BANGMOD INTERNATIONAL JOURNAL OF MATHEMATICAL & COMPUTATIONAL SCIENCE
ISSN: 2408-154X (Print)
ISSN: 3057-0557 (Online)
https://bangmodjmcs.com
Volume 12 (2026)
Pages 21–42



A New Class of Hybrid Spectral Conjugate Gradient Approach for Unconstrained Optimization and Motion Control Problems

Abba Sulaiman [1], **Auwal Bala Abubakar**[1,2,3,*], **Muhammad Abdullahi**[4,5], **Sulaiman Aliyu Yayangida** [6], **Sadiq Bashir Masu** [5],

[1] *Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University Kano. Kano State, Nigeria*
E-mails: absulaiman29@gmail.com
[2] *Department of Mathematics and Applied Mathematics, Sefako Makgatho Health Sciences University, Ga-Rankuwa, Pretoria, Medunsa-0204, South Africa*
[3] *Faculty of Education and Arts, Sohar University, Sohar 311, Oman*
E-mails: ababubakar.mth@buk.edu.ng
[4] *School of Mathematics and Statistics, HNP-LAMA, Central South University, Changsha, Hunan 410083, China*
E-mails: 4.muhammad.abdullahi@gmail.com
[5] *Department of Mathematics, Sule Lamido University, Kafin Hausa, P.M.B. 048, Nigeria*
E-mails: sadeeqmasu@gmail.com
[6] *Department of Mathematics and Computer Science, Federal University Kashere, PMB 0182, Gombe, Nigeria E-mails: yayangida50@gmail.com*

**Corresponding author.*

Received: 14 July 2025 / Accepted: 13 January 2026

Abstract This article presents an efficient hybrid Conjugate gradient (CG) method for solving largescale unconstrained optimization problems. The search direction of the proposed method of four distinct classical CG parameters, which consist of descent (DY), Fletcher-Reeves (FR), Hestenes-Stiefel (HS), and Polak-Ribiére-Polyak (PRP) parameters. The welcoming advantage of the new method is that the direction can be descent in any case of the CG parameter and the line search type being used. We prove the global convergence by employing a modified version of Wolfe line search conditions. Numerical experiments on some benchmark test problems were provided to ascertain the efficiency of the method. Lastly, the method was applied to solve problems arising from the motion control of the robot manipulator.

MSC: 65K10, 90C52, 90C26
Keywords: Hybrid three-term conjugate gradient method; Motion control problem; Global convergence; Large-scale problem

Published online: 31 January 2026
© 2026 By TaCS-CoE, All rights reserve.

Published by Center of Excellence in Theoretical and Computational Science (TaCS-CoE)

Please cite this article as: A. Sulaiman et al., A New Class of Hybrid Spectral Conjugate Gradient Approach for Unconstrained Optimization and Motion Control Problems, Bangmod Int. J. Math. & Comp. Sci., Vol. 12 (2026), 21–42. https://doi.org/10.58715/bangmodjmcs.2026.12.2

1. Introduction

Over a significant period, various iterative algorithms have been introduced, such as the Quasi-Newton methods, Gauss-Newton methods, Trust-region methods, and Levenberg-Marquardt methods [35–39]. Nevertheless, these methods struggle with large-scale problems because they necessitate the computation of the Jacobian matrix or its approximations at each iteration. Consequently, Conjugate Gradient Methods (CGM) have emerged as an alternative technique for addressing large-scale nonlinear equations. Notable features of CGM include its simplicity, rapid convergence rate, and the omission of the Hessian matrix, see ([40–43, 45, 46, 56]). Most importantly, CGM finds valuable applications across various scientific and engineering disciplines, for more detail see([19, 20, 22–25, 55]). This has led scholars to regard CGM as a forefront method for solving unconstrained optimization problems defined as follows:

$$\min_{x\in\mathbb{R}^n} \eta(x), \tag{1.1}$$

where $\eta : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, bounded below and its gradient $q(x) = \nabla\eta(x)$ is Lipschitz continuous. Through this study,the symbol $q_k = q(x_k)$. Starting with an initial point $x_0 \in \mathbb{R}^n$, the general sequence of iterations $\{x_k\}$ is defined as follows

$$x_{k+1} = x_k + s_k,\ \ s_k = \alpha_k d_k,\ \ k = 0, 1, 2, \ldots, \tag{1.2}$$

where α_k is the called step length determined using modified wolf-type line search criteria defined as follows

$$\eta(x_k + \alpha_k d_k) - \eta(x_k) \leq \phi\alpha_k q_k^T d_k - \sigma\alpha_k^2\|d_k\|^2, \tag{1.3}$$

$$q(x_k + \alpha_k d_k)^T d_k \geq \rho q_k^T d_k, \tag{1.4}$$

where $0 < \sigma < \phi < \rho < 1$, and d_k is the search direction defined as follows

$$d_0 = -q_0,\ \ d_k = -q_k + \beta_k d_{k-1},\ \ k > 0,$$

where β_k is a CG parameter that differentiates between different CG methods in terms of efficiency and global convergence. A direction is sufficiently descent if it satisfies the inequality below:

$$q_k^T d_k \leq -c\|q_k\|^2, \quad c > 0. \tag{1.5}$$

Several scholars have presented CG parameters that combine or extend existing parameters. Some of the classical parameters are Fletcher-Reeves (FR)[53], Polak-Ribiére-Polyak (PRP)[54], Hestenes Stiefel (HS)[55] and Dai-Yuan (DY)[4]. The following are the formulas for the parameters mentioned earlier:

$$\beta_k^{FR} = \frac{\|q_k\|^2}{\|q_{k-1}\|^2}, \quad \beta_k^{DY} = \frac{\|q_k\|^2}{d_{k-1}^T y_{k-1}}, \tag{1.6}$$

$$\beta_k^{HS} = \frac{q_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{PRP} = \frac{q_k^T y_{k-1}}{\|q_{k-1}\|^2}, \tag{1.7}$$

in this context $q_k = q(x_k)$, $y_{k-1} = q_k - q_{k-1}$ and $\|.\|$ is the Euclidean norm. To present a more efficient and reliable CG parameter, scholars aim to hybridize one or two existing parameters to propose a more effective one. For instance, Jiang et al. [48] proposed new hybrid three-term CG methods with an application in image restoration. The CG

parameter in the search direction is the hybrid of β^{DY} and any other CG parameter. The search direction is descent and bounded irrespective of the line search and choice of the CG parameter. Liu et al. [8] also proposed three-term CG methods with a new direction structure. Some new CG methods are extended in work, and new three-term CG methods are constructed. An essential property of the proposed methods is that the search direction is descent without the line search technique. The methods globally converged under the standard Wolfe line search.

In a recent development, Abubakar et al. [49] proposed a new three-term hybrid spectral gradient method for finding solutions to monotone operator equations, along with applications. The spectral parameter is selected to ensure the direction meets the descent property. The numerical results indicate that the proposed method is more efficient than the algorithms being compared. Moreover, Abubakar et al. [6] formulated a Liu-Storey-type CG method for solving (1.1) with applications in motion control. The proposed method demonstrates global convergence under standard Wolfe and Armijo-like line search conditions. For more information on conjugate gradient methods used in solving unconstrained optimization problems, interested readers can refer to [1, 9–16] and the references cited therein.

Inspired by the work of Abubakar et al. [49], we proposed a new hybrid of the three-term CG method for solving large-scale unconstrained optimization problems with application in motion control. The proposed parameter is the hybrid of the classical CG parameters β^{DY}, β^{FR}, β^{HS}, and β^{PRP}. The proposed search direction satisfies the descent condition without any line search. The proposed method also satisfies global convergence under the modified Wolfe line search condition. Finally, the proposed method was applied to solve the motion control problem, showcasing the applicability of the proposed method to the engineering field.

The structure of the paper is as follows: Section 2 presents the algorithm and theoretical outcomes of the proposed NH method. In Section 3, we discuss and analyze the numerical results along with the application of the method in motion control. Finally, Section 4 concludes the work.

