and monotone. The solution set of problem (1.3) is assumed to be nonempty and is denoted as χ. The feasible set Λ is assumed to be nonempty, closed and convex and therefore, the set χ is also convex [7]. Various problems arising in different areas and applications, such as optimization, differential equations, motion control problems, ℓ_1-norm regularization problems and so on can be reformulated into problem (1.3) (see, [8, 9, 10, 11]).

On the other hand, given two starting points, say x_0 and x_{-1}, we consider an inertial step

$$w_k := x_k + \alpha_k(x_k - x_{k-1}). \tag{1.4}$$

Iterative algorithms that incorporate inertial step (1.4) are popularly referred as *inertial-type algorithms*. These algorithms were originated from the heavy-ball method of the second-order-in-time dissipative dynamical system. In 1964, Polyak [12] considered the inertial extrapolation as a speed-up method to solve smooth convex minimization problems. Inertial-type methods are two-step iterative schemes, and the next iterate is defined by making use of the previous two iterates [13]. In order to speed up the iteration process, an inertial extrapolation term is required to boost the iterative sequence. These inertial-type methods are basically used to accelerate the iterative sequence towards the required solution. Recently, there is a growing interest in studying inertial-type algorithms for variational inequalities and monotone inclusions; see, e.g., [14, 15, 16, 17, 18, 19] and the references therein. Various studies have shown that iterative algorithms for solving the above nonlinear problems with an inertial step have better numerical performance in terms of the number of iterations and a time of execution compared to their counterparts without the inertial step. These two impressive advantages enhance the researcher's interest in developing new inertial-like methods.

Given a starting point, say x_0, classical iterative algorithms (such as Newton's method and its variants as well as quasi-Newton methods and so on) use formula (1.1) to update their iterative sequences. The search direction d_k in (1.1) is usually updated via x_k and its preceding point

x_{k-1} as well as their images, that is, $F(x_k)$ and $F(x_{k-1})$ (see, e.g., [6, 11, 20, 21]). Following the research line on inertial step (1.4) in variational inequalities, split feasibility problems and so on, we pose the following question. Can a search direction incorporated with inertial effect (1.4) improve the numerical performance of the conjugate gradient-like algorithms?. To answer this question, for a given step size $\alpha_k \in [0,1]$ and any two starting points, say x_{-1} and x_0, we compute the sequence of inertial steps $\{w_k\}$ as well as their images via $w_k := x_k + \alpha_k(x_k - x_{k-1})$, and then use them to build the search direction of the proposed algorithm.

In this paper, based on the projection technique considered by Solodov and Svaiter, we propose two inertial-type algorithms for solving a system of monotone nonlinear equations with convex constraints. One of them is an inertial-type conjugate gradient projection algorithm and the other one is an inertial-type spectral gradient projection algorithm. We propose two search directions that are sufficiently descent and bounded. Furthermore, we consider solving a motion control problem involving a two-joint planar robotic manipulator. Throughout this paper, we denote by $\mathbb{R}^n_+$, $\|\cdot\|_2$ and $\langle\cdot,\cdot\rangle$ the set $\{(x_1,x_2,\ldots,x_n)^T \in \mathbb{R}^n \mid x_i \geq 0,\ i = 1,2,\ldots,n\}$, the Euclidean norm in $\mathbb{R}^n$ and the Euclidean inner product in $\mathbb{R}^n$, respectively.

The remaining part of this paper is organized as follows: In Section 2, we describe our proposed method, its algorithm as well as its global convergence. In Section 3, we report numerical experiments to show the efficiency of our algorithms as well as the application of the proposed algorithms in motion control problems. In Section 4, the last section, we give our conclusion in this paper.

2. Inertial-type Derivative-free Algorithms and Their Convergence Analysis

Let x_0 and x_{-1} be two given starting points, and let $w_k := x_k + \alpha_k(x_k - x_{k-1})$ be an inertial step, where $\alpha_k \in [0,1]$. We begin this section by stating the following assumptions, which are vital in the convergence analysis of the proposed algorithms.

Assumption 2.1.

A. The solution set χ of problem (1.3) is nonempty.
B. The function $F : \mathbb{R}^n \to \mathbb{R}^n$ is monotone and Lipschitz continuous.

If the search direction d_k given by (1.2) satisfies the following sufficient descent condition

$$F(x_k)^T d_k \leq -t\|F(x_k)\|^2,\ t > 0, \tag{2.1}$$

then, it is said to be a descent direction (see, e.g., [22, 23, 24]). It is worth mentioning that inequality (2.1) is very important for conjugate gradient-like iterative algorithms to be globally convergent.

In order to state the proposed algorithms, the following projection operator is of great importance. Given any point $x \in \mathbb{R}^n$, its projection onto the feasible set $\Lambda \subseteq \mathbb{R}^n$ is defined as $P_\Lambda(x) := \operatorname{argmin}\{\|x-y\| : y \in \Lambda\}$, which satisfies the following properties

$$\|P_\Lambda(x) - P_\Lambda(y)\| \leq \|x-y\|, \text{ for all } x,y \in \mathbb{R}^n. \tag{2.2}$$

$$\|P_\Lambda(x) - y\| \leq \|x-y\|, \text{ for all } y \in \Lambda. \tag{2.3}$$

We now state the details of the proposed algorithms for solving problem (1.3).

Algorithm 1: Conjugate Gradient (CG) Algorithm with Inertial–Step (CGAIS)

Input: Give $x_{-1},\ x_0 \in \Lambda,\ \kappa, r > 0,\ \gamma \in (0,2),\ \sigma, \rho \in (0,1)$, the stopping tolerance $Tol \geq 0$, and $\alpha_k \in [0,1]$.

Step 0: Set $k = 0$, and compute $d_0 := -F(x_0)$ and $w_0 := x_0 + \alpha_0(x_0 - x_{-1})$.

Step 1: If $\|F(x_k)\| \leq Tol$, then x_k is a solution and the iteration process stops.

Step 2: Set

$$\upsilon_k := x_k + \eta_k d_k \text{ and } \eta_k := \kappa\rho^i, \tag{2.4}$$

where i is the smallest non–negative integer such that

$$-\langle F(x_k + \kappa\rho^i d_k),\ d_k\rangle \geq \sigma\kappa\rho^i \|d_k\|^2 \|F(x_k + \kappa\rho^i d_k)\|^{1/c},\ c \geq 1. \tag{2.5}$$

Step 3: If $\|F(\upsilon_k)\| = 0$, stop. Else, compute the next iterate

$$x_{k+1} := P_\Lambda\left[x_k - \gamma\frac{\langle F(\upsilon_k),\ x_k - \upsilon_k\rangle}{\|F(\upsilon_k)\|^2}F(\upsilon_k)\right],\ \|F(\upsilon_k)\| \neq 0. \tag{2.6}$$

Step 4: Compute $w_{k+1} := x_{k+1} + \alpha_k(x_{k+1} - x_k)$.

Step 5: Set $k := k+1$, update the search direction and repeat the process from *Step 1*.

$$d_k := -\widehat{\theta}_k F(x_k) + \beta_k(w_k - w_{k-1}) - u_{k-1}z_{k-1}, \tag{2.7}$$

$$\beta_k := \frac{\langle z_{k-1},\ F(x_k)\rangle}{\langle z_{k-1},\ w_k - w_{k-1}\rangle},\ w_k \neq w_{k-1}, \tag{2.8}$$

$$u_{k-1} := \frac{\langle w_k - w_{k-1},\ F(x_k)\rangle}{\langle z_{k-1},\ w_k - w_{k-1}\rangle},\ w_k \neq w_{k-1}, \tag{2.9}$$

where

$$z_{k-1} := F(w_k) - F(w_{k-1}) + r(w_k - w_{k-1}), \tag{2.10}$$

$$\widehat{\theta}_k := \begin{cases} \lambda_k, & \text{if } \theta_k \leq 0 \text{ or } \langle z_{k-1},\ F(x_k)\rangle = 0, \\ \theta_k, & \text{otherwise,} \end{cases} \tag{2.11}$$

$$\theta_k := \frac{1}{\langle z_{k-1},\ F(x_k)\rangle}\left\langle F(x_k),\ w_k - w_{k-1} + z_{k-1} - \frac{w_k - w_{k-1}}{\langle w_k - w_{k-1},\ z_{k-1}\rangle}\|z_{k-1}\|^2\right\rangle, \tag{2.12}$$

$$\lambda_k := \frac{\|w_k - w_{k-1}\|^2}{\langle z_{k-1},\ w_k - w_{k-1}\rangle}. \tag{2.13}$$

Algorithm 2: Spectral Algorithm with Inertial–Step (SAIS)

Input: Give the same inputs as in Algorithm 1.