2. Description of the Algorithm

In this section, we presented the search direction of the algorithm alongside the proposed hybrid parameter. The search direction is defined as

$$d_k := \begin{cases} -q_0, & k = 0; \\ -q_k + \beta_k^{NH} s_{k-1} - \lambda_k^{NH} q_k, & k \geq 1, \end{cases} \tag{2.1}$$

where β_k^{NH} is the hybrid of four distinct CG parameters defined as follows

$$\beta_k^{NH} = \frac{r_1 q_k^T y_{k-1} + r_2 \|q_k\|^2}{r_3 \|q_{k-1}\|^2 + r_4 d_{k-1}^T y_{k-1}}, \tag{2.2}$$

where $y_{k-1} = q_k - q_{k-1}$, r_1, r_2, r_3, $r_4 > 0$. Observe that
when $r_1 = r_3 = 1$ and $r_2 = r_4 = 0$, then $\beta_k^{NH} = \beta_k^{PRP}$.
when $r_1 = r_3 = 0$ and $r_2 = r_4 = 1$, then $\beta_k^{NH} = \beta_k^{DY}$.
when $r_1 = r_4 = 1$ and $r_2 = r_3 = 0$, then $\beta_k^{NH} = \beta_k^{HS}$.
when $r_2 = r_3 = 1$ and $r_1 = r_4 = 0$, then $\beta_k^{NH} = \beta_k^{FR}$.

Remark 2.1. It is clear to see that when $k = 0$, the search direction d_k defined in (2.1) automatically satisfies the descent condition. However, for $k \geq 1$, we choose the parameter λ_k^{NH} in (2.1) as follows

$$\lambda_k^{NH} = \beta_k^{NH} \frac{q_k^T s_{k-1}}{\|q_k\|^2}. \tag{2.3}$$

We finally present the flow framework of the proposed algorithm defined below.

Algorithm 1: New Hybrid Three-term Conjugate Gradient Method (`NH`)

Input : Given $x_0 \in \mathbb{R}^n$, $\epsilon > 0$, $0 < \sigma < \phi < \rho < 1$, and $r_1,\ r_2,\ r_3,\ r_4 > 0$.

Step 1 : **if** $\|q_k\| \leq \epsilon$, **then**
stop.
end
Step 2 : **if** $k = 0$, **then**
set $d_k^{NH} := -q_k$;
else
Compute d_k^{NH} using equation (2.1).
end
Step 3 : Determine the step length α_k satisfying the modified Wolfe conditions (1.3)-(1.4).
Step 4 : Compute the next iterate via (1.2).
Step 5 : Determine $k := k + 1$ and go to Step 1.

Lemma 2.2. *Suppose that the search direction d_k is defined by **NH Algorithm**, then it satisfy the following condition*

$$q_k^T d_k = -\|q_k\|^2. \tag{2.4}$$

Proof. For $k = 0$ from (2.1), we have $q_0^T d_0 = -\|q_0\|^2$ which implies (2.4). Now for $k \geq 1$, we assume that $q_k \neq 0$, then we multiply both sides of (2.1) with q_k^T, we obtain

$$\begin{aligned} q_k^T d_k &= -\|q_k\|^2 + \beta_k^{NH} q_k^T s_{k-1} - \lambda_k^{NH} \|q_k\|^2 \\ &= -\|q_k\|^2 + \beta_k^{NH} \alpha_{k-1} q_k^T d_{k-1} - \beta_k^{NH} \frac{\alpha_{k-1} q_k^T d_{k-1}}{\|q_k\|^2} \|q_k\|^2 \\ &= -\|q_k\|^2 + \beta_k^{NH} \alpha_{k-1} q_k^T d_{k-1} - \beta_k^{NH} \alpha_{k-1} q_k^T d_{k-1} \\ &= -\|q_k\|^2. \end{aligned} \tag{2.5}$$

∎

Before demonstrating the global convergence of our proposed method, we carefully consider the following assumptions

Assumption 2.3. The level set $\mathcal{H} = \{x : \eta(x) \leq \eta(x_0)\}$ at x_0 is bounded.

Assumption 2.4. Assume that the function η is continuously differentiable, $\eta(x)$ is bounded below and that its gradient is Lipschitz continuous on $\mathcal{H}$. In other words, there exists a constant $L > 0$ such that for every x

$$\|q(x) - q(\bar{x})\| \leq L\|x - \bar{x}\|, \bar{x} \in \mathcal{H}. \tag{2.6}$$

Based on Assumption 2.3-2.4, we can infer that for every $x \in \mathcal{H}$, there exist positive constants κ and γ such that

- $\|x\| \leq \kappa$.
- $\|q(x)\| \leq \gamma$.
- $\{x_k\} \in \mathcal{H}$ as $\eta(x_k)$ is decreasing.

Remark 2.5. The sequence $\{x_k\}$ generated by Algorithm 1 is contained in the level set $\mathcal{H}$ defined above.

Proof. Using the modified wolf line search (1.3), and sufficient descent condition define (2.4), we have

$$\begin{aligned}\eta(x_{k+1}) &\leq \eta(x_k) + \phi\alpha_k q_k^T d_k - \sigma\alpha_k^2\|d_k\|^2 \\ &\leq \eta(x_k) - c\phi\alpha_k\|q_k\|^2 \\ &\leq \eta(x_k). \end{aligned} \tag{2.7}$$

Therefore for $j \geq 0$, from the above inequality, we obtain

$$\begin{aligned}&\eta(x_{j+1}) \leq \eta(x_j) \\ &\eta(x_{j+2}) \leq \eta(x_{j+1}) \leq \eta(x_j) \\ &\eta(x_{j+3}) \leq \eta(x_{j+2}) \leq \eta(x_{j+1}) \leq \eta(x_j) \\ &\qquad\vdots \\ &\eta(x_{j+k}) \leq \eta(x_j). \end{aligned} \tag{2.8}$$

Thus from the inequality (2.8), we have $\eta(x_k) \leq \eta(x_0)$. This shows that the sequence $\{x_k\}$ is contained in the set $\mathcal{H}$. ∎

This lemma plays a crucial role in demonstrating the global convergence of the method.

Lemma 2.6. *Suppose that the sequence $\{q_k\}$ and $\{d_k\}$ are generated by **NH Algorithm** then we have*

$$\alpha_k \geq \frac{(1-\rho)\|q_k\|^2}{L\|d_k\|^2}. \tag{2.9}$$

Proof. Using Chauchy-Schwartz inequality and considering equation (1.4), (2.6) and, we have

$$\begin{aligned}(\rho-1)q_k^T d_k &\leq (q_{k+1}-q_k)^T d_k \\ &\leq \|q_{k+1}-q_k\|\|d_k\| \\ &\leq \alpha_k L\|d_k\|^2, \end{aligned} \tag{2.10}$$

from the inequalities above, we have

$$\alpha_k \geq \frac{(1-\rho)\|q_k\|^2}{L\|d_k\|^2}.$$

∎

Lemma 2.7. *Suppose that the sequence* $\{q_k\}$ *and* $\{d_k\}$ *are generated* ***NH Algorithm****, and the step-length* α_k *is obtained via the modified Wolfe-type line search* (1.4)*, then we have*

$$\sum_{k=1}^{\infty} \frac{\|q_k\|^4}{\|d_k\|^2} < +\infty. \tag{2.11}$$

Proof. Using equation (1.3),(1.4), (2.6) and Lemma (2.6), we obtain

$$\begin{aligned} \eta(x_k) - \eta(x_k + \alpha_k d_k) &\geq -\phi\alpha_k q_k^T d_k + \sigma\alpha_k^2 \|d_k\|^2 \\ &\geq \phi\alpha_k \|q_k\|^2 \\ &\geq \frac{\phi(1-\rho)\|q_k\|^4}{L\|d_k\|^2}. \end{aligned} \tag{2.12}$$

By summing the infinite series from the previous inequality and considering Assumption 2.3, we obtain

$$\frac{\phi(1-\rho)}{L} \sum_{k=1}^{\infty} \frac{\|q_k\|^4}{\|d_k\|^2} \leq \sum_{k=1}^{\infty} \eta(x_k) - \eta(x_k + \alpha_k d_k).$$

Let

$$\begin{aligned} s_k &= \sum_{n=0}^{k} (\eta(x_n) - \eta(x_n + \alpha_n d_n)) \\ &= \sum_{n=0}^{k} (\eta(x_n) - \eta(x_{n+1})) \\ &= \eta(x_0) - \eta(x_{k+1}). \end{aligned} \tag{2.13}$$

As we take the limit of the equation above as $k \to \infty$, we obtain

$$\lim_{k\to\infty} s_k = \lim_{k\to\infty} (\eta(x_0) - \eta(x_{k+1})).$$

Clearly, the limit above exist in $\mathbb{R}$, since it is monotone sequence, which implies

$$\frac{\phi(1-\rho)}{L} \sum_{k=1}^{\infty} \frac{\|q_k\|^4}{\|d_k\|^2} \leq \sum_{k=1}^{\infty} [\eta(x_0) - \eta(x_{k+1})] < \infty.$$

Thus,

$$\sum_{k=1}^{\infty} \frac{\|q_k\|^4}{\|d_k\|^2} < \infty.$$

∎

Remark 2.8. It is worth noting that from (1.4) and (2.4), we obtain

$$\begin{aligned} d_{k-1}^T y_{k-1} &\geq \rho q_{k-1}^T d_{k-1} - d_{k-1}^T q_{k-1} \\ &\geq -(1-\rho) q_{k-1}^T d_{k-1} \\ &\geq (1-\rho)\|q_{k-1}\|^2 > 0. \end{aligned} \tag{2.14}$$

Theorem 2.9. *Suppose that the sequence* $\{q_k\}$ *and* $\{d_k\}$ *are generated by FR-DY-HS-PRP, then*

$$\liminf_{k\to\infty} \|q_k\| = 0. \tag{2.15}$$

Proof. Suppose by contradiction (2.15) is not true, that is, there exists a constant $r > 0$ such that

$$\|q_k\| > r, \text{ for some } k \in \mathbb{N}\bigcup\{0\}. \tag{2.16}$$

From (2.2), (2.6), (2.14), (2.16), and $\varpi = \|x_k\| + \|x_{k-1}\|$. we obtain

$$\begin{aligned}
|\beta_k^{NH}| &= \left|\frac{r_1 q_k^T y_{k-1} + r_2\|q_k\|^2}{r_3\|q_{k-1}\|^2 + r_4 d_{k-1}^T y_{k-1}}\right| \\
&\leq \frac{|r_1|\|q_k\|\|y_{k-1}\| + |r_2|\|q_k\|^2}{|r_3|\|q_{k-1}\|^2 + |r_4||d_{k-1}^T y_{k-1}|} \\
&\leq \frac{r_1\|q_k\|\|y_{k-1}\| + r_2\|q_k\|^2}{r_3\|q_{k-1}\|^2 + r_4 d_{k-1}^T y_{k-1}} \\
&\leq \frac{r_1\|q_k\|\|y_{k-1}\| + r_2\|q_k\|^2}{r_4 d_k^T y_{k-1}} \\
&\leq \frac{r_1\|q_k\|\|y_{k-1}\| + r_2\|q_k\|^2}{r_4(1-\rho)\|q_{k-1}\|^2} \\
&\leq \frac{r_1 L\|q_k\|(\|x_k - x_{k-1}\|) + r_2\|q_k\|^2}{r_4(1-\rho)\|q_{k-1}\|^2} \\
&\leq \frac{r_1 L\|q_k\|(\|x_k\| + \|x_{k-1}\|) + r_2\|q_k\|^2}{r_4(1-\rho)\|q_{k-1}\|^2} \\
&\leq \frac{2r_1 L\varpi\gamma + r_2\gamma^2}{r_4(1-\rho)r^2} \\
&\leq M_\beta,
\end{aligned} \tag{2.17}$$

where $M_\beta = \frac{2r_1 L\varpi\gamma + r_2\gamma^2}{r_4(1-\rho)r^2}$.

Therefore, from (2.3) and (2.17) we obtain

$$\begin{aligned}
|\lambda_k^{NH}| &= \left|\beta_k^{NH}\frac{q_k^T s_{k-1}}{\|q_k\|^2}\right| \\
&\leq |\beta_k^{NH}|\frac{\|q_k\|\|s_{k-1}\|}{\|q_k\|^2} \\
&\leq |\beta_k^{NH}|\frac{\alpha_{k-1}\|d_{k-1}\|}{\|q_k\|},
\end{aligned} \tag{2.18}$$

Therefore, from (2.1),(2.17) and (2.18) we have

$$\begin{aligned}\|d_k\| &= \| - q_k + \beta_k^{NH} s_{k-1} - \lambda_k^{NH} q_k\| \\ &\leq \|q_k\| + |\beta_k^{NH}|\|s_{k-1}\| + |\lambda_k^{NH}|\|q_k\| \\ &\leq \|q_k\| + M_\beta \|s_{k-1}\| + \frac{|\beta_k^{NH}|\alpha_{k-1}\|d_{k-1}\|}{\|q_k\|}\|q_k\| \\ &\leq \|q_k\| + M_\beta \alpha_{k-1}\|d_{k-1}\| + M_\beta \alpha_{k-1}\|d_{k-1}\| \\ &= \|q_k\| + 2M_\beta \alpha_{k-1}\|d_{k-1}\| \\ &\leq \gamma + 2M_\beta \alpha_{k-1}\|d_{k-1}\|. \end{aligned} \tag{2.19}$$

Also, since $\{\eta(x_k)\}$ is decreasing, noting that η is bounded below and summing both sides of (1.3) we have

$$\begin{aligned}\sum_{k=1}^{\infty}(-\phi\alpha_k q_k^T d_k + \sigma\alpha_k^2\|d_k\|^2) &\leq \sum_{k=1}^{\infty}(\eta(x_k) - \eta(x_{k+1})) \\ &< \infty. \end{aligned} \tag{2.20}$$

Since both terms in (2.20) are positive for all values of k, it follows that

$$\sum_{k=1}^{\infty}\alpha_k^2\|d_k\|^2 < \infty, \tag{2.21}$$

and hence

$$\lim_{k\to\infty}\alpha_k\|d_k\| = 0. \tag{2.22}$$

From the fact that $\lim_{k\to\infty}\alpha_k\|d_k\| = 0$, it follows that for every $r_1 > 0$, there exist k_0 such that $\alpha_{k-1}\|d_{k-1}\| < r_1$ for every $k > k_0$. Choosing $r_1 = r^2$ and $M = \max\{\|d_0\|, \|d_1\|, \ldots, \|d_{k_0}\|, M_1\}$ where $M_1 = \gamma + 2M_\beta\alpha_{k-1}\|d_{k-1}\|$, it holds that

$$\|d_k\| \leq M, \text{ for all } k \in \mathbb{N}. \tag{2.23}$$

Additionally, it is easy to see that from (2.16) and (2.23) we obtain

$$\sum_{k=0}^{\infty}\frac{\|q_k\|^4}{\|d_k\|^2} \geq \sum_{k=0}^{\infty}\frac{r^4}{M^2} = +\infty, \tag{2.24}$$

which contradicts (2.11). Thus, the result is true. ∎

3. Numerical Simulation

This part presents the findings from numerical experiments aimed at verifying the numerical efficacy of the **NH method**. In the comparison, the method is evaluated alongside **TTCDDY** [3], **MPRP** [54], **HS** [50], and **FR** [58]. A total of 81 test functions were considered during the experiments. The evaluation of the methods is based on metrics such as the number of iterations (NOI), the number of function evaluations (NOF), and the CPU run time. The experiments were conducted on an HP 2000 PC with 4.00 GB of RAM, running a 64-bit Windows 10 Pro operating system. The following conditions were applied while executing the code:

- ♢ Parameters: For the proposed method called **NH method**, we use $\sigma = 0.001$, $\phi = 0.01$, $\rho = 0.1$, $r_1 = r_2 = 0.2$, $r_3 = 0.01$ and $r_4 = 0.25$.

♢ Parameters: The methods used for comparison, namely; **TTCDDY**, **MPRP**, **FR**, and **HS** methods, were adopted from their original reference.
♢ Stopping criteria: The iteration will terminate if $\|g_k\| \leq 10^{-6}$ or after 2000 iterations.