Realize *Step 0* to *Step 4* of Algorithm 1, but replace *Step 5* by:

Step 5: Compute

$$d_k := -\widehat{\theta}_k F(x_k), \tag{2.14}$$

where $\widehat{\theta}_k$ is defined in (2.11).

Remark 2.1. Observe that Algorithm 2 can be obtained by setting $\beta_k = u_{k-1} = 0$ in (2.7) of Algorithm 1.

Remark 2.2. From the definition of z_{k-1} and the monotonicity assumption on F, we have

$$\begin{aligned}\langle z_{k-1},\ w_k - w_{k-1}\rangle &= \langle F(w_k) - F(w_{k-1}) + r(w_k - w_{k-1}),\ w_k - w_{k-1}\rangle \\ &= \langle F(w_k) - F(w_{k-1}),\ w_k - w_{k-1}\rangle + r\langle w_k - w_{k-1},\ w_k - w_{k-1}\rangle \\ &\geq r\|w_k - w_{k-1}\|^2 > 0, \text{ if } w_k \neq w_{k-1}. \end{aligned} \tag{2.15}$$

Next, we show that θ_k is well-defined. We begin by showing that λ_k (according to (2.13)) is bounded. From the Lipschitz continuity of F, we have

$$\langle z_{k-1},\ w_k - w_{k-1}\rangle = \langle F(w_k) - F(w_{k-1}),\ w_k - w_{k-1}\rangle + r\|w_k - w_{k-1}\|^2 \leq (L+r)\|w_k - w_{k-1}\|^2. \tag{2.16}$$

From (2.15) and (2.16) we have

$$\frac{1}{L+r} \leq \frac{\|w_k - w_{k-1}\|^2}{\langle z_{k-1},\ w_k - w_{k-1}\rangle} \leq \frac{1}{r}. \tag{2.17}$$

This implies that λ_k is bounded. Therefore, from (2.11), we can find some constants, say $t_1 > 0$ and $t_2 > 0$, such that

$$t_1 \leq \theta_k \leq t_2. \tag{2.18}$$

Combining (2.17) and (2.18) gives

$$p \leq \widehat{\theta}_k \leq q, \tag{2.19}$$

where $p \in [t_1, \frac{1}{L+r}]$ and $q \in [\frac{1}{r}, t_2]$.

The following lemma shows that the proposed search directions satisfy (2.1) is independent of the line search strategy used.

Lemma 2.1. *The search directions* (2.7) *and* (2.14) *generated by Algorithms 1 and 2 satisfy the descent condition defined by* (2.1).

Proof. Taking the inner product of the search direction d_k defined by (2.7) with $F(x_k)$, for $k = 0$, it follows that $\langle F(x_0),\ d_0\rangle \leq -\|F(x_0)\|^2$. For $k > 0$, we have

$$\begin{aligned}\langle F(x_k),\ d_k\rangle &= -\widehat{\theta}_k\|F(x_k)\|^2 + \frac{\langle z_{k-1},\ F(x_k)\rangle}{\langle z_{k-1},\ w_k - w_{k-1}\rangle}\langle F(x_k),\ w_k - w_{k-1}\rangle \\ &\quad - \frac{\langle w_k - w_{k-1},\ F(x_k)\rangle}{\langle z_{k-1},\ w_k - w_{k-1}\rangle}\langle F(x_k),\ z_{k-1}\rangle \\ &= -\widehat{\theta}_k\|F(x_k)\|^2 \\ &\leq -p\|F(x_k)\|^2. \end{aligned} \tag{2.20}$$

The last inequality follows from (2.19). This shows that the search direction d_k is a descent direction. □

Remark 2.3. Line search (2.5) is well-defined. That is, for all $k \geq 0$, there always exists a step-size η_k satisfying (2.5) in a finite number of iteration.

Suppose on the contrary that there exists some k_0 such that, for any $i = 0, 1, 2, \ldots$, (2.5) does not hold, that is,

$$-\langle F(x_{k_0} + \kappa\rho^i d_{k_0}),\ d_{k_0}\rangle < \sigma\kappa\rho^i\|d_{k_0}\|^2\|F(x_{k_0} + \kappa\rho^i d_{k_0})\|^{1/c},\ \kappa > 0 \text{ and } c \geq 1. \tag{2.21}$$

From the continuity of F and the fact that $0 < \rho^i < 1$, $(i = 0,1,2,\ldots)$, we let $i \to \infty$, which together with (2.21) yields

$$\langle F(x_{k_0}),\ d_{k_0}\rangle \geq 0. \tag{2.22}$$

It is clear that inequality (2.22) contradicts (2.1). Hence line search (2.5) is well-defined.

Lemma 2.2. *Suppose that Assumption 2.1 holds and let $0 < \gamma < 2$. Let $\hat{x}$ be a solution to problem (1.3). If the sequences $\{d_k\}$, $\{\upsilon_k\}$, $\{x_k\}$ and $\{w_k\}$ are generated by (2.7), (2.4) and (2.6) as well as the sequences of scalars $\{\alpha_k\}$ and $\{\eta_k\}$ in Algorithm 1 or 2 with the Lipschitz constant L, then the following assertions hold:*

(i) *$\{x_k\}$ and $\{w_k\}$ are bounded and $\lim_{k\to\infty} \|x_k - \hat{x}\|$ exists.*
(ii) *The sequence of the search direction $\{\|d_k\|\}$ is bounded.*
(iii) *$\{\upsilon_k\}$ and $\{\|F(\upsilon_k)\|\}$ are bounded.*
(iv) $\lim_{k\to\infty} \eta_k \|d_k\| = 0$.

Proof. (i). From projection property (2.3), we have

$$\left\| P_\Lambda\left(x_k - \gamma \frac{\langle F(\upsilon_k),\ x_k - \upsilon_k\rangle}{\|F(\upsilon_k)\|^2} F(\upsilon_k)\right) - \hat{x}\right\| \leq \left\| x_k - \gamma \frac{\langle F(\upsilon_k),\ x_k - \upsilon_k\rangle}{\|F(\upsilon_k)\|^2} F(\upsilon_k) - \hat{x}\right\|. \tag{2.23}$$

Since $\hat{x}$ is a solution of problem (1.3), then $F(\hat{x}) = 0$. Therefore, $\langle F(\hat{x}),\ \upsilon_k - \hat{x}\rangle = 0$. By the monotonicity of F, we have $\langle F(\hat{x}),\ \upsilon_k - \hat{x}\rangle \leq \langle F(\upsilon_k),\ \upsilon_k - \hat{x}\rangle$. This means

$$\begin{aligned} \langle F(\upsilon_k),\ x_k - \upsilon_k\rangle &= \langle F(\upsilon_k),\ x_k - \upsilon_k\rangle + \langle F(\hat{x}),\ \upsilon_k - \hat{x}\rangle \\ &\leq \langle F(\upsilon_k),\ x_k - \upsilon_k\rangle + \langle F(\upsilon_k),\ \upsilon_k - \hat{x}\rangle \\ &= \langle F(\upsilon_k),\ x_k - \hat{x}\rangle. \end{aligned} \tag{2.24}$$