TABLE 1. The Benchmark problems and their dimensions

No	Problem/Dimension	No	Problem/Dimension
1	COSINE 6000	60	PEN2 100
2	COSINE 100,000	61	TRID 500
3	COSINE 800,000	62	TRID 50
4	DIXMAANA 2000	63	HIMMELH 70,000
5	DIXMAANA 30,000	64	HIMMELH 240,000
6	DIXMAANB 8000	65	BD 4
7	DIXMAANB 16,000	66	EXBEALE 5000
8	DIXMAANC 900	67	EXBEALE 10,000
9	DIXMAANC 9000	68	HIMMELBC 500,000
10	DIXMAAND 4000	69	HIMMELBC 1,O00000
11	DIXMAAND 30,000	70	ARWHEAD1 100
12	DIXMAANE 800	71	ARWHEAD1 1000
13	DIXMAANF 5000	72	ENGVA111 500,000
14	DIXMAANG 4000	73	ENGVA111 1,000000
15	DIXMAANH 2000	74	DENSCHNA 500,000
16	DIXMAANJ 5000	75	DENSCHNA 1,000000
17	DIXMAANK 4000	76	DENSCHNB 500,000
18	DIXMAANK 40	77	DENSCHNB 1,000000
19	DIXMAAN1 8000	78	DENSCHNC 500,000
20	DQDRTIC 9000	79	DENSCHNF 500,000
21	DQDRTIC 90,000	80	DENSCHNF 1,000000
22	EDESCH 7000	81	ENGVAL18 500,000
23	EDESCH 40000		
24	EDESCH 500000		
25	FLETCHCR 1000		
26	FLETCHCR 50,000		
27	FLETCHCR 200,000		
28	LIAWHD 15		
29	PENALTY1 1000		
30	PENALTY1 8000		
31	BDEXP 500,000		
32	EXDENSCHNF 90,000		
33	EXDENSCHNF 280,000		
34	EXDENSCHNF 600,00		
35	EXDENSCHNB 6000		
36	EXDENSCHNB 24,000		
37	EXDENSCHNB 300,000		
38	GENQUARTIC 900		
39	GENQUARTIC 90,000		

(Continued on next page)

Table 1 – *Continued*

No	Problem/Dimension	No	Problem/Dimension
40	GENQUARTIC 500,000		
41	BIGGSB1 110		
42	SINE 100,000		
43	NONSCOMP 5000		
44	NONSCOMP 80,000		
45	RAYDAN1 500		
46	RAYDAN2 2000		
47	RAYDAN2 20,000		
48	RAYDAN2 500,000		
49	DIAGONA12 100		
50	DIAGONA12 1000		
51	DIAGONA13 500		
52	BV 2000		
53	BV 20,000		
54	IE 500		
55	IE 1500		
56	LIN 100		
57	LIN 500		
58	PEN1 200		
59	PEN1 1000		

TABLE 2. Numerical results of NH, CDDY, MPRP, FR, and HS methods, respectively, for Problems 1-41.

Problems	NH				CDDY				MPRP				FR				HS			
	CPU	NOF	Nq	NOI	CPU	NOF	Nq	NOI	CPU	NOF	Nq	NOI	CPU	NOF	Nq	NOI	CPU	NOF	Nq	NOI
1	13.4434	2721	2721	1870	0.6524	139	139	52	0.5963	127	127	52	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
2	29.0347	493	493	252	25.6577	424	424	146	23.3952	403	403	165	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
3	53.4906	108	108	29	133.3531	268	268	83	191.3577	395	395	138	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
4	2.5009	86	86	22	2.3494	81	81	16	2.2622	80	80	13	2.334	84	84	22	4.979	180	180	30
5	36.7143	95	95	27	35.8827	94	94	25	31.9266	84	84	16	40.8023	107	107	31	83.3439	218	218	43
6	9.9733	95	95	28	9.7136	93	93	17	8.7499	84	84	13	24.0836	230	230	84	19.216	180	180	32
7	18.9458	92	92	23	18.4383	90	90	21	18.0154	88	88	18	17.9947	88	88	19	37.6746	183	183	33
8	1.339	95	95	27	1.2836	90	90	22	1.1696	83	83	16	1.674	123	123	42	1.911	141	141	27
9	11.4123	97	97	27	10.7134	89	89	24	10.5358	91	91	19	10.9769	94	94	25	18.1761	156	156	26
10	4.7437	88	88	17	6.1031	113	113	30	5.0433	93	93	23	8.6832	162	162	54	9.2498	171	171	40
11	33.361	86	86	18	33.4888	88	88	18	38.0092	99	99	26	38.5265	101	101	20	69.8817	183	183	42
12	5.2976	432	432	221	13.0015	1042	1042	542	7.8048	630	630	408	29.1803	2433	2433	1113	8.0565	665	665	410
13	102.0033	1532	1532	787	224.0054	332	332	1731	149.827	2250	2250	1453	FAIL	FAIL	FAIL	FAIL	71.2335	1021	1021	597
14	52.3855	940	940	473	189.3781	3427	3427	1782	122.5445	2089	2089	1346	FAIL	FAIL	FAIL	FAIL	69.3522	1236	1236	794
15	24.086	818	818	427	65.0083	2258	2258	1209	48.0877	1713	1713	1033	FAIL	FAIL	FAIL	FAIL	25.7686	866	866	552
16	91.3244	1360	1360	696	70.1453	964	964	498	128.2221	1724	1724	1056	FAIL	FAIL	FAIL	FAIL	136.6656	1943	1943	1215
17	79.1246	1459	1459	738	43.6608	817	817	441	107.3569	2019	2019	1383	79.1609	1384	1384	622	100.1021	1869	1869	1140
18	1.8482	936	936	483	3.1409	1798	1798	934	2.8912	829	829	552	FAIL	FAIL	FAIL	FAIL	1.4413	858	858	527
19	26.598	239	239	111	117.2388	1109	1109	591	233.6101	2273	2273	1239	FAIL	FAIL	FAIL	FAIL	178.7242	1708	1708	1076
20	0.6928	263	263	91	0.8414	318	318	131	3.5	1205	1205	845	0.6175	330	330	116	0.478	259	259	57
21	4.5001	244	244	81	5.5508	251	251	105	13.6283	638	638	426	4.2621	270	270	96	5.0777	316	316	91
22	2.2293	172	172	50	1.9806	200	200	42	2.5054	258	258	48	5.7513	605	605	148	26.0103	2754	2754	364
23	25.8223	480	480	73	32.2052	503	503	75	26.338	498	498	69	FAIL	FAIL	FAIL	FAIL	81.0104	1479	1479	356
24	628.3524	1071	1071	134	168.3493	253	253	43	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
25	0.1786	235	235	148	0.5346	1169	1169	155	0.6133	1108	1108	142	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
26	4.5806	301	301	181	35.019	2757	2757	291	33.4895	2801	2801	299	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
27	24.1814	415	415	183	117.3861	2441	2441	259	25.2704	493	493	130	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
28	0.1706	225	225	132	0.0936	136	136	50	0.262	160	160	65	0.0764	198	198	54	0.0736	210	210	33
29	6.5568	211	211	33	6.5807	210	210	30	7.2668	233	233	31	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
30	73.3784	100	100	20	67.9561	92	92	17	135.2688	171	171	30	70.4055	94	94	21	350.762	426	426	80
31	8.6563	14	14	2	7.4219	12	12	2	10.2432	19	19	2	8.5768	14	14	2	FAIL	FAIL	FAIL	FAIL
32	3.0652	104	104	21	3.3229	107	107	26	3.2274	109	109	26	4.9345	173	173	52	7.035	249	249	49
33	10.2331	108	108	26	11.1545	112	112	28	11.0802	114	114	28	25.1789	274	274	104	14.4973	160	160	37
34	22.9352	114	114	28	27.2196	116	116	33	22.6407	118	118	27	187.7032	952	952	417	47.2886	241	241	54
35	0.1694	90	90	25	0.1685	91	91	24	0.1867	83	83	21	0.116	86	86	23	0.2176	156	156	31
36	0.4581	89	89	23	0.4617	80	80	20	0.43	89	89	22	0.3969	90	90	22	0.7799	174	174	33
37	5.7109	100	100	27	5.7352	94	94	22	5.1354	96	96	22	6.2001	119	119	39	15.2884	308	308	45
38	0.2386	83	83	20	0.2891	85	85	20	0.307	84	84	17	0.253	104	104	37	1.1851	509	509	121
39	2.0076	93	93	23	2.0439	90	90	20	2.7735	131	131	32	3.8538	186	186	64	125.4965	6148	6148	1352
40	13.1807	107	107	30	15.7916	111	111	29	15.1222	117	117	30	24.1783	214	214	57	FAIL	FAIL	FAIL	FAIL
41	0.5027	1333	1333	652	0.7819	1969	1969	1019	0.3686	853	853	446	FAIL	FAIL	FAIL	FAIL	0.3257	1006	1006	619

TABLE 3. Numerical results of NH, CDDY, MPRP, FR, and HS methods, respectively, for Problems 1-41.

Problems	NH				CDDY				MPRP				FR				HS			
	CPU	NOF	Nq	NOI	CPU	NOF	Nq	NOI	CPU	NOF	Nq	NOI	CPU	NOF	Nq	NOI	CPU	NOF	Nq	NOI
42	44.3193	754	754	241	69.3068	1187	1187	418	28.5096	513	513	168	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
43	0.2413	132	132	57	0.2378	120	120	46	0.3553	156	156	62	FAIL	FAIL	FAIL	FAIL	2.6029	1981	1981	629
44	3.0355	158	158	74	5.3344	231	231	103	3.1091	164	164	64	23.0361	1190	1190	475	13.2139	815	815	191
45	0.959	1435	1435	934	0.2524	455	455	236	0.3418	504	504	246	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
46	0.0732	71	71	14	0.078	71	71	14	0.0945	71	71	14	0.0566	80	80	12	FAIL	FAIL	FAIL	FAIL
47	0.4721	82	82	12	0.4672	82	82	12	0.4821	77	77	18	0.6493	117	117	35	0.8667	165	165	25
48	16.9552	122	122	20	45.5729	328	328	73	11.7423	86	86	19	56.5402	440	440	102	164.76	1263	1263	204
49	0.1098	254	254	108	0.0806	150	150	76	0.1553	186	186	88	0.1363	474	474	208	0.0696	215	215	89
50	0.6279	831	831	351	0.557	654	654	346	0.5097	635	635	319	0.9164	1390	1390	662	0.3364	458	458	229
51	2.8026	4833	4833	1896	0.6755	1256	1256	319	0.9124	1755	1755	338	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
52	32.8015	539	539	322	14.1725	212	212	102	9.9555	175	175	83	FAIL	FAIL	FAIL	FAIL	19.1522	346	346	46
53	35.278	1	1	1	40.2701	1	1	1	27.8471	1	1	1	19.6848	1	1	1	22.0962	1	1	1
54	20.8231	61	61	15	17.4468	60	60	14	18.4498	63	63	17	18.0058	55	55	16	30.6812	94	94	21
55	173.0902	64	64	16	169.7772	64	64	17	173.0053	66	66	19	186.6768	67	67	19	352.9132	132	132	32
56	4.2342	63	63	13	3.048	63	63	13	3.0547	63	63	13	3.2581	63	63	13	8.3659	173	173	30
57	5.722	82	82	18	4.8566	72	72	15	5.2019	72	72	15	6.2815	90	90	20	9.7144	139	139	22
58	13.8201	1087	1087	153	13.6036	1078	1078	341	2.9037	216	216	54	FAIL	FAIL	FAIL	FAIL	13.0289	1031	1031	136
59	49.3976	583	583	103	198.3397	2406	2406	483	22.439	276	276	60	FAIL	FAIL	FAIL	FAIL	160.2805	1602	1602	277
60	2.1592	987	987	530	0.5618	355	355	153	0.9985	537	537	163	FAIL	FAIL	FAIL	FAIL	1.1198	1099	1099	233
61	1.1137	135	135	72	1.6143	188	188	103	1.0066	118	118	49	7.6453	996	996	450	1.4563	185	185	93
62	0.1276	119	119	65	0.0943	91	91	36	0.0945	93	93	37	0.4913	575	575	264	0.1325	146	146	48
63	4.5063	169	169	32	8.5145	323	323	50	8.6761	328	328	45	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
64	54.7813	634	634	68	135.2618	1512	1512	169	31.9842	367	367	53	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
65	0.8168	2370	2370	431	1.0982	3251	3251	445	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
66	2.2829	320	320	126	FAIL	FAIL	FAIL	FAIL	5.436	755	755	496	5.3842	820	820	348	FAIL	FAIL	FAIL	FAIL
67	3.0256	334	334	78	3.7695	278	278	119	16.3565	1190	1190	975	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
68	12.309	101	101	25	14.175	108	108	28	12.8394	108	108	28	20.9634	185	185	70	25.283	233	233	40
69	26.6127	112	112	31	27.3643	107	107	26	28.4369	106	106	28	60.3699	265	265	113	57.7971	259	259	43
70	0.0419	67	67	21	0.0623	116	116	27	0.1311	71	71	22	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
71	0.0426	75	75	20	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
72	129.0401	1141	1141	124	160.3513	1313	1313	143	153.0154	1346	1346	145	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
73	393.443	1795	1795	198	1.13E+03	4956	4956	456	479.1131	2098	2098	218	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
74	48.2984	121	121	40	41.4164	101	101	28	49.7363	108	108	33	58.1715	128	128	40	87.0857	227	227	36
75	81.7283	100	100	31	87.2568	107	107	29	87.9374	103	103	27	134.6755	165	165	47	201.2618	255	255	43
76	11.5788	114	114	28	14.0196	113	113	28	12.2114	115	115	32	25.3341	272	272	50	25.4303	272	272	50
77	24.5365	116	116	27	25.9817	111	111	26	21.2403	107	107	22	33.1998	160	160	42	50.0088	226	226	45
78	0.1173	129	129	37	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
79	109.2014	598	598	173	715.7783	3681	3681	1270	117.8893	608	608	190	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
80	540.8199	1485	1485	444	995.0614	2534	2534	874	472.8336	1260	1260	455	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
81	1.53E+03	13320	13320	1317	470.0815	4078	4078	413	326.2577	2994	2994	304	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL

A graphical representation will be utilized since it's challenging to ascertain the optimal method from Tables 2 and 3. In this context, we adopted the widely recognized performance profile introduced by Dolan and More [2] to identify the most effective method among those analyzed.

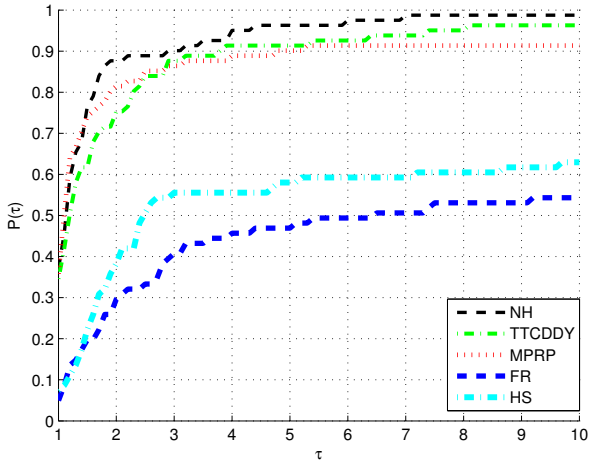


Figure 1. Performance Profiles on NOI.

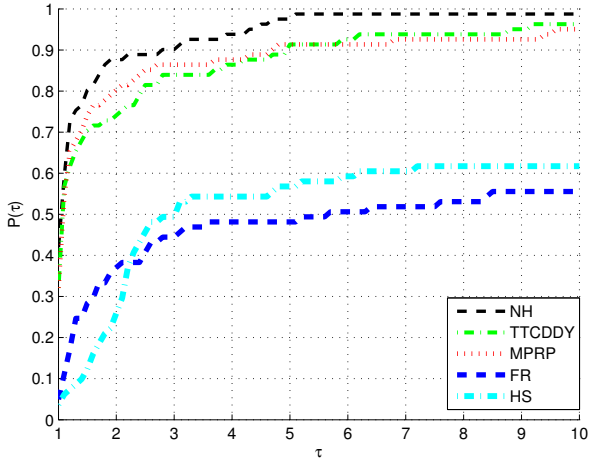


Figure 2. Performance Profiles on NOF.

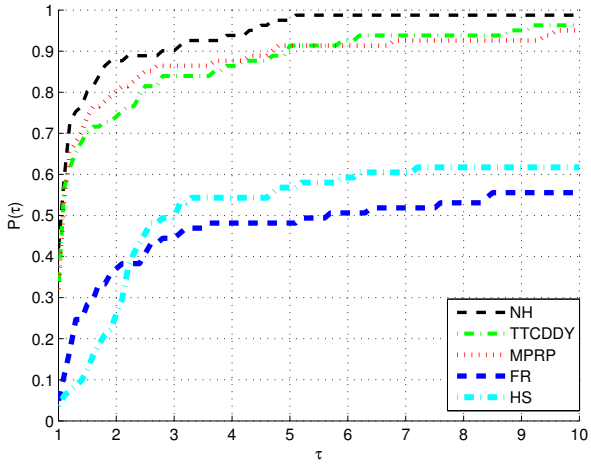


Figure 3. Performance Profiles on Ng.