By (2.23), (2.24) and the definition of x_{k+1} in (2.6), we get

$$\begin{aligned} \|x_{k+1} - \hat{x}\|^2 &\leq \left\| x_k - \hat{x} - \gamma \frac{\langle F(\upsilon_k),\ x_k - \upsilon_k\rangle}{\|F(\upsilon_k)\|^2} F(\upsilon_k)\right\|^2 \\ &= \|x_k - \hat{x}\|^2 - 2\gamma \frac{\langle F(\upsilon_k),\ x_k - \upsilon_k\rangle}{\|F(\upsilon_k)\|^2} \langle F(\upsilon_k),\ x_k - \hat{x}\rangle + \gamma^2 \frac{\langle F(\upsilon_k),\ x_k - \upsilon_k\rangle^2}{\|F(\upsilon_k)\|^2} \\ &\leq \|x_k - \hat{x}\|^2 - 2\gamma \frac{\langle F(\upsilon_k),\ x_k - \upsilon_k\rangle}{\|F(\upsilon_k)\|^2} \langle F(\upsilon_k),\ x_k - \upsilon_k\rangle + \gamma^2 \frac{\langle F(\upsilon_k),\ x_k - \upsilon_k\rangle^2}{\|F(\upsilon_k)\|^2} \\ &= \|x_k - \hat{x}\|^2 - \gamma(2-\gamma) \frac{\langle F(\upsilon_k),\ x_k - \upsilon_k\rangle^2}{\|F(\upsilon_k)\|^2} \qquad (2.25) \\ &\leq \|x_k - \hat{x}\|^2. \qquad (2.26) \end{aligned}$$

The relation (2.26) implies that, for all $k \geq 0$,

$$\|x_{k+1} - \hat{x}\| \leq \|x_k - \hat{x}\| \leq \|x_{k-1} - \hat{x}\| \leq \ldots \leq \|x_0 - \hat{x}\|,$$

where x_0 is one of the given starting points. This means that $\lim_{k\to\infty} \|x_k - \hat{x}\|$ exists and then $\{x_k\}$ is bounded. Since $\alpha_k \in [0,1]$, we have that $\{w_k\}$ is also bounded. Let $b_1 := L\|x_0 - \hat{x}\|$. Since F is Lipschitz continuous, then, for all $k \geq 0$,

$$\|F(x_k)\| = \|F(x_k) - F(\hat{x})\| \leq L\|x_k - \hat{x}\| \leq \cdots \leq L\|x_0 - \hat{x}\| = b_1. \tag{2.27}$$

To show (ii), let $k=0$. From the definition of the search direction d_k in (2.7), we have

$$\|d_0\| = \|F(x_0\| \leq b_1. \tag{2.28}$$

From the Lipschitz continuity of F, we have

$$\|z_{k-1}\| = \|F(w_k) - F(w_{k-1}) + r(w_k - w_{k-1})\| \leq (L+r)\|w_k - w_{k-1}\|. \tag{2.29}$$

Note that

$$\begin{aligned}
\|d_k\| &= \| - \widehat{\theta}_k F(x_k) + \beta_k(w_k - w_{k-1}) - u_{k-1}z_{k-1}\| \\
&\leq \widehat{\theta}_k\|F(x_k)\| + |\beta_k|\|w_k - w_{k-1}\| + |u_{k-1}|\|z_{k-1}\| \\
&= \widehat{\theta}_k\|F(x_k)\| + \left|\frac{\langle z_{k-1},\ F(x_k)\rangle}{\langle z_{k-1},\ w_k - w_{k-1}\rangle}\right|\|w_k - w_{k-1}\| + \left|\frac{\langle w_k - w_{k-1},\ F(x_k)\rangle}{\langle z_{k-1},\ w_k - w_{k-1}\rangle}\right|\|z_{k-1}\| \\
&\leq \widehat{\theta}_k\|F(x_k)\| + \frac{\|z_{k-1}\|\|F(x_k)\|}{\langle z_{k-1},\ w_k - w_{k-1}\rangle}\|w_k - w_{k-1}\| + \frac{\|w_k - w_{k-1}\|\|F(x_k)\|}{\langle z_{k-1},\ w_k - w_{k-1}\rangle}\|z_{k-1}\| \\
&= \widehat{\theta}_k\|F(x_k)\| + 2\frac{\|z_{k-1}\|\|F(x_k)\|}{\langle z_{k-1},\ w_k - w_{k-1}\rangle}\|w_k - w_{k-1}\| \\
&\leq \frac{1}{r}\|F(x_k)\| + 2\frac{(L+r)\|w_k - w_{k-1}\|^2}{r\|w_k - w_{k-1}\|^2}\|F(x_k)\| \\
&= \left[\frac{1}{r} + 2\frac{(L+r)}{r}\right]\|F(x_k)\|.
\end{aligned} \tag{2.30}$$

The first and the second inequality follow from the triangle inequality and the Cauchy-Schwarz inequality, respectively. The third inequality follows from (2.15) and (2.29).

If we let $\widehat{b}_2 := \left[\frac{1}{r} + 2\frac{(L+r)}{r}\right]$, then (2.30) becomes

$$\|d_k\| \leq \widehat{b}_2\|F(x_k)\|. \tag{2.31}$$

Combining (2.31) and (2.28) gives

$$\|d_k\| \leq b, \text{ for all } k \geq 0, \tag{2.32}$$

where $b := b_1\widehat{b}_2$.

(iii). From the definition of υ_k in (2.4), (2.32) and the boundedness of $\{x_k\}$, we have that, for all $k \geq 0$, $\{\upsilon_k\}$ is bounded. Therefore, we can find some constant, say $\overline{b}_2$, such that $\|\upsilon_k - \hat{x}\| \leq \overline{b}_2$. Subsequently, from the Lipschitz continuity of F, there exists some constant, say b_2, such that, for all $k \geq 0$,

$$\|F(\upsilon_k)\| = \|F(\upsilon_k) - F(\hat{x})\| \leq L\|\upsilon_k - \hat{x}\| \leq b_2, \tag{2.33}$$

where $b_2 := L\overline{b}_2$.

(iv). From (2.25), we can deduce that

$$\langle F(\upsilon_k),\ \eta_k d_k\rangle^2 \leq \frac{\|F(\upsilon_k)\|^2}{\gamma(2-\gamma)}(\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2). \tag{2.34}$$

From the definition of η_k in Step 3 of Algorithm 1, and (2.5), we have

$$\sigma^2\eta_k^4\|d_k\|^4\|F(\upsilon_k)\|^{2/c} \leq \langle F(\upsilon_k),\ \eta_k d_k\rangle^2. \tag{2.35}$$

Combining (2.34) and (2.35), we have

$$\sigma^2\eta_k^4\|d_k\|^4\|F(\upsilon_k)\|^{2/c} \leq \frac{\|F(\upsilon_k)\|^2}{\gamma(2-\gamma)}(\|x_k-\hat{x}\|^2-\|x_{k+1}-\hat{x}\|^2). \tag{2.36}$$

Using (2.33) and the fact that $\sigma>0$, $0<\gamma<2$ and $\lim_{k\to\infty}\|x_k-\hat{x}\|$ exists, we obtain from (2.36) that

$$\begin{aligned}\lim_{k\to\infty}\eta_k^4\|d_k\|^4 &\leq \frac{1}{\gamma(2-\gamma)\sigma^2}\lim_{k\to\infty}\|F(\upsilon_k)\|^{2-2/c}(\|x_k-\hat{x}\|^2-\|x_{k+1}-\hat{x}\|^2)\\ &\leq \frac{b_2^{2-2/c}}{\gamma(2-\gamma)\sigma^2}\lim_{k\to\infty}(\|x_k-\hat{x}\|^2-\|x_{k+1}-\hat{x}\|^2)\\ &= 0.\end{aligned}$$

This implies

$$\lim_{k\to\infty}\eta_k\|d_k\| = 0. \tag{2.37}$$

□

Theorem 2.1. *Suppose that Assumption 2.1 holds. For $\kappa>0$, let $\{x_k\}$ be the iterative sequence and let d_k be the search direction generated by Algorithm 1 or 2. Then,*

$$\liminf_{k\to\infty}\|F(x_k)\| = 0. \tag{2.38}$$

Furthermore, $\{x_k\}$ converges to a point $\hat{x}$, which satisfies $F(\hat{x})=0$.

Proof. Suppose that, for all $k>0$, $\|d_k\|\neq 0$. Then, from (2.31), we have

$$\frac{\|F(x_k)\|}{\|d_k\|} \geq \frac{1}{\widehat{b}_2},\ \|d_k\|\neq 0, \tag{2.39}$$

where $\widehat{b}_2 := \left[\frac{1}{r}+2\frac{(L+r)}{r}\right]$, L is a Lipschitz constant and $r>0$ is a positive constant.