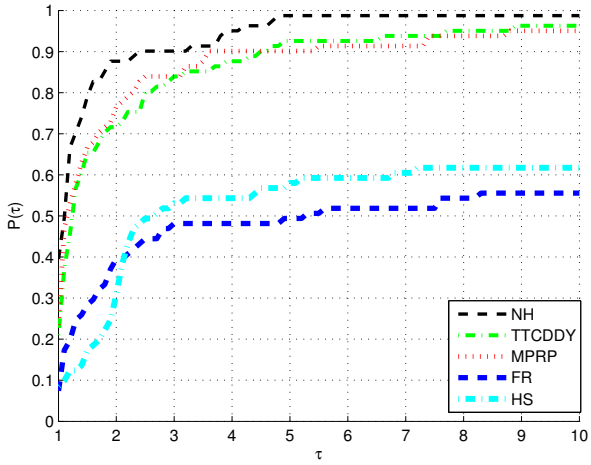


Figure 4. Performance Profiles on CPU Time.

Figures 1-4 depict the performance profiles based on four metrics: NOI, NOF, Ng, and CPU run time. Figure 1 shows the performance profile of the methods concerning the number of iterations. Near the origin, for values of τ between 0 and 1, the curves for the NH, TTCDDY, and MPRP methods overlap, indicating that all three methods perform similarly well within this range. Subsequently, the curve for the NH method rises to the top, signifying that it is the most effective in terms of iterations. Figure 2 illustrates the performance profiles concerning the number of function evaluations. The curves for TTCDDY and MPRP are relatively close to that of the NH method, highlighting a competitive edge among these three methods. The NH curve remains at the top, affirming its

superiority over the other comparison methods. Figure 3 presents the performance profile concerning the number of gradient evaluations, where the NH method is again positioned at the top, indicating its status as the most effective solver. Lastly, Figure 4 showcases the performance profile concerning CPU run time, with the NH method's curve above the others, indicating its best performance among the compared methods. Overall, these figures highlight the effectiveness of the NH method compared to the TTCDDY, MPRP, FR, and HS methods.

The numerical results indicate that the frequent breakdowns of the FR and HS methods, together with the occasional failures of CDDY and MPRP, stem from a combination of inherent algorithmic limitations, the high demands imposed by nonlinear benchmark problems, and the strong dependence on accurate line-search strategies. The classical FR and HS schemes are especially vulnerable because of their well-known susceptibility to numerical inaccuracies and jamming effects, which are intensified in ill-conditioned, non-convex, or poorly scaled problem settings in which their theoretical conjugacy properties become harmful rather than beneficial. The intermittent failures of the otherwise more robust CDDY and MPRP methods on certain instances (for example, problem 24 in the case of MPRP) further demonstrate that even sophisticated algorithms can be undermined by pathological curvature or very high dimensionality. This reinforces the view that no single method is uniformly best and that algorithmic performance is fundamentally shaped by the numerical characteristics of the problem and the robustness of the line-search mechanism.

In the performance profiles presented in Figures 1-4, any solver that failed to converge on a particular test problem was assigned a large penalty value. Specifically, failures were handled by setting the corresponding performance metric (NOI, NOF, Nq, or CPU time) to a value of for non-convergent cases. This approach follows the standard practice in benchmarking nonlinear optimization methods, ensuring that solvers with poor reliability are appropriately penalized in the performance comparisons. The profiles therefore, reflect both the efficiency and robustness of each method across the test set.

3.1. Application in Motion Control

This section evaluates the effectiveness of **NH Algorithm** in addressing the motion control problem. We focus on the motion control of a two-joint planar robotic manipulator, which aims to follow a desired trajectory represented by the vector $v_d \in \mathbb{R}^2$ at time $t_k \in [0, t]$. Therefore, the kinematic equation in discrete time is formulated as follows: $r_k \in \mathbb{R}^2$ such that

$$\eta(r_k) = v_d,$$

where

$$\eta(r) \mapsto \begin{bmatrix} k_1 b_1 + k_2 b_2 \\ k_1 a_1 + k_2 a_2 \end{bmatrix}$$

with $b_1 = \cos r_1$, $a_1 = \sin r_1$, $b_2 = \cos(r_1 + r_2)$, $s_2 = \sin(r_1 + r_2)$, and k_i represent the length of the i-th rod . Hence, the problem can be formulated as:

$$\min_{r_k \in \mathbb{R}^2} \frac{1}{2} \|f(r_k) - v_d\|^2. \tag{3.1}$$

For implementation sake, we chose $k_i = 1$ $(i = 1, 2)$ and the end-effector is controlled to track the Lissajous curve [21] given by

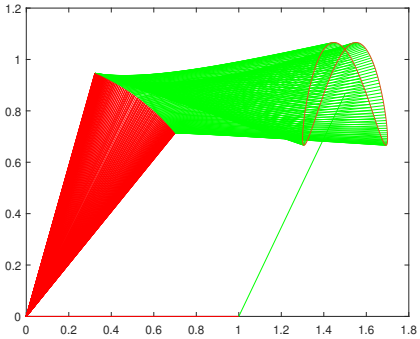
$$v_d = \begin{bmatrix} 1.5 + 0.2\sin(\pi t_k/5) \\ \sqrt{3}/2 + 0.2\sin(2\pi t_k/5 + \pi/3) \end{bmatrix}.$$

The initial problem starts with $r_0 = [0, \pi/3]^\top$, and for each subsequent problem, the solution approximates that of the previous problem for $i = 2, 3, \ldots$. The task duration ranges from $[0, 10]$ and is segmented into 200 equal intervals. To address problem (3.1), we employed **NH Algorithm**. The implementation featured a modified Wolfe line search as described by (1.3)-(1.4), with the parameters set to $\rho = 0.8$, $\sigma = 0.001$, $\phi = 0.01$, and $r_1 = r_2 = r_3 = r_4 = 1$.

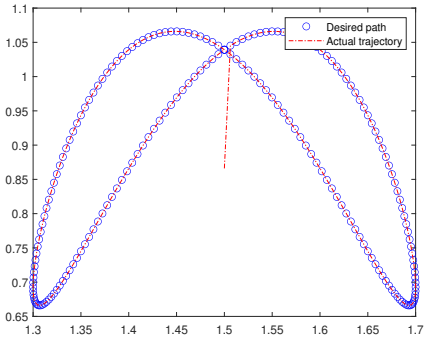
The results presented in Figures 5 clearly demonstrate that the proposed NH Algorithm performs reliably in generating smooth and accurate robot trajectories. Figure 5 confirms the successful execution of the overall task, while Figure 5(A) shows that the algorithm produces synthesized joint-space trajectories capable of driving the robot along the intended motion. Similarly, Figure 5(B) illustrates that the end-effector trajectory closely follows the desired path, exhibiting no visually noticeable deviation. These observations provide initial evidence of the correctness and consistency of the proposed hybrid CG framework when applied to motion control.

Beyond trajectory tracking, the results also provide strong indications of stability of the NH algorithm. Figures 5(C) and 5(D) reveal that the tracking errors along both the x and y axis remains below 10^{-5} demonstrating that the algorithm maintains extremely small deviations throughout the entire motion. The absence of oscillatory behavior, divergence, or abrupt fluctuations in the error plots reflects a stable closed-loop response under the proposed CG-based updates. Since the hybrid CG direction integrates descent properties with conjugacy features, the resulting update rule ensures monotonic error reduction, thereby contributing to stable convergence of the overall control scheme.

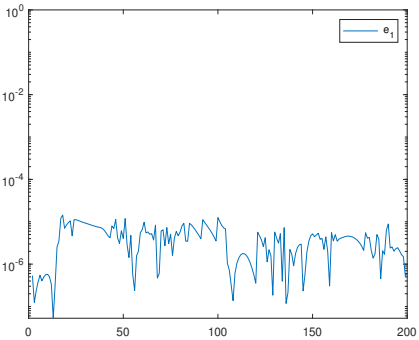
Furthermore, the consistently low error values and smooth trajectory generation highlight the robustness of the NH algorithm with respect to numerical and model-related uncertainties. In practical motion control settings, small variations in the system model, discretizations, or initial conditions often lead to noticeable deviations in end-effector tracking. However, the proposed NH algorithm demonstrates resilience by maintaining accuracy across the entire trajectory without requiring parameter retuning or additional corrective layers. The combination of global convergence characteristics and efficient gradient information utilization allows the algorithm to preserve performance even in the presence of minor disturbances. Collectively, these results emphasize that the NH algorithm is not only effective in task execution but also exhibits strong stability and robustness properties within the motion control framework.



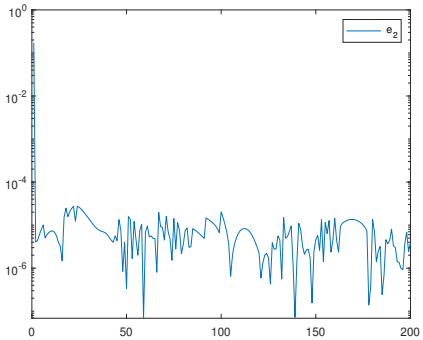
(A) Manipulator trajectories



(B) End-effector trajectory and desired path



(C) Tracking errors on the horizontal x-axis



(D) Tracking errors on the vertical y-axis

Figure 5. Numerical results generated by NH Algorithm

Furthermore, to illustrate the absolute error produced by **NH Algorithm**, we present the Euclidean distance between the end-effector trajectory and the desired path in Figure 5(C)-5(D). The results indicate that the absolute error remains below 3.5×10^{-5}, demonstrating that **NH Algorithm** achieves the task with remarkable precision.

In Figure 5(A), one of the lines stands out as significantly distant from the others, suggesting an abrupt shift in the manipulator's motion. This could potentially lead to harm to the manipulator. Consequently, alongside the unconstrained optimization model (3.1), it is essential to develop a new mathematical model for motion control that ensures smooth and consistent movement. This represents an area for further research.

4. Conclusion

This paper presents a hybrid nonlinear CG method for solving unconstrained optimization. The CG parameter is a hybrid of FR, DY, HS, and PRP classical CG parameters. It is essential to notice that the hybrid CG parameter behaves in four cases; the search direction is proven to be a descent and bounded regardless of the cases and the line search technique used. The method is globally converged via a modified Wolfe line search and

the assumption that the objective function is Lipschitz. The numerical experiment compared the NH method with two classical CG and other methods designed in the past decade. The results reveal that the NH method is quite preferable in terms of efficiency. The applicability of the NH method in the motion control problem was also explored.

Acknowledgements

The second author thanks the Department of Mathematics and Applied Mathematics at Sefako Makgatho Health Sciences University.

Conflict of interest

The authors declare that there is no conflict of interest.

References

[1] S. Jamilu, M.S. Ibrahim, P. Kaelo, M. Malik, A.K. Saadi, An optimal choice Dai-Liao conjugate gradient algorithm for unconstrained optimization and portfolio selection, AIMS Mathematics 9 (1) (2024) 642–664. Available from: https://doi.org/10.3934/math.2024034.

[2] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, Mathematical Programming 91 (2002) 201–213. Available from: https://doi.org/10.1007/s101070100263.

[3] J. Deepho, A.B. Abubakar, M. Malik, I.K. Argyros, Solving unconstrained optimization problems via hybrid CD-DY conjugate gradient methods with applications, Journal of Computational and Applied Mathematics 405 (2022) 113823. Available from: https://doi.org/10.1016/j.cam.2021.113823.

[4] A.B. Abubakar, P. Kumam, M. Malik, P. Chaipunya, A.H. Ibrahim, A hybrid FR-DY conjugate gradient algorithm for unconstrained optimization with application in portfolio selection, AIMS Mathematics 6 (6) (2021) 6506–6527. Available from: https://doi.org/10.3934/math.2021383.

[5] A.B. Abubakar, P. Kumam, M. Malik, A.H. Ibrahim, A hybrid conjugate gradient based approach for solving unconstrained optimization and motion control problems, Mathematics and Computers in Simulation 201 (2022) 640–657. Available from: https://doi.org/10.1016/j.matcom.2021.05.038.

[6] A.B. Abubakar, M. Malik, P. Kumam, H. Mohammad, M. Sun, A.H. Ibrahim, A.I. Kiri, A Liu-Storey-type conjugate gradient method for unconstrained minimization problem with application in motion control, Journal of King Saud University – Science 34 (4) (2022) 101923. Available from: https://doi.org/10.1016/j.jksus.2022.101923.

[7] R. Fletcher, Practical methods of optimization, John Wiley & Sons, 2000. Available from: https://doi.org/10.1002/9781118723203.

[8] J. Liu, Y. Feng, L. Zou, Some three-term conjugate gradient methods with the inexact line search condition, Calcolo 55 (2018) 1–16. Available from: https://doi.org/10.1007/s10092-018-0258-3.

[9] M. Malik, I.M. Sulaiman, A.B. Abubakar, G. Ardaneswari, Sukono, A new family of hybrid three-term conjugate gradient method for unconstrained optimization with application to image restoration and portfolio selection, AIMS Mathematics 8 (1) (2023) 1–28. Available from: https://doi.org/10.3934/math.2023001.

[10] A. Abubakar, K. Duangkamon, S. Thidaporn, An accelerated Halpern-type algorithm for solving variational inclusion problems with applications, Bangmod International Journal of Mathematical and Computational Science 8 (1) (2022) 37–55. Available from: https://doi.org/10.58715/bangmodjmcs.2022.8.4.

[11] A.H. Ibrahim, P. Kumam, A.B. Abubakar, A. Adamu, Accelerated derivativefree method for nonlinear monotone equations with an application, Numerical Linear Algebra with Applications 29 (3) (2022) e2424. Available from: https://doi.org/10.1002/nla.2424.

[12] S. Sangkhae, W. Sanhan, T. Jitpeera, C. Mongkolkeha, The Hybrid Steepest Descent Method with Implicit Double Midpoint for Solving Variational Inequality over Triple Hierarchical Problems, Bangmod International Journal of Mathematical and Computational Science 11 (2025) 1–22. Available from: https://doi.org/10.58715/bangmodjmcs.2025.11.1.

[13] Y. Li, C. Li, W. Yang, W. Zhang, A new conjugate gradient method with a restart direction and its application in image restoration, AIMS Mathematics 8 (12) (2023) 28791–28807. Available from: https://doi.org/10.3934/math.20231475.

[14] N. Salihu, P. Kumam, I.M. Sulaiman, T. Seangwattana, An efficient spectral minimization of the Dai-Yuan method with application to image reconstruction, AIMS Mathematics 8 (12) (2023) 30940–30962. Available from: https://doi.org/10.3934/math.20231583.

[15] J. Guo, Z. Wan, A new three-term conjugate gradient algorithm with modified gradient-differences for solving unconstrained optimization problems, AIMS Mathematics 8 (2) (2023) 2473–2488. Available from: https://doi.org/10.3934/math.2023128.

[16] I.A. Masmali, Z. Salleh, A. Alhawarat, A decent three term conjugate gradient method with global convergence properties for large scale unconstrained optimization problems, AIMS Mathematics 6 (10) (2021) 10742–10764. Available from: https://doi.org/10.3934/math.2021624.

[17] M. Sun, J. Liu, Y. Wang, Two improved conjugate gradient methods with application in compressive sensing and motion control, Mathematical Problems in Engineering (2020). Available from: https://doi.org/10.1155/2020/9175496.

[18] S. Aji, A.B. Abubakar, A.I. Kiri, A. Ishaku, A spectral conjugate gradient-like method for convex constrained nonlinear monotone equations and signal recovery, Nonlinear Convex Analysis and Optimization 1 (1) (2022) 1–23. Available from: https://doi.org/10.2307/j.ctv2jhjw5h.24.

[19] N. Salihu, H.A. Babando, I. Arzuka, S. Salihu, A hybrid conjugate gradient method for unconstrained optimization with application, Bangmod International Journal of Mathematical and Computational Science 9 (2023) 24–44. Available from: https://doi.org/10.58715/bangmodjmcs.2023.9.3.

[20] J. Yahaya, I. Arzuka, M. Isyaku, Descent modified conjugate gradient methods for vector optimization problems, Bangmod International Journal of Mathematical and Computational Science 9 (2023) 72–91. Available from: https://doi.org/10.58715/bangmodjmcs.2023.9.6.

[21] 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, Journal of Computational and Applied Mathematics 347 (2019) 314–329. Available from: https://doi.org/10.1016/j.cam.2018.08.017.