Let $\upsilon_k' := x_k+\eta_k' d_k$ and suppose that $\eta_k\neq\kappa,\ \kappa>0$. Then, for $\rho\in(0,1)$, $\eta_k' := \rho^{-1}\eta_k$ does not satisfy (2.5), i.e.,

$$\langle F(\upsilon_k'),\ d_k\rangle + \sigma\eta_k'\|F(\upsilon_k')\|^{1/c}\|d_k\|^2 > 0,\ c>0. \tag{2.40}$$

Under our assumptions, $\hat{x}$ is a solution to problem (1.3). Then, from the Lipschitz continuity of F and the boundedness of $\|d_k\|$ in (2.32) with $b=b_1\widehat{b}_2$, for any given starting point, say x_0, we have

$$\begin{aligned}\|F(\upsilon_k')\| &\leq L\|x_k+\eta_k' d_k-\hat{x}\|\\ &\leq L\|x_k-\hat{x}\|+L\eta_k'\|d_k\|\\ &\leq L\|x_0-\hat{x}\|+L\eta_k'\|d_k\|\\ &\leq b_1+L\rho^{-1}b,\end{aligned}$$

where $b_1 := L\|x_0-\hat{x}\|$. Now, letting $b_3 := b_1+L\rho^{-1}b$, we have, for all $k\geq 0$,

$$\|F(\upsilon_k')\| \leq b_3. \tag{2.41}$$

Using (2.40), we find from inequality (2.1), the Lipschitz continuity of F, and the Cauchy-Schwarz inequality that

$$\begin{aligned} t\|F(x_k)\|^2 &\leq -\langle F(x_k),\, d_k\rangle \\ &< -\langle F(x_k),\, d_k\rangle + \langle F(\upsilon_k'),\, d_k\rangle + \sigma\eta_k'\|F(\upsilon_k')\|^{1/c}\|d_k\|^2 \\ &= \langle F(\upsilon_k') - F(x_k),\, d_k\rangle + \sigma\eta_k'\|F(\upsilon_k')\|^{1/c}\|d_k\|^2 \\ &\leq \|F(\upsilon_k') - F(x_k)\|\|d_k\| + \sigma\eta_k'\|F(\upsilon_k')\|^{1/c}\|d_k\|^2 \\ &\leq L\eta_k'\|d_k\|^2 + \sigma\eta_k'\|F(\upsilon_k')\|^{1/c}\|d_k\|^2 \\ &= \rho^{-1}\eta_k(L+\sigma\|F(\upsilon_k')\|^{1/c})\|d_k\|^2. \end{aligned}$$

This further gives

$$\eta_k \geq \frac{\rho t\|F(x_k)\|^2}{(L+\sigma\|F(\upsilon_k')\|^{1/c})\|d_k\|^2} \geq \frac{\rho t}{(L+\sigma b_3^{1/c})}\frac{1}{\widehat{b}_2^2}, \tag{2.42}$$

where the last inequality follows from (2.39) and (2.41). Combining (2.37) with (2.42), we obtain

$$\liminf_{k\to\infty}\|d_k\| = 0. \tag{2.43}$$

Moreover, we can deduce from (2.1) the following relation

$$t\|F(x_k)\|^2 \leq -F(x_k)^T d_k \leq \|F(x_k)\|\|d_k\|,\ t>0,$$

which further gives

$$\|d_k\| \geq t\|F(x_k)\|,\ t>0. \tag{2.44}$$

Therefore,

$$0 = \liminf_{k\to\infty}\|d_k\| \geq t\liminf_{k\to\infty}\|F(x_k)\|, \tag{2.45}$$

which gives (2.38).

Furthermore, since F is continuous and the sequence $\{x_k\}$ is bounded, then there is some accumulation point of $\{x_k\}$, say $\hat{x}$, for which $\|F(\hat{x})\| = 0$. From the boundedness of $\{x_k\}$, we can find a subsequence $\{x_{k_j}\}$ of $\{x_k\}$ such that $\lim_{k\to\infty}\|x_{k_j} - \hat{x}\| = 0$. Since $\lim_{k\to\infty}\|x_k - \hat{x}\|$ exist (according to Lemma 2.2), we can conclude that $\lim_{k\to\infty}\|x_k - \hat{x}\| = 0$ and the proof is complete. □

3. Numerical Experiments and Comparisons

In this section, we present some numerical experiments to assess the performance of the proposed algorithms (Algorithm 1 and 2) as well as their computational advantages in comparison with some existing methods. We implement these algorithms to solve a collection of monotone systems of nonlinear equations, see Test Problems for Algorithm 1 (Experiment One), i.e. **C**onjugate **G**radient **A**lgorithm with **I**nertial Step (CGAIS) as well as Algorithm 2 (Experiment Two), i.e., **S**pectral **A**lgorithm with **I**nertial **S**tep (SAIS). Finally, we modify and implement Algorithm 1 to solve a motion control problem involving two-joint planar robotic manipulator. All the solvers were coded in MATLAB R2017a software and run on a PC with Intel(R) Core(TM) i7–7500U processor with 8.00 *GB* of RAM and CPU of 2.70 *GHz*.

3.1. **Experiment I.** In this experiment, we compare the performance of our proposed algorithm, i.e., Algorithm 1 (CGAIS) with the same algorithm but without inertial step denoted by CGWOI (**C**onjugate **G**radient **W**ith**O**ut **I**nertial step) and Algorithm 1 of Awwal et al. [25], denoted by HSS on Problems: 3.1, 3.2, 3.3, 3.4, 3.5 and 3.10 of the Test Problems. In other words, we try to compare the numerical performance of the method that incorporate the inertial step with two other methods without the inertial step. In order to have best possible results, the following parameters were chosen for the implementation of Algorithm 1 (CGAIS) as well as CGWOI method: $\sigma = 10^{-4}$, $r = 0.01$, $c = 2$, $\rho = 0.50$, $\gamma = 1.99$, $\kappa = 1$ and $\alpha_k = \frac{1}{(k+1)^2}$. The parameters for the HSS method were chosen as reported in [25]. Furthermore, the iteration process for the Test Problems is terminated whenever the inequality $\|F(x_k)\| < 10^{-6}$ or $\|F(\upsilon_k)\| < 10^{-6}$ is satisfied and failure is declared when the number of iterations exceed 1000 and the terminating criterion mentioned above has not been satisfied.

The Test Problems were taken from the existing literature and the function F is taken as $F(x) = (f_1(x), f_2(x), \ldots, f_n(x))^T$, $f_i : \mathbb{R}^n \to \mathbb{R}$, for $i = 1, 2, \ldots, n$. However, since our proposed algorithm uses two starting points, i.e., x_{-1} and x_0, then, for each x_j, $j = 1, 2, \ldots, 14$, taken from Table 1, we set $x_{-1} := \{x_j^1 + i, x_j^2 + i, \ldots, x_j^n + i\}$, $i \geq 0$, and update them subsequently. All the Test Problems were solved using the dimension (DIM) of $n = 1000,\ 5000,\ 10000,\ 50000$, and 100000.

TABLE 1. The initial points used for Algorithm 1 and 2

Initial Points (INP)	Values
x_1	$(1, 1, 1, \ldots, 1)^T$
x_2	$\left(\frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \ldots, \frac{1}{10}\right)^T$
x_3	$\left(\frac{1}{2}, \frac{1}{2^2}, \frac{1}{2^3}, \ldots, \frac{1}{2^n}\right)^T$
x_4	$\left(1 - \frac{1}{n}, 1 - \frac{2}{n}, 1 - \frac{3}{n}, \ldots, 0\right)^T$
x_5	$\left(0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n-1}{n}\right)^T$,
x_6	$\left(1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{n}\right)^T$
x_7	$\left(\frac{n-1}{n}, \frac{n-2}{n}, \frac{n-3}{n}, \ldots, 0\right)^T$
x_8	$\left(\frac{1}{n}, \frac{2}{n}, \frac{3}{n}, \ldots, 1\right)^T$
x_9	$\text{rand}(0, 1)$
x_{10}	$\left(\frac{3}{2}, \frac{3}{2}, \frac{3}{2}, \ldots, \frac{3}{2}\right)^T$
x_{11}	$(2, 2, 2, \ldots, 2)^T$
x_{12}	$\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}\right)^T$
x_{13}	$5\min(ih,\ 1 - ih),\ 1 \leq i \leq n,\ h = 1/(n+1)$
x_{14}	$\left(\frac{-1}{4}, \frac{2}{5}, \frac{-3}{6}, \ldots, \frac{(-1)^n}{n+3}\right)^T$

Test Problems:

Problem 3.1. [26]: $f_1(x) = e^{x_1} - 1$, $f_i(x) = e^{x_i} + x_{i-1} - 1$, for $i = 1, 2, \ldots, n-1$ and $\Lambda = \mathbb{R}^n_+$.