[22] J. Liu, S. Du, Y. Chen, A sufficient descent nonlinear conjugate gradient method for solving M-tensor equations, Journal of Computational and Applied Mathematics 371 (2020) 112709. Available from: https://doi.org/10.1016/j.cam.2019.112709.
[23] R. Ziadi, R. Ellaia, A. Bencherif-Madani, Global optimization through a stochastic perturbation of the Polak-Ribire conjugate gradient method, Journal of Computational and Applied Mathematics 317 (2017) 672–684. Available from: https://doi.org/10.1016/j.cam.2016.12.021.
[24] X. Zhu, A Riemannian conjugate gradient method for optimization on the Stiefel manifold, Computational Optimization and Applications 67 (2017) 73–110. Available from: https://doi.org/10.1007/s10589-016-9883-4.
[25] J. Liu, X. Du, A gradient projection method for the sparse signal reconstruction in compressive sensing, Applied Analysis 97 (12) (2018) 2122–2131.
[26] G. Yuan, T. Li, W. Hu, A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems, Applied Numerical Mathematics 147 (2020) 129–141. Available from: https://doi.org/10.1016/j.apnum.2019.08.022.
[27] Z. Shi, J. Guo, A new family of conjugate gradient methods, Journal of Computational and Applied Mathematics 224 (1) (2009) 444–457. Available from: https://doi.org/10.1016/j.cam.2008.05.012.
[28] E.M. Beale, A derivation of conjugate-gradients, Numerical methods for non-linear optimization, Academic Press, 1972.
[29] L. Zhang, W. Zhou, D. Li, A descent modified Polak-Ribire-Polyak conjugate gradient method and its global convergence, IMA Journal of Numerical Analysis 26 (4)(2006) 629–640. Available from: https://doi.org/10.1093/imanum/drl016.
[30] C. Kou, Y. Dai, A modified self-scaling memoryless Broyden-Fletcher-Goldfarb-Shanno method for unconstrained optimization, Journal of Optimization Theory and Applications 165 (2015) 209–224. Available from: https://doi.org/10.1007/s10957-014-0528-4.
[31] Y. Narushima, H. Yabe, J.A. Ford, A three-term conjugate gradient method with sufficient descent property for unconstrained optimization, SIAM Journal on Optimization 21 (1) (2011) 212–230. Available from: https://doi.org/10.1137/080743573.
[32] J. Liu, Y. Zhao, X. Wu, Some three-term conjugate gradient methods with the new direction structure, Applied Numerical Mathematics 150 (2020) 433–443. Available from: https://doi.org/10.1016/j.apnum.2019.10.011.
[33] J.C. Gilbert, J. Nocedal, Global convergence properties of conjugate gradient methods for optimization, SIAM Journal on Optimization 2 (1) (1992) 21–42. Available from: https://doi.org/10.3102/0013189X021001042.
[34] M. Li, A modified Hestense-Stiefel conjugate gradient method close to the memoryless BFGS quasi-Newton method, Optimization Methods and Software 33 (2) (2018) 336–353. Available from: https://doi.org/10.1080/10556788.2017.1325885.
[35] M.V. Solodov, B.F. Svaiter, A globally convergent inexact Newton method for systems of monotone equations, Reformulation: Nonsmooth, piecewise smooth, semi-smooth and smoothing methods, Springer, 1998, 355–369. Available from: https://doi.org/10.1007/978-1-4757-6388-1_18.
[36] J. Martınez, Practical quasi-Newton methods for solving nonlinear systems, Journal of Computational and Applied Mathematics 124 (1-2) (2000) 97–121. Available from: https://doi.org/10.1016/S0377-0427(00)00434-9.

[37] N. Yamashita, M. Fukushima, On the rate of convergence of the Levenberg-Marquardt method, Topics in numerical analysis, Springer, 2001, 239–249. Available from: https://doi.org/10.1007/978-3-7091-6217-0_18.
[38] S. Bellavia, M. Macconi, B. Morini, STRSCNE: A Scaled Trust-Region Solver for Constrained Nonlinear Equations, Computational Optimization and Applications 28 (1) (2004) 31–50. Available from: https://doi.org/10.1023/B:COAP.0000018878.95983.4e.
[39] D. Li, M. Fukushima, A globally and superlinearly convergent gauss-Newton-based BFGS method for symmetric nonlinear equations, SIAM Journal on Optimization 37 (1) (1999) 152-172. Available from: https://doi.org/10.1137/S0036142998335704.
[40] M. Powell, Nonconvex minimization calculations and the conjugate gradient method, Numerical Analysis: Proceedings of the 10th Biennial Conference held at Dundee, Scotland, June 28-July 1, 1983, Springer, 1984, 122–141. Available from: https://doi.org/10.1007/BFb0099521.
[41] Z. Aminifard, S. Babaie-Kafaki, A modified descent Polak-Ribire-Polyak conjugate gradient method with global convergence property for nonconvex functions, Calcolo 56 (2019) 1–11. Available from: https://doi.org/10.1007/s10092-019-0312-9.
[42] Y. Dai, Z. Liao, New conjugacy conditions and related nonlinear conjugate gradient methods, Applied Mathematics & Optimization 43 (2001) 87–101. Available from: https://doi.org/10.1007/s002450010019.
[43] S. Babaie-Kafaki, R. Ghanbari, A descent extension of the Polak-Ribiére-Polyak conjugate gradient method, Computers & Mathematics with Applications 68 (12) (2014) 2005–2011. Available from: https://doi.org/10.1016/j.camwa.2014.09.019.
[44] R. Fletcher, C.M. Reeves, Function minimization by conjugate gradients, Computational Journal 7 (2) (1964) 149–154. Available from: https://doi.org/10.1093/comjnl/7.2.149.
[45] W. Sun, Y.X. Yuan, Optimization theory and methods: nonlinear programming, Vol. 1, Springer Science & Business Media, 2006.
[46] L. Grippo, S. Lucidi, A globally convergent version of the Polak-Ribiere conjugate gradient method, Mathematical Programming 78 (3) (1997) 375–391. Available from: https://doi.org/10.1007/BF02614362.
[47] Y. Liu, C. Storey, Efficient generalized conjugate gradient algorithms, part 1: theory, Journal of Optimization Theory and Applications 69 (1991) 129-137. Available from: https://doi.org/10.1007/BF00940464.
[48] X. Jiang, W. Liao, J. Yin, J. Jian, A new family of hybrid three-term conjugate gradient methods with applications in image restoration, Numerical Algorithms 91 (1) (2022) 161–191. Available from: https://doi.org/10.1007/s11075-022-01258-2.
[49] A.B. Abubakar, P. Kumam, A.H. Ibrahim, P. Chaipunya, S.A. Rano, New hybrid three-term spectral-conjugate gradient method for finding solutions of nonlinear monotone operator equations with applications, Mathematics and Computers in Simulation 201 (2022) 670–683. Available from: https://doi.org/10.1016/j.matcom.2021.07.005.
[50] M.R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards 49 (1) (1952). Available from: https://doi.org/10.6028/jres.049.044.
[51] Y. Dai, Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM Journal on Optimization 10 (1) (1999) 177–182. Available

from: https://doi.org/10.1137/S1052623497318992.
[52] A.B. Abubakar, P. Kumam, A.H. Ibrahim, J. Rilwan, Derivative-free HS-DY-type method for solving nonlinear equations and image restoration, Heliyon 6 (11) (2020) e05400. Available from: https://doi.org/10.1016/j.heliyon.2020.e05400.
[53] A.B. Abubakar, P. Kumam, H. Mohammad, A.M. Awwal, K. Sitthithakerngkiet, A modified Fletcher-Reeves conjugate gradient method for monotone nonlinear equations with some applications, Mathematics 7 (8) (2019) 745. Available from: https://doi.org/10.3390/math7080745.
[54] A.B. Abubakar, P. Kumam, H. Mohammad, A.H. Ibrahim, PRP-like algorithm for monotone operator equations, Japanese Journal of Industrial and Applied Mathematics (2021) 1–18.
[55] A.B. Abubakar, K. Muangchoo, A.H. Ibrahim, A.B. Muhammad, L.O. Jolaoso, K.O. Aremu, A new three-term Hestenes-Stiefel type method for nonlinear monotone operator equations and image restoration, IEEE Access 9 (2021) 18262–18277. Available from: https://doi.org/10.1109/ACCESS.2021.3053141.
[56] A.B. Abubakar, P. Kumam, H. Mohammad, A.M. Awwal, A Barzilai-Borwein gradient projection method for sparse signal and blurred image restoration, Journal of the Franklin Institute 357 (11) (2020) 7266–7285. Available from: https://doi.org/10.1016/j.jfranklin.2020.04.022.
[57] P. Kumam, A.B. Abubakar, M. Malik, A.H. Ibrahim, N. Pakkaranang, B. Panyanak, A hybrid HS-LS conjugate gradient algorithm for unconstrained optimization with applications in motion control and image recovery, Journal of Computational and Applied Mathematics 433 (2023) 115304. Available from: https://doi.org/10.1016/j.cam.2023.115304.
[58] R. Fletcher, Practical methods of optimization. A Wiley Interscience Publication. 392 (1987).