Problem 3.2. [26]: $f_i(x) = 2x_i - \sin|x_i|$, $i = 1, 2, \ldots, n$ and $\Lambda = \mathbb{R}^n_+$.

Problem 3.3. [27]: $f_i(x) = e^{x_i} - 1$, $i = 1, 2, \ldots, n$ and $\Lambda = \mathbb{R}^n_+$.

Problem 3.4. [28]:

$$f_1(x) = x_1 - e^{\cos(h(x_1+x_2))},$$
$$f_i(x) = x_i - e^{\cos(h(x_{i-1}+x_i+x_{i+1}))},\ i = 2,\dots,n-1,$$
$$f_n(x) = x_n - e^{\cos(h(x_{n-1}+x_n))},\ \text{where } h = \frac{1}{n+1} \text{ and } \Lambda = \mathbb{R}^n_+.$$

Problem 3.5. [6]:

$$f_i(x) = x_i - \sin(|x_i - 1|),\ i = 1,2,\dots,n-1,$$
$$\text{and } \Lambda = \{x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i \le n,\ x_i \ge -1,\ i = 1,2,\dots,n\}.$$

Problem 3.6. [29]: $f_i(x) = (e^{x_i})^2 + \frac{3}{2}\sin(2x_i) - 1$, $i = 1,2,\dots,n$ and $\Lambda = \mathbb{R}^n_+$.

Problem 3.7. [29]:

$$f_1(x) = \frac{5}{2}x_1 + x_2 - 1,$$
$$f_i(x) = x_{i-1} + \frac{5}{2}x_i + x_{i+1} - 1,\ i = 2,\dots,n-1,$$
$$f_n(x) = x_{n-1} + \frac{5}{2}x_n - 1 \text{ and } \Lambda = \mathbb{R}^n_+.$$

Problem 3.8. [29]:

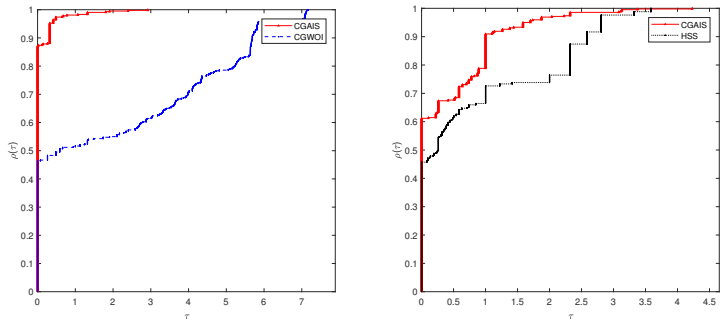
$$f_1(x) = 2x_1 - x_2 + e^{x_1} - 1,$$
$$f_i(x) = -x_{i-1} + 2x_i - x_{i+1} + e^{x_i} - 1,\ i = 2,\dots,n-1,$$
$$f_n(x) = -x_{n-1} + 2x_n + e^{x_n} - 1 \text{ and } \Lambda = \mathbb{R}^n_+.$$

Problem 3.9. [30]:

$$f_1(x) = x_1 + \sin x_1 - 1,$$
$$f_i(x) = -x_{i-1} + 2x_i + \sin x_i - 1,\ i = 2,\dots,n-1,$$
$$f_n(x) = x_n + \sin x_n - 1 \text{ and } \Lambda \in [-3,+\infty].$$

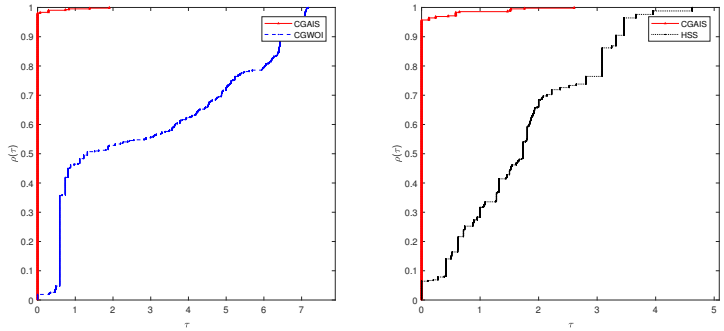
Problem 3.10. [29]: $f_i(x) = \frac{i}{n}e^{x_i} - 1$, for $i = 1,2,\dots,n$ and $\Lambda = \mathbb{R}^n_+$.

The metrics used for the comparison are ITER (number of iterations), FVAL (number of function evaluations) and TIME (CPU time in seconds) where the information together with their NORM (norm of the objective functions at the solutions) is reported. The table of the numerical results is available in the following link https://github.com/AMBakoji/CGAIS-CGWOIS. The NORM values reported indicate that each solver successfully obtained solutions of virtually all the test problems with least ITER and FVAL. These information are summarized in Figures 1 and 2 based on the Dolan and Moré performance profile [31]. Figure 1 (A) compares the performance between CGAIS and CGWOI methods based on ITER while Figure 1 (B) illustrates the Dolan and Moré performance profile of CGAIS and HSS methods based on ITER, respectively. Figure 2 (A) and (B) demonstrate the performance profile based on FVAL of CGAIS with CGWOI and FVAL of CGAIS with HSS methods, respectively. We see from Figure 1 ((A) and (B)) and Figure 2 ((A) and (B)) that CGAIS solver performs better with higher percentage win of



(A) (B)

FIGURE 1. Performance profiles based on number of iterations (ITER)



(A) (B)

FIGURE 2. Performance profiles based on number of function evaluations (FVAL)

ITER and FVAL than CGWOI and HSS solvers for solving all the four hundred and twenty (420) Test Problems. In other words, this experiment reveals that CGAIS solver has advantage over CGWOI and HSS solvers with regards to ITER and FVAL. Thus, we may conclude that the inertial step incorporated in Algorithm 1 has impacted positively on improving the numerical performance of the proposed algorithm.

On the other hand, in order to see the cost expensive optimization of the solvers for the Test Problems with expensive function evaluations, we employ Data Profile proposed by Moré and Wild [32]. In this case the performance measure is the number of function evaluations (FVAL) because this is assumed to be the dominant cost per iteration. Performance profiles provide an accurate view of the relative performance of solvers within a given number of function evaluations, (see Figure 2). Performance profiles do not, however, provide sufficient information for a user with an expensive optimization problem.

Users with expensive optimization problems are often interested in the performance of solvers as a function of the number of functions evaluations. In other words, these users are interested in the percentage of problems that can be solved (for a given tolerance τ) with say μ function evaluations. Performance profiles and data profiles are cumulative distribution functions, and thus monotone increasing, step functions with a range in $[0,1]$. However, performance profiles

compare different solvers while data profiles display the raw data as it is. In particular, performance profiles do not provide the number of function evaluations required to solve any of the problems. Also note that data profile for a given solver say $s \in S$ is independent of other solvers; this is not the case for performance profiles [32].

Data profiles are useful to users with a specific computational budget who need to choose a solver that is likely to reach a given reduction in function value. The user needs to express the computational budget in terms of simplex gradients and examine the values of the data profile for all the solvers. For example, if the user has a budget of 20 simplex gradients (FVAL), then the data profiles in Figure 3 (top-left, top-right and bottom-left) show that the proposed CGAIS solver solves about 99% of the problems at this level of accuracy while solver CGWOI solves about 50% of the problems and HSS solver solves about 63% of the entire problems. This means that CGWOI and HSS methods are more computationally expensive compared to the proposed CGAIS method in order to successfully solve certain percentage of the entire four hundred and twenty (420) Test Problems we have considered. However, this information is not available from the performance profile in Figure 2. This interpretation means that data profile measures the reliability of the solver for a given tolerance as a function of the budget.

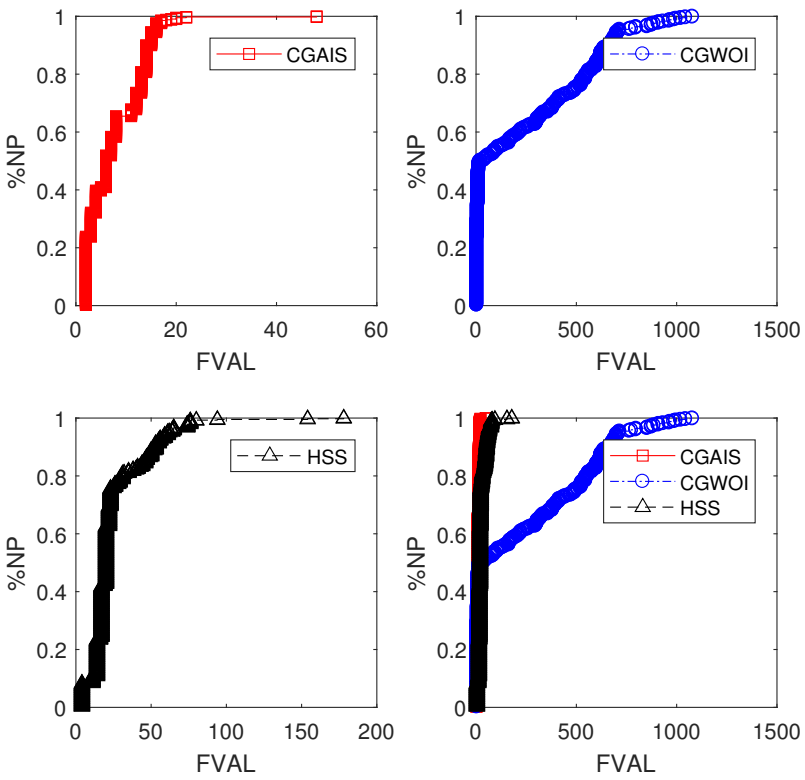


FIGURE 3. Percentage Data Profile of CGAIS, CGWOI and HSS methods

3.2. **Experiment II.** Similar to Experiment I, we compare the performance of Algorithm 2 (SAIS) with Algorithm 1 and 2 of Awwal et al. [14] denoted by DAIS 1 and DAIS 2. All the three solvers are spectral gradients in nature and each of the solver has an inertial step incorporated in it, which means that the three solvers are similar in nature. So, the aim of this

experiment is to compare their numerical performance. Here, we solved Problems: 3.1, 3.2, 3.6, 3.7, 3.8 and 3.10 of the Test Problems for the three solvers. The following parameters were chosen for Algorithm 2 (SAIS): $\sigma = 10^{-4}$, $r = 0.01$, $c = 2$, $\gamma = 1.99$, $\rho = 0.45$, $\kappa = 1$ and $\alpha_k = \frac{1}{(k+1)^2}$ while the parameters for DAIS 1 and DAIS 2 were chosen as reported in [14]. A total of four hundred and twenty (420) Test Problems were solved using the same initial points and dimensions as in experiment one. A detail of the table of numerical results are reported and can be found via the link https://github.com/AMBakoji/CGAIS-CGWOIS.

In order to have a graphical view on the numerical performance of SAIS solver relative to DAIS 1 and DAIS 2 solvers with respect to the metrics ITER and FVAL, we employ the Dolan and Moré performance profile. Figure 4 illustrates the performance profile of the three solvers based on ITER while Figure 5 demonstrates the performance profile based on FVAL. From Figures 4 and 5, we can see that SAIS solver recorded least ITER and FVAL with about 90% win of the entire experiments. This clearly shows that our proposed algorithm (SAIS) outperformed DAIS 1 and DAIS 2 algorithms, respectively. In other words, we can say that the experiment reveals that SAIS solver has advantage over DAIS 1 and DAIS 2 solvers with regard to the metrics ITER and FVAL.

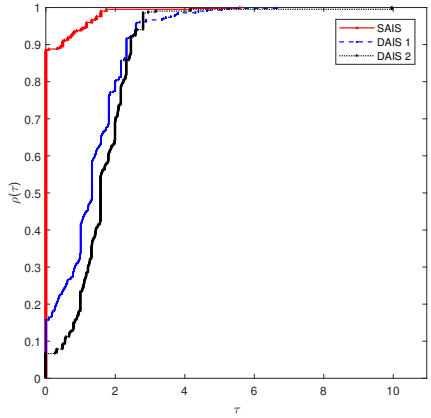


FIGURE 4. Performance profiles based on number of iterations

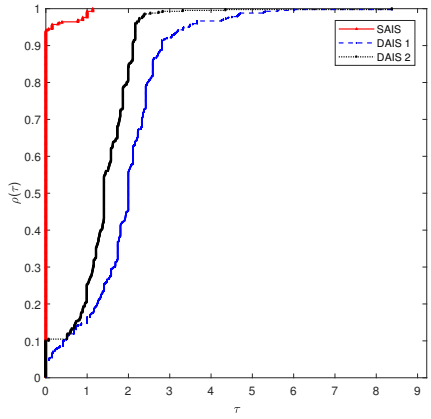


FIGURE 5. Performance profiles based on number of function evaluations

3.3. **The motion control model.** Problems arising in the concept of a robot system is now under spotlight and a number of methods for dealing with them were proposed [33]. Due to the wide range of applications of *n-link*-robots, Zhang et al. [34] discussed one of its fundamentals, known as the *1-link* robot system. According to [35], the characteristics of a motor dynamics should be taken into consideration for a robot movement to satisfy the stability and accuracy requirements. For the requirements, a motor dynamics needs to satisfy is that the actual output of the system should track the desired output within an acceptable minimal error [36]. Recently, a number of methods were raised to deal with the tracking control problems of a nonlinear system. Here, we can mention the proportional-integral-derivative (PID) control [37, 38], the feed back linearization [39, 40] and the optimal output tracking control using the approximation approach [41]. Furthermore, some motion control models can be formulated as a planar location problem where the distances are measured by certain norms related to the configuration of the robot that motion should be controlled. These location problems can be considered as special approximation problems. locational analysis gives effective methods for solving these problems (see [42]).

In this experiment, we consider a motion control problem involving a two-joint planar robotic manipulator. We modify Algorithm 1 to be suitable for solving the problems of the nature $\min\{f(x) : x \in \mathbb{R}\}$, where $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be a continuously differentiable and convex function. The modified algorithm is implemented in solving the following motion control model.

Algorithm 3: Modified CGAIS (MCGAIS)

Input: Give the same inputs as in Algorithm 1 with $\alpha_k = 0$, for all $k \geq 0$. Let $F(x_k) = \nabla f(x_k)$.

Replace *Step 2* and *Step 3* of Algorithm 1 with the following:

Step 3: Compute the step size $\eta_k := \kappa\rho^i$ with i being the smallest non-negative integer for which

$$f(x_k + \kappa\rho^i d_k) - f(x_k) \leq \sigma\kappa\rho^i F(x_k)^T d_k. \tag{3.1}$$

Step 4: Update the next iterate using the following

$$x_{k+1} := x_k + \eta_k d_k. \tag{3.2}$$

Consider the problem $\min\{f(x) : x \in \mathbb{R}\}$, where $f : \mathbb{R}^n \to \mathbb{R}$ has a solution and the level set $\{x : f(x) \leq f(x_0)\}$ is bounded. Since $F(x)$ is assumed to be Lipschitz continuous on $\mathbb{R}^n$, we obtain from *Theorem* 2 of [43] that $\liminf_{k\to\infty} \|F(x_k)\| = 0$.

As described in [44], the discrete-time kinematics equation of a two-joint planar robot manipulator at the position level is given as

$$Q(\theta_k) := q_k. \tag{3.3}$$

The vectors $\theta_k \in \mathbb{R}^2$ and $q_k \in \mathbb{R}^2$ represent the joint angle vector and the end effector position vector, respectively. The function $Q(\cdot)$ is the kinematics mapping with the following known structure

$$Q(\theta_k) = \begin{bmatrix} \ell_1 c_1 + \ell_2 c_2 \\ \ell_1 s_1 + \ell_2 s_2 \end{bmatrix}, \tag{3.4}$$

where ℓ_i $(i = 1,2)$ is the length of the i^{th} rod, $c_1 = \cos(\theta_1)$, $c_2 = \cos(\theta_1 + \theta_2)$, $s_1 = \sin(\theta_1)$ and $s_2 = \sin(\theta_1 + \theta_2)$. In view of robotic control, we need to solve the following minimization problem

$$\min_{q_k \in \mathbb{R}^2} f(q_k), \text{ where } f(q_k) = \frac{1}{2}\|q_k - q_{dk}\|^2, \tag{3.5}$$

q_{dk} is the end effector control track. Consider computational time intervals with $t_k \in [0, t_f]$, where t_f is the end of task duration.

Following the approach in [43, 45], we take the length of the rod $\ell_i = 1$, $(i = 1,2)$ and the end effector is controlled to track the following two Lissajous curves expressed by

$$q_{dk}^{(1)} = \begin{bmatrix} \frac{3}{2} + \frac{1}{5}\sin(3t_k) \\ \frac{\sqrt{3}}{2} + \frac{1}{5}\sin(2t_k) \end{bmatrix}, \tag{3.6}$$

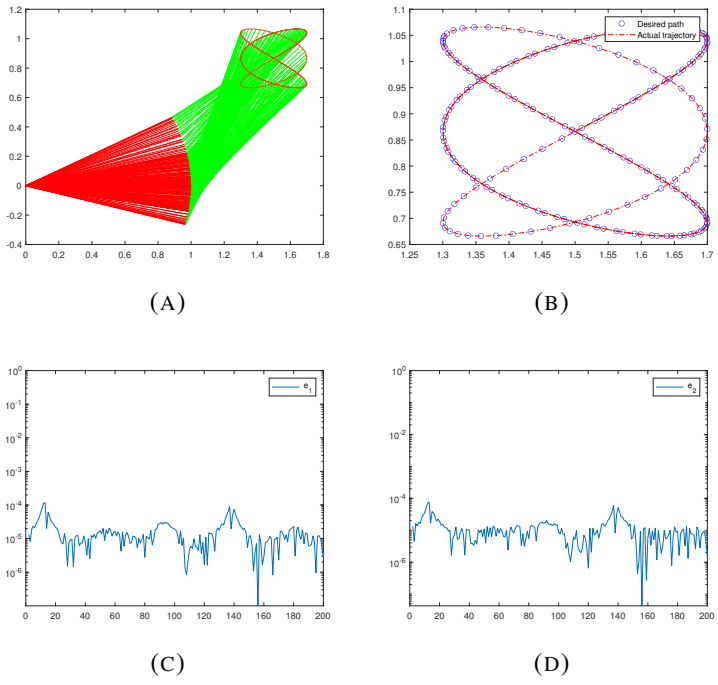
and

$$q_{dk}^{(2)} = \begin{bmatrix} \frac{3}{2} + \frac{1}{5}\sin(t_k) \\ \frac{\sqrt{3}}{2} + \frac{1}{5}\sin(2t_k) \end{bmatrix}. \tag{3.7}$$

In Algorithm 3 (MCGAIS), we set the parameters $\rho = 0.6$, $\sigma = 0.08$ and the task duration $t_f = 10$ seconds. For the initial point, we choose $\theta_0 = [0, \frac{\pi}{3}]^T$ and divide the task duration $t = [0, 10]$ into 200 equal parts. The numerical results generated by Algorithm 3 are depicted in Figures 6 and 7 for the Lissajous curves given in $q_{dk}^{(1)}$ and $q_{dk}^{(2)}$, respectively. The Figures 6(A), 6(B), 7(A) and 7(B) show that Algorithm 3 (MCGAIS) completes the task at hand successfully for $q_{dk}^{(1)}$ and $q_{dk}^{(2)}$, respectively (compare Figures 7 and 8 in [45]). The Figures 6(C) and 6(D) present the residual errors $\varepsilon(t_{k+1})$ along the x and y axes, respectively for $q_{dk}^{(1)}$ while Figures 7(C) and 7(D) present the residual errors along the x and y axes for $q_{dk}^{(2)}$. Looking at the residual errors on both x and y axes of Figures 6(C), 6(D), 7(C) and 7(D), we can say that Algorithm 3 recorded an error of about 10^{-5}, which is acceptable. Hence, from the two figures, we can say that Algorithm 3 can be implemented to handle successfully motion control models, which are real world problems.

4. The Conclusion

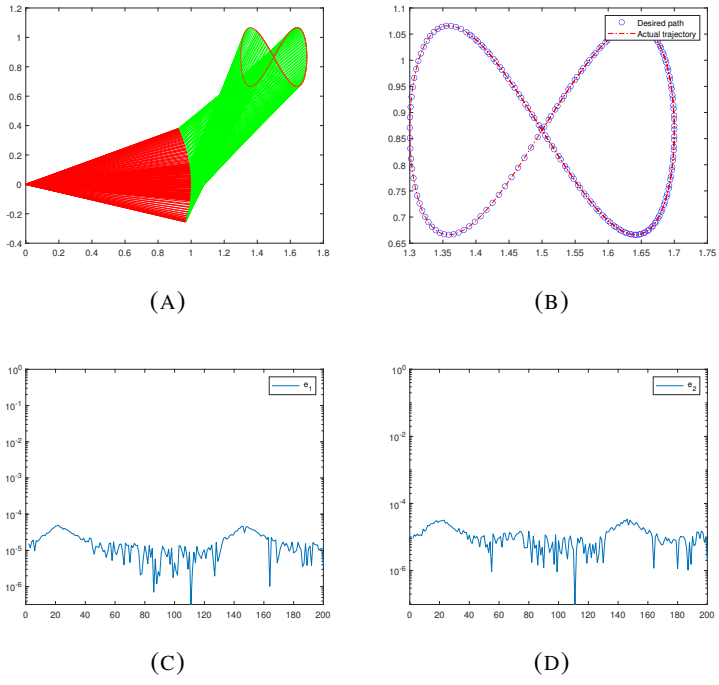
In this paper, based on the projection method considered by Solodov and Svaiter [3], we proposed two derivative-free algorithms with inertial effects for solving problem (1.3). We showed that the proposed search directions are well-defined and satisfy the sufficient descent condition, which is independent on the line search strategy. Under the assumptions that the underlying function is monotone and Lipschitz continuous, the global convergence of the proposed algorithms were established. Preliminary numerical experiments showed that the proposed algorithms were capable of solving convex constrained nonlinear monotone operator equations with acceptable efficiency. The good numerical performance recorded by Algorithms 1 and 2 may be attributed by the inertial effect incorporated in our algorithms. Finally, we modified Algorithm 1 and solved a real world problem involving a motion control problem for a two-joint planar robotic manipulator system.



(A) (B)

(C) (D)

FIGURE 6. Robot trajectories path and residual errors $\varepsilon(t_{k+1})$ along x and y axes of motion control model for the Lissajous curve $q_{dk}^{(1)}$ of Algorithm 3 (MCGAIS)



(A) (B)

(C) (D)

FIGURE 7. Robot trajectories path and residual errors $\varepsilon(t_{k+1})$ along x and y axes of motion control model for the Lissajous curve $q_{dk}^{(2)}$ of Algorithm 3 (MCGAIS)

Acknowledgments

The authors are very grateful to the anonymous reviewers for their suggestions and comments that improved the presentation of this paper. The first author, Abubakar Bakoji Muhammad, was supported by the DAAD (Germany) and PTDF (Nigeria) scholarship at the Martin-Luther-University Halle-Wittenberg, 06099 Halle (Saale), Germany.

REFERENCES

[1] Y. Dai, Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM J. Optim. 10 (1999), 177-182.
[2] J. Liu, Y. Feng, L. Zou, Some three-term conjugate gradient methods with the inexact line search condition, Calcolo, 55 (2018), 16.
[3] M. V. Solodov, B. F. Svaiter, A globally convergent inexact Newton method for systems of monotone equations, in Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, pp. 355-369, Springer, 1998.
[4] W. Cheng, A PRP type method for systems of monotone equations, Math. Comput. Model. 50 (2009), 15-20.
[5] C. W. Wang, Y. J. Wang, A superlinearly convergent projection method for constrained systems of nonlinear equations, J. Glob. Optim. 44 (2009), 283-296.
[6] Y. Xiao, H. Zhu, A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing, J. Math. Anal. Appl. 405 (2013), 310-319.
[7] J. M. Ortega, W. C. Rheinboldt, Iterative solution of nonlinear equations in several variables, SIAM, 2000.
[8] C. L. Chan, A. K. Katsaggelos, A. V. Sahakian, Image sequence filtering in quantum-limited noise with applications to low-dose fluoroscopy, IEEE Trans. Medical Imaging 12 (1993), 610-621.
[9] S.Y. Cho, A convergence theorem for generalized mixed equilibrium problems and multivalued asymptotically nonexpansive mappings, J. Nonlinear Convex Anal. 21 (2020), 1017-1026.
[10] N. A. Iusem, V. M. Solodov, Newton-type methods with generalized distances for constrained optimization, Optimization, 41 (1997), 257-278.
[11] Y. Xiao, Q. Wang, Q. Hu, Non-smooth equations based method for ℓ_1-norm problems with applications to compressed sensing, Nonlinear Anal. 74 (2011), 3570-3577.
[12] B. T. Polyak, Some methods of speeding up the convergence of iteration methods, USSR Comput. Math. Math. Phys. 4 (1964), 1-17.
[13] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (2009), 183-202.
[14] A. M. Awwal, P. Kumam, L. Wang, S. Huang, W. Kumam, Inertial-based derivative-free method for system of monotone nonlinear equations and application, IEEE Access, 8 (2020), 226921-226930.
[15] X. Qin, L. Wang, J.C. Yao, Inertial splitting method for maximal monotone mappings, J. Nonlinear Convex Anal. 21 (2020), 2325-2333.
[16] Y. Shehu, J.C. Yao, Rate of convergence for inertial iterative method for countable family of certain quasi-nonexpansive mappings, J. Nonlinear Convex Anal. 21 (2020), 533-541.
[17] L. Liu, B. Tan, S.Y. Cho, On the resolution of variational inequality problems with a double-hierarchical structure, J. Nonlinear Convex Anal. 21 (2020), 377-386.
[18] N. T. Vinh, L. D. Muu, Inertial extragradient algorithms for solving equilibrium problems, Acta Math. Vietnamica 44 (2019), 639-663.
[19] B. Tan, S.Y. Cho, Strong convergence of inertial forward–backward methods for solving monotone inclusions, Appl. Anal. (2021), 10.1080/00036811.2021.1892080.
[20] A. B. Abubakar, A. H. Ibrahim, A. B. Muhammad, C. Tammer, A modified descent Dai-Yuan conjugate gradient method for constraint nonlinear monotone operator equations, Appl. Anal. Optim. 4 (2020), 1-24.
[21] W. W. Hager, H. Zhang, A survey of nonlinear conjugate gradient methods, Pacific J. Optim. 2 (2006), 35-58.
[22] M. Al-Baali, Descent property and global convergence of the fletcher-reeves method with inexact line search, IMA J. Numer. Anal. 5 (1985), 121-124.

[23] E. Pola, G. Ribiere, Note sur la convergence de methodes de directions conjugées, Rev Française Informat Recherche Operationelle, 3e Année, 16 (1969), 35-43.
[24] M.J.D. Powell, Nonconvex minimization calculations and the conjugate gradient method. In: Griffiths D.F. (eds) Numerical Analysis. Lecture Notes in Mathematics, vol 1066. Springer, Berlin, Heidelberg, 1984.
[25] A. M. Awwal, L. Wang, P. Kumam, H. Mohammad, W. Watthayu, A projection Hestenes–Stiefel method with spectral parameter for nonlinear monotone equations and signal processing, Math. Comput. Appl. 25 (2020), 1-29.
[26] A. M. Awwal, P. Kumam, A. B. Abubakar, A modified conjugate gradient method for monotone nonlinear equations with convex constraints, Appl. Numer. Math. 145 (2019), 507-520.
[27] A.M. Awwal, L. Wang, P. Kumam, H. Mohammad, A two-step spectral gradient projection method for system of nonlinear monotone equations and image deblurring problems, Symmetry, 12 (2020), 874.
[28] Y. Bing, G. Lin, An efficient implementation of merrills method for sparse or partially separable systems of nonlinear equations, SIAM J. Optim. 1 (1991), 206–221.
[29] A.M. Awwal, P. Kumam, H. Mohammad, W. Watthayu, A.B. Abubakar, A Perry-type derivative-free algorithm for solving nonlinear system of equations and minimizing ℓ_1 regularized problem, Optimization, 70 (2021), 1231-1259.
[30] S. Aji, P. Kumam, A.M. Awwal, M.M. Yahaya, W. Kumam, Two hybrid spectral methods with inertial effect for solving system of nonlinear monotone equations with application in robotics, IEEE Access, 9 (2021), 30918-30928.
[31] E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2002), 201-213.
[32] J. J. Moré, S. M. Wild, Benchmarking derivative-free optimization algorithms, SIAM J. Optim. 20 (2009), 172-191.
[33] A. Miele, T. Wang, S. Mancuso, Optimization of missions to mars for robotic and manned spacecraft, Nonlinear Anal. 47 (2001), 1425-1443.
[34] Y. Zhang, W. Li, B. Qiu, Y. Ding, D. Zhang, Three-state space reformulation and control of md-included one-link robot system using direct-derivative and zhang-dynamics methods, in 2017 29th Chinese Control And Decision Conference (CCDC), pp. 3724-3729, IEEE, 2017.
[35] Y. Qiang, F. Jing, J. Zeng, Z. Hou, Dynamic modeling and vibration mode analysis for an industrial robot with rigid links and flexible joints, in 2012 24th Chinese Control and Decision Conference (CCDC), pp. 3317-3321, IEEE, 2012.
[36] G.Y. Tang, L. Sun, C. Li, M.Q. Fan, Successive approximation procedure of optimal tracking control for nonlinear similar composite systems, Nonlinear Anal. 70 (2009), 631-641.
[37] E. M. Jafarov, M. A. Parlakci, Y. Istefanopulos, A new variable structure pid-controller design for robot manipulators, IEEE Trans. Control Syst. Tech. 13 (2004), 122-130.
[38] V. Parra-Vega, S. Arimoto, Y.-H. Liu, G. Hirzinger, P. Akella, Dynamic sliding pid control for tracking of robot manipulators: Theory and experiments, IEEE Trans. Robotics Auto. 19 (2003), 967-976.
[39] J. Na, X. Ren, D. Zheng, Adaptive control for nonlinear pure-feedback systems with high-order sliding mode observer, IEEE Trans. Neural Networks Learn. Sys. 24 (2013), 370-382.
[40] B.-S. Chen, H.-J. Uang, C.-S. Tseng, Robust tracking enhancement of robot systems including motor dynamics: A fuzzy-based dynamic game approach, IEEE Trans. Fuzzy Syst. 6 (1998), 538-552.
[41] G.Y. Tang, Y.D. Zhao, B.L. Zhang, Optimal output tracking control for nonlinear systems via successive approximation approach, Nonlinear Anal. 66 (2007), 1365-1377.
[42] H. Hamacher, Mathematische Lösungsverfahren für planare Standortprobleme, Vieweg+Teubner Verlag, Wiesbaden, 1995.
[43] M. Sun, J. Liu, Y. Wang, Two improved conjugate gradient methods with application in compressive sensing and motion control, Math. Probl. Eng. 2020 (2020), 1-11.
[44] Y. Zhang, L. He, C. Hu, J. Guo, J. Li, Y. Shi, General four-step discrete-time zeroing and derivative dynamics applied to time-varying nonlinear optimization, J. Comput. Appl. Math. 347 (2019), 314-329.
[45] M. M. Yahaya, P. Kumam, A.M. Awwal, S. Aji, A structured quasi-Newton algorithm with nonmonotone search strategy for structured NLS problems and its application in robotic motion control, J. Comput. Appl. Math. 395 (2021), 113582.